Welcome to the JReport Designer User's Guide

This User's Guide describes JReport Designer. JReport Designer is a report development environment that enables sophisticated report design and presentation of critical business data. Report designers can easily access data from any data source and generate highly formatted reports to deliver information to authorized end users in the most relevant and intuitive manner.

This guide is written for report developers or Java application developers that want to embed JReport reports into their application.

Other JReport documentation

This guide is one in the complete JReport documentation set. The documentation set includes the following:

- Getting Started with JReport
- JReport Tutorial
- JReport Server User's Guide
- JReport Designer User's Guide
- JReport Server Monitor User's Guide
- JReport API Javadoc

What's New in This Release

This document introduces the new features and significant enhancements of JReport in this release.

Embedding

Integration

- Embedding web reports, page reports and dashboards as a whole using JavaScript API for host applications in web browsers.
- Embedding web controls in host applications to open, save, export, filter, sort, print and navigate.
- API for applying and modifying NLS at runtime.

Connectivity

- RESTful web services as a data source with JSON formatted data with support for authentication and parameters.
- Unified semantic layer: Business View (BV) now integrates Report Cube (RC) and Business Cube (BC) features.
- User authentication when setting up notification e-mails.
- Backdoor admin authentication when bypassing LDAP is required.

Internationalization and Customization

- Full NLS support on product UI, data, metadata, contents, formulas, and resources.
- CSS enhancements with more objects and properties, as well as better category organization.

Other JReport Server Live Enhancements

- Custom aggregations using formulas.
- One-click sort and filter on table headers.
- Custom sort on table groups.
- Support for all built-in functions for crosstab.
- Exporting selected components from a report.
- Exporting multiple tabs of a page report into multiple sheets of an Excel file.
- Local parameter support on web reports and library components.
- Comparison function to compare crosstab values within reports and dashboards.
- External web controls via user defined dialogs.
- Configurable go-to-detail report window target (e.g., same window or new window).
- Default style group to Run, Advanced Run and Schedule reports.
- Setting a default report component when filtering using the Filter dialog.
- Dynamic Java home to start up/shut down Derby database when integrated in an application server.
- JRE only (without JDK) when integrated in an application server.
- Enable/disable CSRF (Cross-Site Request Forgery) checking for JReport Server Live.
- Support for synonyms when accessing Oracle databases.
- IE Navigate2 option for linking reports to URL when viewing in IE.
- Enhanced logs with more details.

UI and UX

Appearance

- Overall UI improvements including new styles and icons.
- Enhanced catalog data source view.
- Auto scaling Y-axis when scrolling through a Scrollable Chart.
- Formatting individual lines or a group of lines.
- Option in line charts to skip null data values in order to draw continuous lines.

Easy of Use

- Property Inspector to manage object properties in Web Report Studio.
- Windows Edge browser support.
- Direct linking from charts to reports without manually setting a URL.
- Enable/disable items in Options menu in JDashboard.
- Operate (e.g., sort, filter) on all fields of the dataset in a chart.
- New default for manually inputting filter values into Filter dialogs or optionally pulling filter values from the database.
- Select and operate on multiple items at a time in Business View Editor.
- Quick search in Data Mapping Editor.

Release Notes for Enhancements and Fixes

This document describes the feature enhancements, resolved issues and any known issues for JReport in this release.

Resolved issues

Item	Case #	Issue
1	66088	Resolved the XSS security issue when accessing a few JSP pages in JReport Sever.
2	66886	Resolved the ArrayIndexOutOfBoundsException error when loading reports in JDashboard.
3	67331	The parameter sessionid which is a given value in runReport.jsp can be displayed normally in the report's parameter page when calling UDS for the first time.
4	67331	JReport Server now calls the database only once when loading the parameter value list for bind column parameters which are bound to UDSs and retrieve data from the database in a web main report.
5	67331	The value list for the bind column parameters in a web main report will no longer be triggered when opening its linked report.
6	67914	Resolved the OutOfMemory issue when SSL is turned on with JReport Server.
7	68120	Resolved the jet.export.ExpException error when changing the width of an aggregation field in a page crosstab report to 0.01 and then exporting the crosstab to an Excel .xlsx file with column format.
8	68223	JReport no longer requests data from the data source when user tries to view through JReport Server the web report component whose Invisible property has been set to true.
9	68254	Solved the empty Excel result file problem when exporting a page banded report to Excel in JReport Server.
10	68255	Resolved the duplicated table and field names issue when merging two catalogs using Catalog Doctor in JReport Designer.
11	68278	JReport Designer can now create the node of hierarchy to business view normally when saving the catalog as XML.
12	68305	Resolved the IllegalArgumentException error when migrating JReport Server from version 5 to 13 by using command MigrationBV52.bat.
13	68331	The API function cat.getHandle(mapNames[i], CatalogAPI.COLUMN) now gets handled successfully when creating aggregations for an imported SQL using JReport Designer API.
14	68345	Resolved the java.lang.NumberFormatException error when adding a filter object to a page report that is running on Internet Explorer.
15	68365	Resolved the java.lang.ArrayIndexOutOfBoundsException error when scheduling a report using a bursting schema.
16	68365	The security error message can pop up properly in linked page reports.
17	68373	Corrected the 'Data type unmatched error' when doing format filter on the design mode of a web chart report and the field text field and the value text field of the condition are both filled with ToDate() expressions.
18	68390	The word wrap function now works normally on a page crosstab report when setting properties Suppress When Empty and Suppress When No Records to true.
19	68395	JReport Server no longer pops up the error page showing 'The report session has expired.' when user sets property server.session.timeout to 20 and performs continuous filters using the Filter dialog in Page Report Studio.
20	68422	Solved the GenerateFormulasCodeException error when saving a catalog after JReport Designer has been upgraded to V13. 1 Update 1.
21	68425	The resource permission on the Public Resource folder of JReport Server now gets enabled normally when using customized Security API.
22	68448	Support exporting page reports in IE 11 browser when Page Report Studio is embedded in a page via IFrame with HTTPS (SSL) URL.
23	68449	Corrected the error of a full user name in the Security > User panel of the JReport Administration page.
24	68550	The filter panel now fully displays in the edit mode of a web report.
25	68461	The scheduled tasks can get triggered normally when user schedules to run a few tasks at a specified time with option 'Run missed task upon server restart' checked, shuts down JReport Server before the specified time and then restarts it.
26	68540	Resolved the security issue of data access race condition that caused by storing user data in Servlet member fields.
27	68547	The page report formula on JReport Designer can now get correct value when calling isnull(next(@column)).
28	68563	Resolved the java.lang.ArrayIndexOutOfBoundsException: 0 >= 0 error when logging onto JReport Server in an LDAP configured environment.
29	68596	The new line at the bottom of the table column no longer disappears in the exported Excel workbook (.xlsx) of a page table report.
30	68597	Add a blank item to the From DB Catalog drop-down list in the Input SQL Name dialog when adding a new SQL to a data source so that database and schema name defined in the URL could be applied.
31	68601	The save as function can be performed normally in Page Report Studio when JReport Server is run with a JDashboard disabled license key.
32	68601	The formatting changes of a page report no longer gets lost after saving it to a report folder to overwrite the report with the same name.
33	68608	JReport Server now pops up with a clearer error message when someone exports a web report with invalid file name characters in the Export dialog.
34	68667	Resolved the ClassCastException error when selecting Top N, specifying "other" group and using category sort in Category Options dialog of a web report.
35	68679	The node icons on the legend of a page chart report now display normally.
36	68684	Labels in web reports can be edited successfully now when their width extends beyond the page in JReport Server.
37	68688	The field items available for the Conditional Formatting dialog in a web crosstab report are staying the same after JReport Server is upgraded to a higher version.
38	68696	Solved the payment_type_name field value missing issue on Page Report Studio when Group By Transaction Type option of parameter p_GroupByTransactionType gets unchecked in the Enter Parameter Values dialog.
39	68726	Solved the ClassCastException and ConcurrentModificationException errors when initializing multiple threads concurrently to call the method runReport.
40	68728	Resolved the NullPointerException error when creating cached query result for several queries in JReport Designer.
41	68746	The Edit Business View Security dialog appears normal when right-clicking Business View Security for a catalog and then select Edit Business View Security from the shortcut menu.
42	68748	Resolved the script error when opening a report folder and its subfolders in the server resource tree in Icons or Thumbnails view.
43	68798	Resolved the data type unmatched error when choosing to use a dynamic formula in a web report format filter.
44	68816	Resolved the java.lang.NullPointerException error when running a page report in JReport Server with trace level logs set to TRIVIAL.
45	68835	Lines in page reports are now displayed normally when viewing the reports in JReport Server.
46	68842	Resolved the jave.lang.NullPointerException error when doing a drill down action in a library component running in JDashboard.
47	68842	Web Report Studio now shows the field's new dynamic display name if a user modifies the dynamic display name of a field and adds the field to an existing ad hoc web report.
48	68863	JReport Server now shows a loading icon when a new filter is inserted in the Filter panel of Web Report Studio.
49	68880	User can now successfully insert new imported SQL files to a catalog using JReport Catalog API.
50	68880	The column elements under the SQL element now get updated successfully after an SQL file is updated via JReport Catalog API.
51	68925	Enhanced JReport Server performance when flex key CPU licensing is used.
52	68940	Fixed the unknown exception error when linking to a report in Page Report Studio.
53	68964	The summary result is displayed correctly in JReport Designer when creating a library crosstab component that uses a dynamic formula as aggregate field.
54	68964	The dynamic formula which is being used as an aggregate field in a library component of a crosstab now works normally with conditional format when the library component is being viewed on JReport Designer or run in JDashboard.
55	68969	Resolved the 100% CPU utilization problem when there is some typo on the keystore file name that user provides to enable SSL on JReport Server.
56	69054	The page crosstab report which uses table style and has two or more repeated fields no longer is displayed with a part of the field values or additional columns when previewed in JReport Designer.
57	69065	The page header panel which is set to be repeated on each page of a page report can now be displayed correctly on each sheet of the report's exported Excel result.
58	69078	Updated the file welcome.jsp of JReport Server for a typo error.
59	69082	The label of the field added to a web report is now displayed with its dynamic display name by default in Web Report Studio if user has configured a dynamic display name for the field.
60	69082	Resolved the java.lang.reflect.InvocationTargetException error when adding parameter control to a web report in JReport Designer.
61	69082	The standalone JReport Server running on Windows 7 on Internet Explorer 11 no longer gets two headers after the submission of a schedule task.
62	69090	The connection name of an imported SQL file can now get changed accordingly on JReport Designer after changing the name of the same catalog's JDBC connection via Catalog Doctor.
63	69106	Setting property Keep Group Together of a table group in a page report no longer results in a blank report page when running the report in Page Report Studio.
64	69131	Allow users to add new categories or hierarchies to their business views in any catalog on JReport Designer.
65	69133	Resolved the scrollbar and chart tooltips displaying issue when running JDashboard with multiple dashboard tabs via URL.
66	69160	The Query Filter dialog in a web report no longer shows an empty value list if a user has added a filter to this query on JReport Designer and the filter condition is on parameter.
67	69219	Enhanced JReport Server performance when running web reports with conditional formatting.
68	69219	Resolved the NullPointerException error when opening a linked report in another linked report in Web Report Studio.
69	69219	Improved JReport Server performance when opening linked reports in a web report created using user data source (UDS).
70	69219	Improved JReport Server performance when running a web report created with user data source (UDS) and the cached report data in the report is expired.
71	69219	Web report charts can be loaded normally in Web Report Studio which is running in Internet Explorer in case filters has been applied or removed in this report.
72	69219	Proper working of undo actions when two or more sorts have been applied to a web table report in JReport Server.
73	69221	The access privileges error message now pops up normally for web reports running in Google Chrome.
74	69249	Resolved the java.lang.ClassCastException error when calling method catalog.delete (sqlHandle) of Catalog API instance to delete an imported SQL object from a catalog.
75	69273	Dynamic formulas that are created as aggregation can now be used to create summary tables on JReport Designer plus crosstabs which use these dynamic formulas can show aggregation values normally in Web Report Studio.
76	69288	The error pages in JReport Server can now be closed with the Close button.
77	69300	The new line character '\r\n' added in the formula now works normally in the web table report running in Web Report Studio.
78	69365	Table alias no longer gets changed in the where condition of the SQL statement when adding a table in the catalog using JReport Catalog API.
79	69367	Resolved the java.lang.NullPointerException error when visiting JReport Administration page with a JReport Server running on Tomcat 8.
80	69411	Resolved the NullPointerException error when running a page report on a JReport production server.
81	69417	Resolved the NullPointerException error when creating a web crosstab report with the property Push Down Group Query of the JDBC connection being set to true in JReport Designer.
82	69461	The last row (i.e. total/aggregation) of a page crosstab report now gets displayed properly when it shows up in the next page.
83	69467	The schedule task of publishing a report to e-mail now can be submitted successfully using Internet Explorer 11.
84	69472	The content of the error message in the unknown exception for Query Optimizer can be displayed normally in linked reports.
85	69493	Resolved the OutOfMemory issue when sorting fields and group by fields in a page report that has a large amount of data.
86	69513	The filter control of a web report no longer is exported to Excel after user sets the Export to Excel property of the web report filter control to false on JReport Designer.
87	69556	Web reports with dynamic formulas running in Web Report Studio now show consistent result with the same reports in JReport Designer.
88	69593	The error message now shows consistent date format with the user defined one when user enters an invalid date format in JReport User's Enter Parameter Values dialog.
89	69621	The description of binded column in the page header of a page banded report now stays as it is when JReport Designer is upgraded from V12 to V13.
90	69650	Enables user to insert tables to a catalog using method insert(java.lang.String dataSourceName, java.lang.String tableName, java.lang.String catalogName, java.lang. String schemaPattern, java.lang.String tablePattern, int type, java.lang.String character) when the method is simultaneously used on database catalogs and schemas.
91	69678	The suppress property now works normally in a page report in which a formula with function IsNoRecord() is used to control this property.
92	69684	The web report label whose width extends beyond the report page in JReport Server no longer disappears when it is no selected.
93	69705	Creating web reports using the wizard that has been embedded into an IFrame is now smoother.
94	69708	User can no longer run an inactive report via URL or JReport Server API.
95	69729	Rebuilds the installation file of JReport Server 13.1 Update 2 to ensure its successful installation in the console interface.
96	69735	Users now have no problem connecting to MongoDB 3.0 and loading tables on JReport Designer once they have a user role with the privilege of Read.
97	69737	The search function can now work normally in a parameter control in Page Report Studio.
98	69739	JReport Server now displays the business view tree correctly in the Group screen when creating an ad hoc page report with a business view and the first category in this business view has no sub node.
99	69755	Resolved the CSS parsing javascript error in Web Report Studio by preventing business view elements' names from starting with numbers.
100	69769	The parameter value's drop-down control no longer persists when user tries to enter a parameter value manually in the Enter Parameter Values dialog and press the Tab key to move onto the next parameter.
101	69807	Keep the Web Report Wizard window sizable when adding a table component to a web report.
102	69869	Support logarithmic scaling on the Y-axis of all chart objects running in Page Report Studio, Web Report Studio and JDashboard.
103	69892	The security setting for non-admin users now works normally in JReport Server when method getFolderInPath is called in the resource manager.
104	69892	Upgrading JReport Server from V11 Update 1 to V13.1 Update 2 now makes the horizontally aligned icons in Page Report Studio consistent.
105	69892	Fixed the error information getting lost in the Error.log of JReport Server 13.1 Update 2.
106	69894	Resolved the merged cells and extra columns in the Excel result of a page crosstab report when the report is exported to Excel with column format and the width of a certain column in the crosstab has been changed.
107	69938	The web controls are now kept in right position in Page Report Studio when JReport Server is upgraded from V11 Update 1 to V13.
108	69941	Resolved the access denied error when hitting OK button in a web report Save As dialog or hitting Save button in the Bind Data screen of the Web Report Wizard before the contents in both dialogs are loaded.
109	69958	The Reset button in the Enter Parameter Values dialog can now function normally after JReport Server is upgraded from V11 to V13.
110	69997	Solved the OutOfMemory issue caused by the heavy load on a specific JReport cluster node.
111	69999	Support default display sequence of parameters in the parameter form control for cascading parameters in a web report.
112	70037	The date values in a web report now get displayed correctly when the report is previewed or exported in Excel format.
113	70112	Support running JReport Designer and JReport Server using Sun's Java Runtime Environment (JRE) in user's production environment.
114	70127	JReport Server 10 Update 1 now works normally when running on JDK 1.8.
115	70141	The filter control now works successfully for a table which locates in the page header panel of a page report.
116	70156	Resolved the IndexOutOfBoundsException error when running a dashboard component with cached cube.
117	70237	Resolved the NullPointerException error when passing a parameter to a linked report in Web Report Studio.
118	70262	Removed the invalid object name error when adding a table to a catalog in the JDBC connection without specifying a table schema.
119	70312	User now has no problem selecting a folder in the New Schedule dialog with Internet Explorer 11 when accessing JReport Server with a domain name.
120	70335	Recovered the word wrap function for label texts in page crosstab reports.
121	70353	Table detail rows in a library component are displayed correctly when running JDashboard against cube.
122	70369	Improved JReport logging so additional errors and exceptions can be logged when issues occur.
123	70421	Removes the vulnerability in JReport products with updated lib files containing serializer-2.7.2 in makewar.xml file.
124	70441	Removes the java.lang.NullPointerException error when running a bursting report on JReport Server.
125	70451	The version date sorting now works properly in the Download from JReport Server dialog.
126	70492	The query that contains the special field "UserName" can no longer be used to create in-memory cube.
127	70531	Fixed the reports hanging issue when running reports on JReport Server.
128	70552	Removes the javascript error when browsing the JReport Server resource tree or selecting a folder via Internet Explorer with its compatibility version being set to 7.
129	70565	Labels of the scrollbar no longer are overlapped when running a library component of a scrollbar chart on JDashboard.

Known issues

Summary field disappears from design view when deleting table column

If a summary is inserted in the GroupFooterPanel of a table, when you delete a column from the table, or unmerge the cell of the group footer row, sometimes you may find the summary field disappears from design view, however, from the report structure tree in the Report Inspector, you can still find the object. You can reset its coordinate related properties so as to make the summary field display again.

Compiling formulas in JReport under JDK 6 or higher gets warnings

Since generic type is introduced to JDK 6 or higher version, when compiling formulas in JReport under JDK 6 or higher, you may come across warning messages as follows:

- Note: Test.java uses unchecked or unsafe operations.
- Note: Recompile with -Xlint:unchecked for details.

You may see that the situations are logged as Javac errors. However, the warning messages do not prevent Javac from creating .class files and reports can still run correctly.

In this case, you can simply ignore the messages and logged errors.

TOC Browser tree doesn't work well for reports with cached report bursting

When end user runs a report which has been defined with some cached report bursting policy in Page Report Studio, and then opens the TOC Browser, he will see all the groups of the report instead of just the ones he is supposed to view. In addition, no matter if he has access to a group, he cannot click the group name from the TOC tree to get corresponding details. JReport will further enhance the cached report bursting feature in a future release to resolve this issue.

Report data gets cut off in PDF result

When you export a report to PDF format, if the report contains a large amount of data but its page mode was specified to be continuous page mode, or its page size was set to be larger than 200 inches, you will find that in the PDF result some data of the report are cut off. This is because in one PDF page, the data displayed can be no larger than 200 inches.

Starting JReport with 64-bit JDK 1.7

To use 64-bit JDK 1.7 to start JReport on 64-bit Sun Solaris System, you need to modify the launch script JReport.sh (for JReport Designer) or JRServer.sh (for JReport Server) in `$REPORTHOME/bin` by changing the value $JAVAHOME/bin/java to $JAVAHOME/bin/amd64/java.

Deploying monitor.war (servlet.war) and jreport.war in the same domain gets exception

When integrating JReport Server and JReport Server Monitor into one application server, you need to make sure that they are deployed to different domains. JReport Server Monitor is supposed to be just an application for the administrator which doesn't need to come along with a JReport Server as the report system. So it is recommended that JReport Server Monitor be installed on a separate system or on systems for system administrators.

Limitation of going to another group on chart

When you perform the go-to action on a chart which uses a dynamic formula as its shown value, and the formula contains group information, you will get exceptions. This is a limitation in current version. We will resolve the issue in future release.

The zooming in or out of JReport browsers is not supported

All JReport browsers such as Page Report Studio, Web Report Studio, JDashboard and so on do not support being zoomed in or out by the way of scrolling the mouse while holding the Ctrl key. This may result in that the JReport UI cannot display well.

API change

jet.formula.ParamDesc.value has been set to private now. You can use getDisplayValue(DbValue paramvalue, java.util.Locale locale) instead.

For example:

```
ParamDesc desc = new ParamDesc();

... ...

desc.getDisplayValue(desc.getValue(), Locale.getDefault());
```

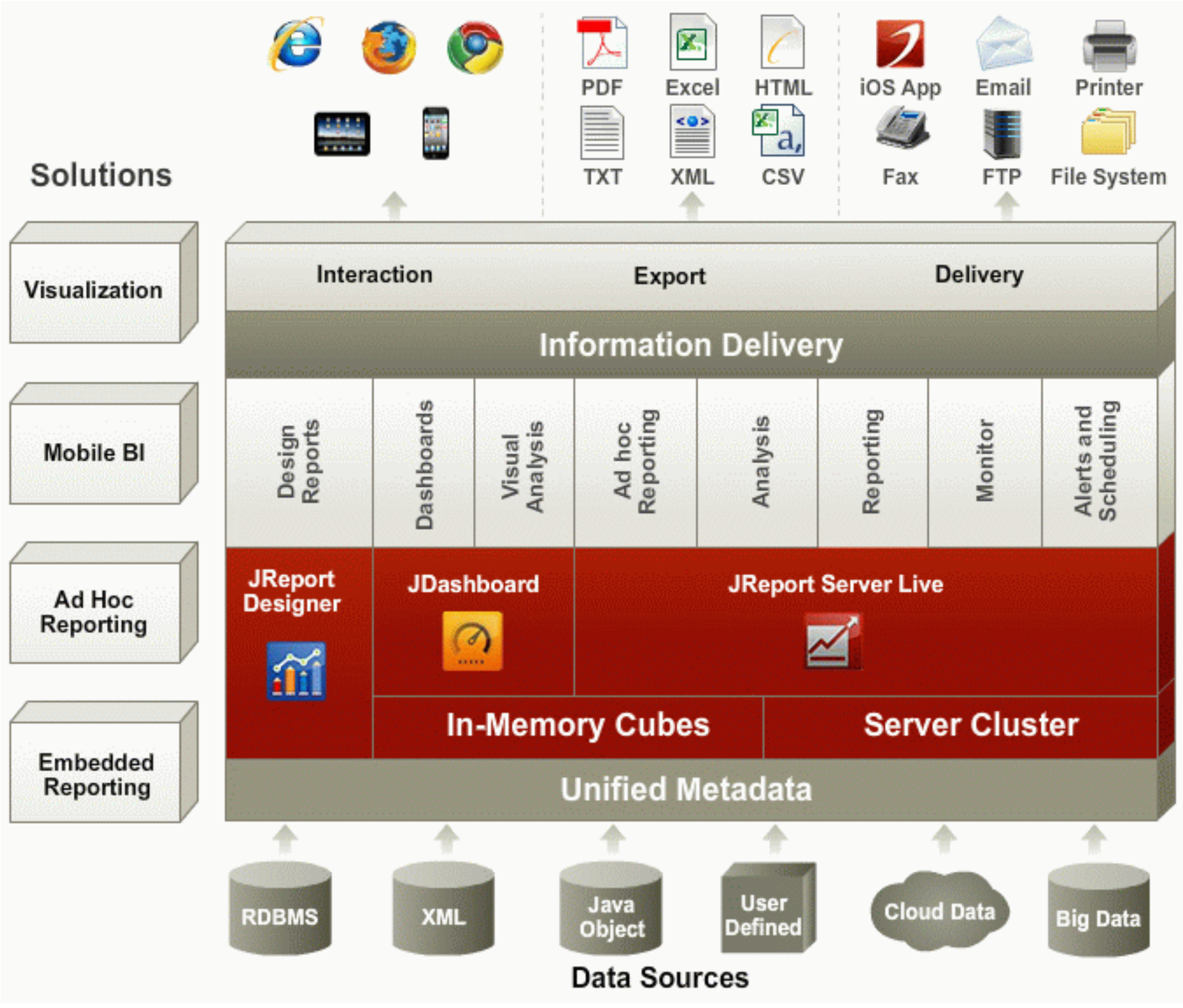
JReport Product Overview

JReport delivers operational business intelligence to enterprise applications through powerful embedded reporting.

JReport is a complete Java reporting solution that provides sophisticated enterprise reporting, ad hoc reporting, and data analysis. A 100% Java EE architecture and a rich set of APIs allow JReport to be seamlessly embedded into any application, providing end users with a transparent interface to easily generate reports, share information, and analyze data. With JReport, any report can be made interactive, extending the "life" of a report by allowing users to easily sort, group, navigate, and filter via the Web. This wide range of functionality, including the ability to drill down on data, enables users to quickly derive value from their business intelligence.

JReport product architecture

JReport's Java architecture takes advantage of the portability, scalability, and ease of integration associated with J2EE technology to provide a powerful, flexible reporting solution that fits perfectly within any architecture.



JReport Designer is a Swing-based Integrated Development Environment (IDE) that enables sophisticated report design and presentation of critical business data. It provides an intuitive interface, reusable report components, flexible layout, and a toolset for designing and testing reports. With JReport Designer, you can build reports using simple drag and drop techniques or by using the Report Wizard. Data can be accessed from any data source to design and preview reports in order to deliver information to end users in the most relevant and intuitive manner. Rapid creation and modification of reports is accomplished by toggling between design mode and view mode where the report will be displayed with the actual dataset. Once report design is complete, the report is published to JReport Server for generation, delivery, and management.

JReport Server is a 100% Java report generation and management tool. It enables efficient management, sharing, scheduling, versioning, and delivery of reports and enables reporting to be integrated into the workflow of any Java application. The high-performance engine can scale to any workload. Report results can be saved to a versioning system, sent to enterprise/workgroup printers, or e-mailed. With JReport, reports can be viewed in any modern enterprise format including Page Report, Web Report, HTML and standard business documents, such as PDF, Excel, and RTF.

JDashboard delivers information using a user portal user interface rather than a report. Users can freely choose the objects they want to display in the dashboard, without having to know how these objects were created, what data sources to use, what styles to set, and so on. A dashboard can hold multiple data components so that when browsing the dashboard users are able to see multiple data aspects. Within a dashboard, data components are able to communicate with each other via the message mechanism. This allows actions such as common filters to be applied to all the components of a dashboard even when coming from different data sources.

Page Report Studio and **Web Report Studio** enable reports to be accessed through a web browser via Dynamic HTML, or AJAX. With Page Report Studio and Web Report Studio, reports can be modified using dynamic filter, sort, and drill capabilities. Using Page Report Studio and Web Report Studio's advanced capabilities, users can drag and drop columns to and from an existing report, dynamically change chart types, pivot crosstabs, add groups, convert report components or create an entirely new report.

Visual Analysis is a WYSIWYG product to visualize the result of every work step. Simply by dragging and dropping data fields onto a layout module, users are able to visually create crosstabs and charts step by step. The use of colors, sizes, shapes, and pie slices demonstrates the data in rich aspects.

Reporting a problem or requesting a feature

If you are having trouble running JReport Designer or encounter any problems during reporting, take the following steps to troubleshoot it yourself first:

1. Check whether the system on which JReport Designer is running meets the system requirements.
2. Check the configuration of your JDBC drivers. JReport Designer supports database connections via various JDBC drivers and ODBC data sources via Sun's JDBC-ODBC bridge.

   Note: You need to install the JDBC or ODBC driver before installing JReport Designer. For more information about how to install the driver please refer to the appropriate JDBC or ODBC manuals.

   For more information, see Setting up the JDBC driver.
3. Check the FAQ in the Customer Care Portal for frequently-asked questions and their solutions.

If the problem persists, report it to Jinfonet Support (support@jinfonet.com) with the following information:

1. Describe the precise steps leading to the problem.
2. Run the batch file called jrenv.bat in the `<install_root>\bin` folder, which will generate a file called report.env in the current directory. Send this file to us.
3. Send us the catalog file (*.cat and*.fml) and the report file *.cls that you are having problem with.
4. Start JReport Designer with the batch file DJReport.bat. Running this batch file will generate the log file in the `<install_root>\logs` directory. Try to reproduce the problem, and send this log file to us.
5. In order to resolve the technical issues that you have reported, we need to access your report data so that we can recreate and analyze the problem. Regardless of the size of your database, we only require access to the data returned by the query of the particular report. We will execute a confidentiality agreement with you if necessary.

   To extract the report data, in the Catalog Browser, right-click on the data object that your report is using, select **Create Cached Query Result**. Input the data file name and click the **Save** button. The query result will be saved in the file. Send us all the files <file name>.* (including the description file also). If the primary report contains subreports, repeat this also for each subreport.

Visiting web pages for more information

- Products information
  http://www.jinfonet.com/products
- News center
  http://www.jinfonet.com/company/news
- Demo center
  http://www.jinfonet.com/resources/jreport-demo
- Products download center
  http://www.jinfonet.com/product/download-jreport
- White papers
  http://www.jinfonet.com/resources/white-papers
- JReport product documentation
  http://www.jinfonet.com/jreport-documentation
- JReport Javadoc
  http://www.jinfonet.com/jreport-documentation
- JReport Technical Support center
  http://www.jinfonet.com/services/technical-support

Installing, Running and Uninstalling

JReport Designer can run on both the Windows and Unix platforms. In this chapter, you will be shown how to install and uninstall JReport Designer in your computer. Also, after reading this chapter, you will have a basic understanding of the launch files in JReport Designer.

- System requirements
- JReport licenses
- Supported report databases
- JReport Designer on Windows
- JReport Designer on Unix
- Silent installation
- Solving installation problems
- JReport Designer launch files
- Command-line startup

System requirements

The following table shows the basic system requirements for installing JReport Designer. Check your system to make sure that all the requirements have been met before installation.

JReport Designer System Requirements		
	Recommended Requirements	**Minimum Requirements**
OS:	Windows x64, Unix x64 and Linux x64	Windows, Unix, Linux
CPU:	Quad Core processor	Dual Core processor
Free Memory:	4 GB	2 GB
Free Disk:	2 GB	1 GB
JDK:	6 or above	6 or above
Browser:	Latest releases	IE 9, Firefox 20, Chrome 23

Reference: Download a JDK version at http://www.oracle.com/technetwork/java/javase/downloads/index.html.

Notes:

- JReport supports Java VMs from Sun and IBM. Other vendor's Java VMs may cause unexpected errors. Jinfonet technical support welcomes any feedback on the use of any other Java VMs.
- You are not recommended to run JReport in the Internet Explorer Compatibility View mode.

JReport licenses

JReport has several add-on licenses which enable specific features:

- JReport Live license
- JReport Server Cluster license
- JDashboard license
- Visual Analysis license
- Organization license

Contact your Jinfonet Software account manager to obtain your required license.

JReport Live license

The JReport Live license enables the use of web reports and ad hoc page reports and all their related functions. JReport Designer and JReport Server have separate live licenses.

Live license for JReport Designer

A JReport Live license for JReport Designer controls the data sources for ad hoc page reports and web reports and the creation of web reports from those data sources on JReport Designer. The Live license allows you to utilize the following features on JReport Designer:

- Create and edit business views. These are meta-data descriptions created in JReport Designer to be used as data sources for ad hoc page reports and web reports. They can be created on any type of data source such as JDBC, ODBC, XML and Web Services.
- Create web reports from business views.
- Export and print web reports.
- The following functions also require that JReport Server has a Live license:
  - Preview web reports in Web Report Studio.
  - Publish web reports to JReport Server.
  - Download reports which use business views as the data source from JReport Server.

Live license for JReport Server

A JReport Live license for JReport Server controls the real time ad hoc and analysis reporting on JReport Server. The Live license allows you to utilize all of the following features and functions on JReport Server:

- Create, view, run, edit, export, print, and publish ad hoc page reports and web reports. The data sources for these reports are created in JReport Designer, which requires JReport Designer has a Live license.
- All actions in Page Report Studio involving business view or changes of report template:
  - Create new page reports or page report tabs
  - Delete report tabs from a page report
  - Add components and data fields into reports
  - Remove components from reports
  - Move and resize components
  - Edit component properties
  - Drill
  - Change chart type
  - Rotate tables and crosstabs
  - Convert between crosstab and chart
  - Create query filters
  - Display the Resource View panel which shows the data resources for the current open report

JReport Server Cluster license

JReport Server Cluster license enables a group of JReport Servers to work together with shared resources, load balancing, and failover in a distributed cluster.

JDashboard license

A JDashboard license enables the use of JDashboard and all related functions. JReport Designer and JReport Server have separate JDashboard licenses.

Since JDashboard requires using business views as data sources, the JReport Live license is also required. JReport Designer and Server have separate Live licenses too.

JDashboard license for JReport Designer

A JDashboard license for JReport Designer allows for creation of library components which are used to build dashboards and publishing of library components to JReport Server.

JDashboard license for JReport Server

A JDashboard license for JReport Server enables the management of library components in the component library, the creation of dashboards using library components, and the use of dashboards.

Visual Analysis license

A Visual Analysis license enables the use of Visual Analysis and all related functions. Since Visual Analysis requires using business view as the data source, the JReport Live license is also needed in order to perform visual analysis.

Organization license

An organization license enables organizing users into different groups with their own administrators.

Supported report databases

JReport supports all of the current mainstream databases as well as most databases which support ODBC or JDBC drivers. Except for traditional databases, databases on the cloud are also supported, such as Vertica, Amazon RDS, and RedShift.

The following table lists the databases and JDBC drivers that have been tested with JReport. If you are using any of the databases listed below, you are recommended to use the corresponding driver version with JReport although any driver which the DBMS supplier recommends is also fine. If you encounter problems when using a database or driver version that is not listed here, you can contact Jinfonet Support (support@jinfonet.com) for help.

You can also refer to the page http://wiki.netbeans.org/DatabasesAndDrivers for additional information on database and driver.

Database	Version	Driver File Name	JDBC Driver	Example URL
MS SQL Server	2014 (12.00.2000)	sqljdbc4.jar	com.microsoft.sqlserver.jdbc.SQLServerDriver	jdbc:sqlserver://<host>:1433;DatabaseName=test
MS SQL Server	2012 (11.00.2100)	sqljdbc4.jar	com.microsoft.sqlserver.jdbc.SQLServerDriver	jdbc:sqlserver://<host>:1433;DatabaseName=test
MS SQL Server	2008R2 (10.50.4000)	sqljdbc4.jar	com.microsoft.sqlserver.jdbc.SQLServerDriver	jdbc:sqlserver://<host>:1433;DatabaseName=test
MS SQL Server	2008R2 (10.50.4000)	Merlia.jar;	com.inet.tds.TdsDriver	jdbc:inetdae7:<host>:1433?database=test
MS SQL Server	2008	sqljdbc4.jar	com.microsoft.sqlserver.jdbc.SQLServerDriver	jdbc:sqlserver://<host>:1433;DatabaseName=test
MS SQL Server	2005	sqljdbc.jar	com.microsoft.sqlserver.jdbc.SQLServerDriver	jdbc:sqlserver://<host>:1433;user=sa;password=1234;database=test
MS SQL Server	2000	msbase.jar; msutil.jar; mssqlserver.jar	com.microsoft.jdbc.sqlserver.SQLServerDriver	jdbc:microsoft:sqlserver://<host>:1433
MS SQL Server	2000	Opta2000.jar	com.inet.tds.TdsDriver	jdbc:inetdae7:<host>:1433?database=test
MS SQL Server	2000	tds-1.0.3.jar	net.sourceforge.jtds.jdbc.Driver	jdbc:jtds:sqlserver://<host>:1433/test
MemSQL	5.5.8	mysql-connector-java-5.1.25-bin.jar; mysql-connector-java-5.1.7-bin.jar	com.mysql.jdbc.Driver	jdbc:mysql://<host>:3306/test
ScaleDB	0.2.3	mysql-connector-java-5.1.25-bin.jar; mysql-connector-java-5.1.7-bin.jar	com.mysql.jdbc.Driver	jdbc:mysql://<host>:3306/test
MySQL	5.5.24 (64bit)	mysql-connector-java-5.1.25-bin.jar; mysql-connector-java-5.1.7-bin.jar	com.mysql.jdbc.Driver	jdbc:mysql://<host>:3306/test
MySql	mysql via SSL	mysql-connector-java-5.1.6-bin.jar	com.mysql.jdbc.Driver	jdbc:mysql://db06:3306/test?p?useSSL=true?clientCertificateKeyStoreUrl= D:\test\SSL_Client\ca-cert.pem?clientCertificateKeyStorePassword=1234
MySql	mysql 5	mysql-connector-java-5.0.4-bin.jar	com.mysql.jdbc.Driver	jdbc:mysql://<host>:3306/test
MySql	mysql-5.0.2-alpha-win	mysql-connector-java-3.1.5-gamma-bin.jar	com.mysql.jdbc.Driver	jdbc:mysql://<host>:3306/test
MySql	mysql-5.0.18-win32	mysql-connector-java-5.0.3-bin.jar	com.mysql.jdbc.Driver	jdbc:mysql://<host>:3306/test
MySql	mysql-4.1.12-win32	mysql-connector-java-3.1.10-bin.jar	com.mysql.jdbc.Driver	jdbc:mysql://<host>:3306/test
MySql	mysql 4	mysql-connector-java-3.0.14-production-bin.jar	com.mysql.jdbc.Driver	jdbc:mysql://<host>:3306/test
Oracle	11.2.0.1.0 (64bit)	classes12.jar; ojdbc14.jar; ojdbc5.jar; ojdbc6.jar	oracle.jdbc.driver.OracleDriver	jdbc:oracle:thin:@<host>:1521:orcl (Oracle JDBC Thin using an SID)
Oracle	11.1.0.6.0 (11g)	JDK1.5: ojdbc5.jar; JDK1.6: ojdbc6.jar; classes12.jar	oracle.jdbc.OracleDriver	jdbc:oracle:thin:@<host>:1521:ora11g
Oracle	11.1.0.6.0 (11g)	JDK1.5: ojdbc5.jar; JDK1.6: ojdbc6.jar; classes12.jar	oracle.jdbc.OracleDriver	jdbc:oracle:thin:@//<host>:1521/ora11gsn (Oracle JDBC Thin using a ServiceName)
Oracle	11.1.0.6.0 (11g)	JDK1.5: ojdbc5.jar; JDK1.6: ojdbc6.jar; classes12.jar	oracle.jdbc.OracleDriver	jdbc:oracle:thin:@ora11gtn (Oracle JDBC Thin using a TNSName)
Oracle	10.1.0.2.0 (10g)	JDK1.2&1.3: classes12.zip; JDK1.4: ojdbc14.jar	oracle.jdbc.driver.OracleDriver	jdbc:oracle:thin:@<host>:1521:ora9i
Oracle	9i	JDK1.2&1.3: classes12.zip; JDK1.4: ojdbc14.jar	oracle.jdbc.driver.OracleDriver	jdbc:oracle:thin:@<host>:1521:ora9i
Oracle	8.1.7.0.0	classes12.zip	oracle.jdbc.driver.OracleDriver	jdbc:oracle:thin:@<host>:1521:userdb
DB2	9.7.200.358	db2jcc4.jar	com.ibm.db2.jcc.DB2Driver	jdbc:db2://<host>:50000/test
DB2	9.7.0.4	db2java.zip; db2jcc.jar; db2jcc_license_cu.jar; db2jcc4.jar; sqlj.zip; sqlj4.zip	com.ibm.db2.jcc.DB2Driver	jdbc:db2://<host>:50000/test
DB2	9.1.0.356	db2java.zip; db2jcc.jar; db2jcc_javax.jar; db2jcc_license_cu.jar; db2policy.jar	com.ibm.db2.jcc.DB2Driver	jdbc:db2://<host>:50000/test
DB2	8.2	Db2jcc.jar; db2jcc_license_cu.jar (Linux, Unix and Windows); db2jcc_license_cisuz.jar (Linux, Unixand Windows, z/OS, OS/390,z/OS, iSeries, and so on)	com.ibm.db2.jcc.DB2Driver	jdbc:db2://<host>:50000/test
DB2	8.2	Db2java.zip	com.ibm.db2.jdbc.app.DB2Driver	jdbc:db2:test
DB2	8.1.9.917	db2java.zip; db2jcc.jar; db2jcc_javax.jar; db2jcc_license_cisuz.jar; db2jcc_license_cu.jar	com.ibm.db2.jcc.DB2Driver	jdbc:db2://<host>:50000/test
DB2	8.1.7.380	db2java.zip; db2jcc.jar; db2jcc_javax.jar; db2jcc_license_cisuz.jar; db2jcc_license_cu.jar	com.ibm.db2.jcc.DB2Driver	jdbc:db2://<host>:50000/test
DB2	8.1	Db2java.zip; db2jcc.jar	com.ibm.db2.jdbc.net.DB2Driver	jdbc:db2://<host>/test
DB2	8.1	Db2java.zip	com.ibm.db2.jdbc.app.DB2Driver	jdbc:db2:test
RedBrick warehouse		redbrick.jar	redbrick.jdbc.RBWDriver	jdbc:rbw:protocol:<host>:5050/test/
Informix	11.70.TC7DE	ifxjdbc.jar	com.informix.jdbc.IfxDriver	jdbc:informix-sqli://<host>:9088/demo:INFORMIXSERVER=ol_informix1170
Informix	11.50.TC7DE	ifxjdbc-g.jar; jdbc.jar	com.informix.jdbc.IfxDriver	jdbc:informix-sqli://<host>:9090/informixdatatype:INFORMIXSERVER=ol_informix
Informix	9.40.TC1E1	ifxjdbc.jar	com.informix.jdbc.IfxDriver	jdbc:informix-sqli://<host>:1527/Demo:INFORMIXSERVER=ol_informix
Informix	9.30	ifxjdbc.jar	com.informix.jdbc.IfxDriver	jdbc:informix-sqli://<host>:1526/stores_demo:informixserver=DBSC
HSQL		hsqldb.jar	org.hsqldb.jdbcDriver	jdbc:hsqldb:D:\JReport\Demo\db\SampleDB
Sybase	12.5.2	jconn2.jar	com.sybase.jdbc2.jdbc.SybDriver	jdbc:sybase:Tds:<host>:5000/master
Sybase	12.5	jconn2.jar	com.sybase.jdbc2.jdbc.SybDriver	jdbc:sybase:Tds:<host>:5000/master
Sybase	12.5	jconn3d.jar	com.sybase.jdbc3.jdbc.SybDriver	jdbc:sybase:Tds:<host>:5000/master
Sybase	11.5	jconn2.jar	com.sybase.jdbc2.jdbc.SybDriver	jdbc:sybase:Tds:<host>:5000/master
Sybase IQ	15.4.0.3019	jconn3.jar; jconn4.jar	com.sybase.jdbc3.jdbc.SybDriver	jdbc:sybase:Tds:<host>:2638/iqdemo
PostGre SQL	8.3.0	postgresql-8.3-607.jdbc2.jar	org.postgresql.Driver	jdbc:postgresql://<host>:5432/postgres
PostGre SQL	8.2.13	postgresql-8.2-506.jdbc3.jar	org.postgresql.Driver	jdbc:postgresql://<host>:5432/postgres
PostGre SQL	8.0	postgresql-8.0-310.jdbc3.jar	org.postgresql.Driver	jdbc:postgresql://<host>:5432/test
Cache	Cache 4	CacheDB.jar	com.intersys.jdbc.CacheDriver	jdbc:Cache://<host>:1972/samples
Derby	10.8.1.2	derby.jar; derbyclient.jar	org.apache.derby.jdbc.ClientDriver	jdbc:derby://<host>:1528/test
Derby	10.5.3.0	derby.jar; derbyclient.jar	org.apache.derby.jdbc.ClientDriver	jdbc:derby://<host>:1527/test
Derby	10.5.1.1	derby.jar	org.apache.derby.jdbc.EmbeddedDriver	jdbc:derby:D:\derby\demo\databases\toursdb
MongoDB	2.2.2	already within JReport	toolkit.db.mongo.MongoDriver	host, port, databaseName:demo
HIVE	0.10.0	hadoop-common-2.0.0-cdh4.1.1.jar; hadoop-core-2.0.0-mr1-cdh4.1.1.jar; hive-exec-0.9.0-cdh4.1.1.jar; hive-jdbc-0.9.0-cdh4.1.1.jar; hive-metastore-0.9.0-cdh4.1.1.jar; hive-service-0.9.0-cdh4.1.1.jar; libfb303-0.7.0.jar; libthrift-0.7.0.jar; slf4j-api-1.6.1.jar; slf4j-simple-1.6.1.jar	org.apache.hadoop.hive.jdbc.HiveDriver	jdbc:hive://<host>:10000
PSQL	V11 SP3	jpscs.jar; pvjdbc2.dll; pvjdbc2.jar; pvjdbc2x.jar	com.pervasive.jdbc.v2.Driver	jdbc:pervasive://<host>:1583/test
Vertica		vertica-jdk5-6.1.2-0.jar	com.vertica.jdbc.Driver	jdbc:vertica://<host>:5433/vmartdb
Amazon RDS		mysql-connector-java-5.0.4-bin.jar	com.mysql.jdbc.Driver	jdbc:mysql://jrdbtest.c4fb8hiicidz.us-west-2.rds.amazonaws.com:3306/sampledb1110 (The URL is dynamically generated when you apply an instance.)
RedShift		postgresql-8.4-703.jdbc4.jar	org.postgresql.Driver	jdbc:postgresql://jinfonet-rsdw-demo.cfcn5ogc14yr.us-east-1.redshift.amazonaws.com:5439/sampledb (The URL is dynamically generated when you apply an instance.)
Hortonworks	0.14.0	hive-jdbc-0.14.0-standalone.jar; hadoop-common-2.6.0.jar	org.apache.hive.jdbc.HiveDriver	jdbc:hive2://<host>:10000/default;auth=noSasl

Notes:

- If you want to use the DB2 app connection, you need to install the client and configure the net address first.
- The database MySql with the example URL `jdbc:mysql://db06:3306/test?p?useSSL=true?clientCertificateKeyStoreUrl=D:\test\SSL_Client\ca-cert.pem?clientCertificateKeyStorePassword=1234` is connected by SSL. For how to install SSL, refer to http://www.openssl.org. For how to configure MySql for SSL, refer to http://dev.mysql.com/doc/refman/5.1/en/ssl-connections.html. For more information about JDBC driver, refer to http://dev.mysql.com/doc/connector-j/en/connector-j-reference-configuration-properties.html.
- For Oracle JDBC Thin using a TNSName, you need to configure the tnsname first, then add the parameter `-Doracle.net.tns_admin=<designer_install_root>\lib` into the file JReport.bat which is located in `<designer_install_root>\bin`, assuming you have copied the file tnsnames.ora from your Oracle server to `<designer_install_root>\lib`.

JReport Designer on Windows

Installing JReport Designer on Windows

Follow the steps listed below to install JReport Designer on Windows:

1. Download the JReport Designer installation file for Windows from the Jinfonet download center: http://www.jinfonet.com/downloadjreport/.
2. Run the installation file and follow the default prompts to install.

During the installation, pay attention to the following:

- The Installation Wizard will first find a JVM to get started. If no JVM is found, the JReport installer will fail to launch. To resolve this issue, you can try either way:
  - Set JAVA_HOME variable and append `%JAVA_HOME%\bin` to Path variable in system environment.
  - Install JReport from a DOS command by specifying the `LAX_VM` option for the Installation Wizard as follows:

    `jrdesigner-xxx-windows.exe LAX_VM "C:\jdk1.6.0_17\bin\java.exe"` (change jrdesigner-xxx-windows.exe to the real file name of the installation file)

    The JDK path should use absolute path and be quoted by "".
- You are recommended to use JDK 6 or higher versions.
- The installer allows you to add additional class paths. Also, after installation, you can choose to add them manually into the setenv.bat file in `<install_root>\bin`.
- If you select to install JReport Designer into a folder that already contains an existing copy, the installer will replace the packages and create new batch/script files. Meanwhile, a copy of the old batch/shell files will be kept for your reference. You are recommended to use the batch/script files that come with the installer to ensure that all new packages have been added into the class path.

Running JReport Designer on Windows

Use one of the following ways to run JReport Designer on Windows:

- Double-click the **JReport Designer 13.5 Update 2** shortcut on your desktop.
- Click **Start** > **All Programs** > **JReport 13.5 Update 2** > **Report Designer**.
- Run the JReport.bat/JReport.sh file located in `<install_root>\bin`.
- Run the startup batch file from a MS-DOS command prompt. For example, assume that JReport Designer has been installed in `C:\JReport\Designer`, you can type the following commands:

```
C:\>cd JReport\Designer\bin
C:\JReport\Designer\bin>jreport.bat
```

Uninstalling JReport Designer on Windows

Use either of the following ways to uninstall JReport Designer on Windows:

- From the Control Panel, open **Add or Remove Programs**, and then select **JReport Designer 13.5 Update 2** to remove it.
- Run uninstaller.exe in `<install_root>\_uninst`.

Note: The uninstaller removes all files that were generated by the installer, while any files that were created later by the program are retained. These files should be removed manually.

JReport Designer on Unix

Installing JReport Designer on Unix

To install JReport Designer on Unix, follow the steps below:

1. Download the JReport Designer installation file for Unix from the Jinfonet download center: http://www.jinfonet.com/downloadjreport/.
2. Click the installation file to launch the Installation Wizard. Alternatively, you can open a console window, and change the directory to the location of the file. The following are examples of the commands that can be used.

   `$ cd thepath`

   To make the installation file executable, type the command:

   `$ chmod +x jrdesigner-xxx-linux.bin` (change jrdesigner-xxx-linux.bin to the real file name of the installation file)

   To run the installation file:

   `$ ./jrdesigner-xxx-linux.bin` (change jrdesigner-xxx-linux.bin to the real file name of the installation file)
3. Once the Installation Wizard has been successfully loaded, you can then follow the default prompt to install JReport Designer.

During the installation, pay attention to the following:

- The Installation Wizard will first find a JVM to get started. If no JVM is found, the JReport installer will fail to launch. To resolve this issue, you can try either way:
  - Set JAVA_HOME variable and append `$JAVA_HOME/bin` to PATH variable in system environment.
  - Specify a JVM for Installation Wizard with the option `LAX_VM` as follows:

    `$ ./jrdesigner-xxx-linux.bin LAX_VM "/home/jdk1.6.0_17/bin/java"` (change jrdesigner-xxx-linux.bin to the real file name of the installation file)

    The JDK path should use absolute path and be quoted by "".
- You are recommended to use JDK 6 or higher versions.

Running JReport Designer on Unix

To run JReport Designer on Unix, run the script file JReport.sh in `<install_root>/bin`. You can modify this script file by adding additional classes before launching JReport Designer.

`$ ./JReport.sh`

Uninstalling JReport Designer on Unix

To uninstall JReport Designer on Unix, execute uninstaller.sh in `<install_root>/_uninst`.

Alternatively, you can open a console window, and change the directory to the location of the uninstaller.sh file. The following are examples of the commands that can be used:

`$ cd thepath`

Run the uninstaller file:

`$ ./uninstaller`

Silent installation

JReport provides a file for installing JReport Designer silently without user participation in the installation process, both on Windows and Unix platforms.

Follow the steps below to install JReport Designer silently:

1. Copy the file DesignerInstall.properties from `<install_root>\help\samples\SilentInstall` to the directory where the JReport Designer installation file is saved and edit it to suit your requirements.

2. Run the following command, and JReport Designer will be installed in the designated path:

   `$ ./jrdesigner-xxx-linux.bin -i silent -f DesignerInstall.properties`

   Or

   `$ ./jrdesigner-xxx-linux.bin LAX_VM "USER_JAVA_HOME/bin/java" -i silent -f DesignerInstall.properties` (recommended)

   Change jrdesigner-xxx-linux.bin to the real file name of the installation file.

Notes:

- When installing JReport Designer silently, make sure you do not use overwrite installation, instead, install it to a new directory.

- When you install the Update or Service Pack silently, edit the file update.properties in `<install_root>\help\samples\SilentInstall` to your own requirements. This file is used to create an option file (that is, response file) for the Installation Wizard. It predefines all the information that is required for the installation.

  You can also create a property file and save it as follows:

  ```
  USER_INSTALL_DIR=/usr/local/JReport/Designer
  USER_KEY=UID
  USER_PASSWORD=Password
  ```

  Modify the above lines according to your own environment and configurations.

Solving installation problems

This section is trying to help you solve the problems you encounter during the installation.

Where to find log information

If error occurs during the installation, you can check the log information recorded to find out what the problem is. Where logs are generated depends on when the installation process get stuck:

- If the installation is cancelled before you click the Install button on the installation wizard, logs are created on the desktop for Windows and in the userhome directory for Unix/Linux.
- If the installation is cancelled after you click the Install button on the installation wizard, logs are created in the logs folder in the installation root directory.

Besides, on a Windows platform, you can choose to specify the log destination that should use absolute path and log file name when launching the installation wizard by running the following command:

```
jrdesigner-xxx-windows.exe -D$INSTALL_LOG_NAME$="Install.log" -D$INSTALL_LOG_DESTINATION
$="D:\temp"
```

or

```
$ ./jrdesigner-xxx-linux.bin -D$INSTALL_LOG_NAME$="Install.log" -D
$INSTALL_LOG_DESTINATION$="/opt/temp"
```

Change jrdesigner-xxx-windows.exe or jrdesigner-xxx-linux.bin to the real file name of the installation file.

Feel free to send your questions to support@jinfonet.com.

An issue on Windows Vista

Problem

When running JReport Designer's installer on Windows Vista, the installer cannot find the browser and the installed JDKs, or when running installers of other JReport products, the installer cannot find the installed JDKs. These browsers are supported in JReport Designer's installer: Internet Explorer, Firefox, and Google Chrome.

Reason

By default Vista's security settings are stricter than Windows 2000, XP, and 2003. The children processes do not inherit the execution right from their parent process.

Solution

Make the compatibility property of the installer file (.exe) available:

1. Right-click the installer file, and select **Properties** from the shortcut menu.
2. In the Compatibility mode panel of the Compatibility tab, check the **Run this program in compatibility mode for** option, and then select **Windows 2000** from the drop-down list.
3. Click **OK**.
4. Run the installer.

JReport Designer launch files

You can find a set of utilities in the directory `<install_root>\bin`. All of these batch files (Windows)/ script files (Unix) can be edited to suit different circumstances. However, you need to make sure of their functions before doing this. In addition, the information in this topic is current at the time of release, however, due to changes that take place during the release cycle it may no longer be valid. The most current information can be found in the Readme.html file with each version in the path `<install_root>`.

CatDr.bat/CatDr.sh

This tool is for starting the Catalog Doctor in order to make changes to the catalog.

Usage

```
java <-Dreporthome=reporthome> jet.builder.JReport [-options] [reportfile]
```

Options

- **-help**
  Prints out this message.
- **-vDebug**
  Enables output debug message.
- **-vError**
  Enables output error message.
- **-log[:<file>]**
  Outputs message to `.\<file>` or the default.

DJReport.bat/DJReport.sh

This tool is for starting JReport Designer in logging mode. In case of problems, you are requested to launch JReport Designer with this batch file/script file in order to track detailed log information. Running this batch file/script file generates log file in the `<install_root>\logs` directory.

Usage

```
java <-Dreporthome=reporthome> com.jinfonet.designer.JReport [-options][reportfile]
```

Options

- **-help**
  Prints out this message.
- **-vDebug**
  Enables output debug message.
- **-vError**
  Enables output error message.

EConvert.bat/EConvert.sh

This tool is for removing the evaluation mark from reports. If you have purchased the JReport product and is successful in running JReport Designer with the new key, but the evaluation watermarks are still left, you will need to run this batch file to remove the evaluation watermarks. Run the batch file with the report names to be converted with the full path as parameter.

Usage

```
java -Dreporthome=<drive>:<path> jet.report.EvlConverter
[[drive:][path][filename] +]
[drive:][path][filename]
```

Econvert.bat [ReportName.cls]

Example: `Econvert.bat C:\JReport\Designer\Demo\Reports\TutorialReports\*.cls`

jrenv.bat/jrenv.sh

This tool is for generating the report environment file report.env in the current directory. This file can help the Jinfonet support staff assist you when you run into problems.

JReport.bat/JReport.sh

This tool is for starting JReport Designer.

Usage

```
java <-Dreporthome=reporthome> com.jinfonet.designer.JReport [-options][reportfile]
```

Options

- **-help**
  Prints out this message.
- **-vDebug**
  Enables output debug message.
- **-vError**
  Enables output error message.

Note: You may need to set an appropriate -Dfile.encoding option in the file to start JReport Designer in order to view characters correctly.

JRTutorial.bat/JRTutorial.sh

This tool is for running this batch file to display the tutorial help.

ModelWizard.bat/ModelWizard.sh

This tool is for importing ObjectDataSource.

Usage

```
java com.jinfonet.jdbc.model.wizard.ModelWizard [-options]
```

Options

- **-help**
  Prints out this message.
- **-vDebug**
  Enables output debug message.
- **-vError**
  Enables output error message.
- **-log[:<file>]**
  Outputs message to JGUIEditor.log or `.\<file>`.

NJReport.bat/NJReport.sh

This tool is for starting JReport Designer without the command window.

Usage

```
java <-Dreporthome=reporthome> com.jinfonet.designer.JReport [-options][reportfile]
```

Options

- **-help**
  Prints out this message.
- **-vDebug**
  Enables output debug message.
- **-vError**
  Enables output error message.

Note: You may need to set an appropriate -Dfile.encoding option in the file to start JReport Designer in order to view characters correctly.

PropConvert.bat/PropConvert.sh

This tool is for converting property values. Running this batch file will display a window from which you can modify many properties of a selected report file. This is most useful when you want to modify several properties using the same value at the same time. For example, if a Label and several DBFields share the same background of blue, and now you want to change them all to transparent, you can simply use this tool.

rp.bat/rp.sh

This tool is for replacing user ID and license key.

Usage

```
rp UID Key
```

rptconv.bat/rptconv.sh

This tool is for converting old report schema to be current version.

Usage

```
rptconv "-source=source_path" ["-target=destination_path"] [-r] [-s]
```

Options

- **-source**
  Specifies the source path of the reports that are to be converted.
- **-target**
  Specifies the destination path for the converted reports.
- **-r**
  Replaces the source report with the converted version.

  If this option is set, ["-target=destination_path"] will be ignored.

  If both "-r" and "-target" are not specified, the converted reports are saved in the same directory as the source reports and named as "converted_SourceReportName".
- **-s**
  Converts all the reports in the specified directory, including the reports in all subdirectories.

Examples

- **To convert a single report:**

  `rptconv "-source=C:\JReport\Designer\demo\reports\SampleReports\InvoiceReport.cls" "-target=C:\temp"`

  This will convert C:\JReport\Designer\demo\reports\SampleReports\InvoiceReport.cls to C:\temp \InvoiceReport.cls.

  `rptconv "-source=C:\JReport\Designer\demo\reports\SampleReports\InvoiceReport.cls" "-target=C:\temp\i.cls.xml"`

  This will convert C:\JReport\Designer\demo\reports\SampleReports\InvoiceReport.cls, save the converted report to `C:\temp`, and name it as "1.cls.xml" (if license allows).

  `rptconv "-source=C:\JReport\Designer\demo\reports\SampleReports\InvoiceReport.cls"`

  This will convert C:\JReport\Designer\demo\reports\SampleReports\InvoiceReport.cls, save the converted report in the same directory, and name it as "converted_InvoiceReport.cls".

  `rptconv "-source=C:\JReport\Designer\demo\reports\SampleReports\InvoiceReport.cls" -r`

  This will overwrite C:\JReport\Designer\demo\reports\SampleReports\InvoiceReport.cls.

- **To convert all reports (*.cls, *.rpt, *.clx, *.cls.xml) in a directory:**

  `rptconv "-source=C:\JReport\Designer\demo\reports" "-target=C:\temp"`

  This will convert all the reports in `C:\JReport\Designer\demo\reports` and save the converted reports to `C:\temp`. The converted reports use the same file names as source reports.

  `rptconv "-source=C:\JReport\Designer\demo\reports" "-target=C:\temp" -s`

  This will convert all the reports in `C:\JReport\Designer\demo\reports` and in the subdirectories and save the converted reports to `C:\temp`. The converted reports take the same file names and directory structure as source reports.

  `rptconv "-source=C:\JReport\Designer\demo\reports" "-target=C:\temp\*.cls" -s`

  This will convert all the reports in `C:\JReport\Designer\demo\reports` and in the subdirectories and save the converted reports to `C:\temp`. The converted reports take the same directory structure as source reports and the suffixes of their file names are all changed to ".cls".

  `rptconv "-source=C:\JReport\Designer\demo\reports" -r -s`

  This will convert all the reports in `C:\JReport\Designer\demo\reports` and in the subdirectories. The converted reports overwrite the source reports.

  `rptconv "-source=C:\JReport\Designer\demo\reports"`

  This will convert all the reports in `C:\JReport\Designer\demo\reports`. All the converted reports are saved in the same directory and named as "converted_SourceReportName".

- **To convert a type of reports with same suffixes in a directory:**

  The usage is similar to converting a directory. You can specify the wildcard to filter reports, for example:

  `rptconv "-source=C:\JReport\Designer\demo\reports\SampleReports\*.cls" "-target=C:\temp"`

  This will convert all the reports with the suffix ".cls" in `C:\JReport\Designer\demo\reports \SampleReports` and save the converted reports to `C:\temp`.

Notes:

- There must be one and only one catalog file in the directory where the reports to be converted reside.
- If the reports to be converted contain UDO or UDF, make sure the corresponding classes or jars are included in the class path of rptconv.bat/rptconv.sh.

setenv.bat/setenv.sh

This tool is for generating the report environment variables before starting JReport Designer.

Command-line startup

You can launch JReport Designer from the Command Prompt window. This method is optional on some platforms (Windows for example) and preferred on others. Using command-line startup enables you to use the command-line syntax options.

java -Dreporthome=path **-classpath** path **jet.builder.JReport -vError -vDebug -help**

The command line arguments for JReport Designer are described below. Note that the setup process creates several startup batch files which may contain additional parameters.

- **-Dreporthome**
  Specifies the directory where JReport Designer has been installed. This parameter is required. When you set the report home, upon launching, JReport will try to find jslc.dat and report.ini files in `<install_root>\bin` and check whether they are valid. jslc.dat is the License control file. If you open report.ini, you will find configuration information, including the temp path, template path and the help path. JReport uses the temp path for exporting temporary files, so you should make sure that the temp folder specified in report.ini actually exists.

  Assume that JReport Designer has been installed in `C:\JReport\Designer`, this parameter is `-Dreporthome=C:\JReport\Designer`.

- **-classpath** path
  Tells the Java interpreter the class path. It is usually used for appending the JReport Designer lib path to the Java class path. For example, assume that JReport Designer has been installed in `C:\JReport\Designer` and Java (JDK1.x.x) has been installed in `C:\java`, this parameter is then:

  `-classpath C:\java\lib\classes.zip;C:\jreport\designer\lib\report.jar;`

- **-Djava.compiler**=NONE
  Turns off just in time compiling, which sometimes creates problems.

- **-Djrpt.outer**=true
  Enables JReport Engine to generate an SQL statement with full outer join syntax instead of with abbreviated syntax.

- **-Xmx200m**
  Sets the maximum Java heap size.

- **Logging and debugging switches**
  You can use the following two command switches together to track detail debug and error information.

  - **-vError** - Enables JReport Designer to output error messages. If you start Designer with this command in the command-line, correspondingly in the LogConfig.properties file, value of the engine Error level will be set to ERROR and value of the engine Trace level will be set to OFF.

  - **-vDebug** - Enables JReport Designer to output information and warning messages. If you start Designer with this command in the command-line, correspondingly in the LogConfig.properties file, value of the engine Error level will be set to WARN and value of the engine Trace level will be set to INFO.

  You can specify the log file format, such as Text, or HTML. You can then decide whether or not to append the log files to an existing file, or to create a new file. You can even specify the log layout using the abundant options that JReport provides. To do this, you need to edit the file LogConfig.properties in `<install_root>\bin`. For detailed information on how to configure logging and debugging information, see the LogConfig.properites file.

- **-help**
  Prints the help message.

Upgrading from a Previous Version

Reports created in JReport Designer 7 or prior must be converted to Version 13.5 before they can be modified with JReport Designer 13.5. The conversion procedure is described below. However, JReport Version 7 or prior reports can continue to be published to, and run by JReport Server 13.5 without any modifications.

To upgrade Version 7.1 or prior reports to Version 13.5:

1. Be sure that JReport Designer 13.5 has been installed in a folder that does not contain any previous versions of JReport Designer. If it does, uninstall and reinstall to a new directory.
2. Convert a set of reports by either of following methods:
   - Run the conversion tool rptconv.bat/rptconv.sh which is located in `<install_root>\bin`.
   - Convert a report by opening it directly and then saving it with JReport Designer 13.5.

A Version 7 or prior report is converted to a flow layout report with a banded object. A banded object has the same report sections that a Version 7 or prior report template has, however in Version 13.5 the sections are called panels. Any other data objects in the Version 7 or prior report are converted to the corresponding data containers within the banded object.

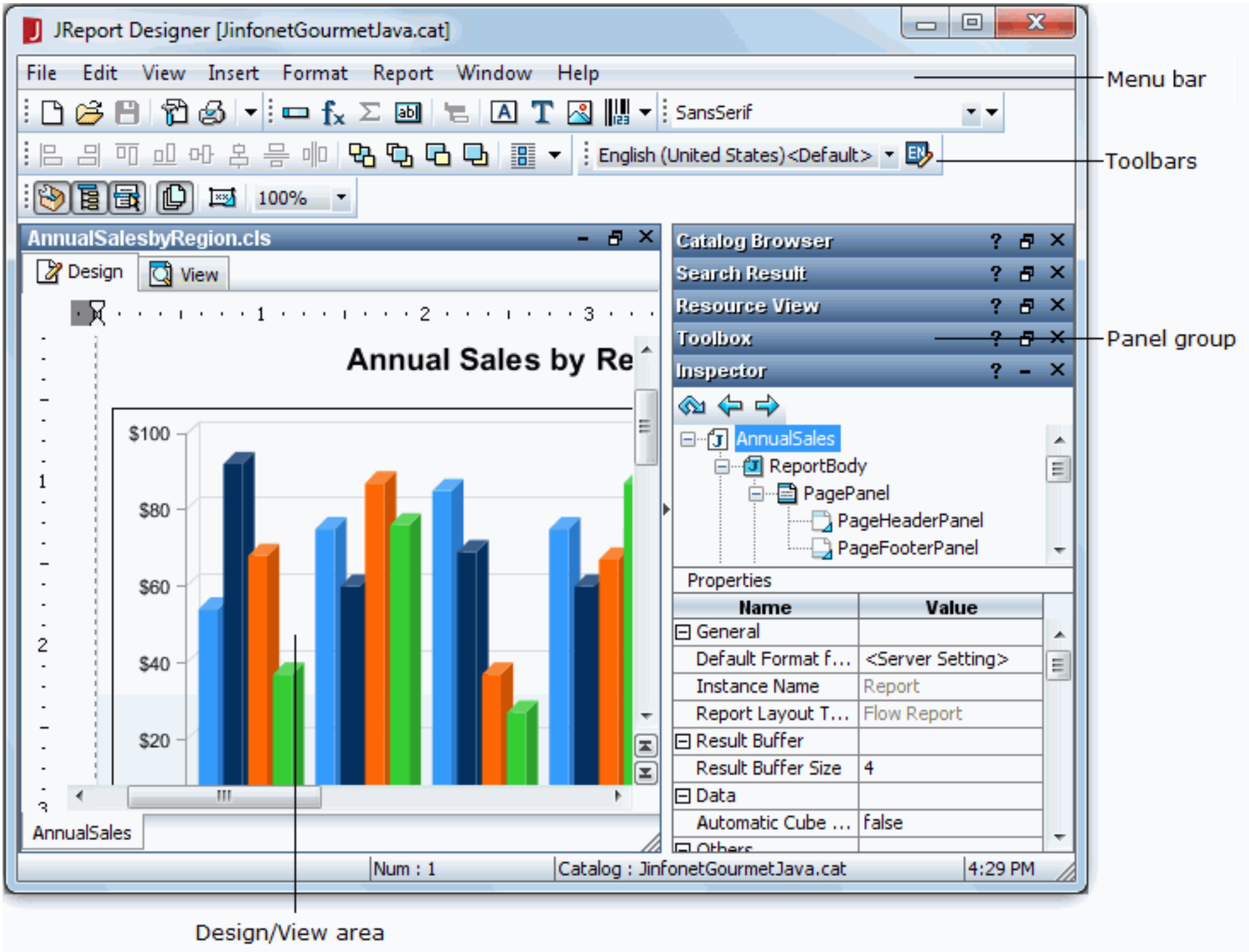
All converted reports can be used in the same way as a new Version 13.5 report that contains a banded object. You can insert other components into the banded panels as described in Component placement. In addition, the features of Version 13.5, such as page mode, displays types, CSS styles and so on can be applied to the converted reports. However, in the converted reports, the layout cannot extend beyond the banded object as it can for a new report created in Version 13.5. That is, while you can insert components into the banded object created as a result of the migration, you cannot insert components after the banded object. If your reports require this functionality, you will need to re-create the report as a new report in Version 13.5.

Note: Migrated reports need to be analyzed to determine how Version 13.5 functionality can best be applied. For example, converted reports that share a dataset should be combined into page reports to take full advantage of Version 13.5 performance improvements. Contact Jinfonet Professional Services for assistance if desired.

Report Development Environment

This chapter focuses on the layout of the JReport Designer reporting environment, and the functions of each component of JReport Designer. For simple operations, you have been given a brief how-to introduction. Other complex commands associated with this topic have been covered in detail in the specific feature's chapters and sections.

The JReport development environment consists of the parts as illustrated in the following diagram.



- Menu bar
- Toolbars
- Resource View panel
- Inspector panel
- Toolbox panel
- Catalog Browser panel
- Search Result panel
- Design/View area

Menu bar

The following menus are available on the menu bar. Click the links to get details.

- File menu
- Edit menu
- View menu
- Insert menu
- Format menu
- Report menu
- Window menu
- Help menu

File menu

The File menu contains the following items:

New

- **Page Report**
  Creates a page report. For details, see Creating a page report.
- **Page Report Tab**
  Creates a report tab in the current page report. For details, see Creating report tabs in a page report.
- **Web Report**
  Creates a blank web report. For details, see Creating a web report.
- **Library Component**
  Creates a library component. For details, see Creating a library component.
- **Query**
  Creates a query. For details, see Creating a query.
- **Business View**
  Creates a business view. For details, see Adding a business view.
- **Data Source**
  Creates a data source. For details, see Data Source Connections.

Open

Brings up the Open Report dialog to open a report.

Open Recent

Lists recently accessed files for you to open.

Close

Closes the active report.

Close All

Closes all open reports.

Save

Saves the active report.

Save As

Saves the active report with a different name in the current directory.

Save To

Saves the active report with a different name to another directory. This command will also merge the report's resources to the other directory by either creating a new .cat file (if there is no .cat file in that directory), or merging with the existing .cat file (if a .cat file already exists in that directory).

For details about saving reports in JReport Designer, see Saving a page report.

Catalog Management

- **New Catalog**
  Creates a catalog file. For details, see Creating a catalog.
- **Open Catalog**
  Opens a catalog file. For details, see Opening a catalog.
- **Save Catalog**
  Saves the open catalog file.
- **Save Catalog As**
  Saves the open catalog file with a different name to another directory. For details, see Saving a catalog.
- **Export Security**
  Exports the security of the open catalog into a security file. For details, see Exporting catalog security into a security file.

Import From

- **JReport Page Report**
  Imports report tabs to a page report from another page report in the same catalog. For details, see Importing report tabs into a page report.

Export To

Exports the current report result to JReport Result, HTML, PDF, Excel, Text, RTF, XML, PostScript, Mail, Fax, or Applet format. For details, see Exporting report results.

Publish and Download

- **Publish to Server**
  Publishes resources from their original location to JReport Server. For details, see Publishing resources remotely.
- **Publish to Local Directory**
  Publishes resources from their original location to a local directory. For details, see Publishing resources locally.
- **Download from Server**
  Downloads resources on JReport Server to a local directory. For details, see Downloading resources from JReport Server.

Options

Opens the Options dialog to configure JReport to suit your requirements.

Information Bus

Opens the Information Bus dialog to edit the Information Bus content saved with user information in the current catalog.

Page Setup

Opens the Page Setup dialog to set up page properties.

Print Preview

Switches to the view mode for you the preview the report result.

Print

Prints the current report result. For details, see Printing report results.

Exit

Closes JReport Designer. If there are any files not saved, you will be prompted to save them.

Edit menu

The Edit menu contains the following commands:

Undo

Reverses a previous action.

Redo

The counter-operation of Undo.

Cut

Erases an object and places it in the clipboard.

Copy

Copies an object and places it in the clipboard.

Paste

Takes an object from the clipboard and places it into the report.

Delete

Erases an object from the report.

Select All

Selects all the objects in the report.

Find DBField

Finds a specific DBField in the report.

Parameter Order

Sets the parameter order. For details, see Editing the display sequence of parameters.

Freeze Field in Table

If checked, the fields in a table will be stopped from accepting any mouse or keyboard events.

Merge

Merges two or more cells in a table or tabular.

NLS Editor

Shows the NLS Editor for you to edit NLS file. For details, see NLS Editor.

CSS Editor

Shows the CSS Editor for you to manage CSS styles. For details, see Managing CSS styles.

Bind Data

Spesifies to bind a data source to the current web report or library component. For details, see Using formulas to control showing or hiding components.

View menu

The View menu contains the following commands:

Preview As

Previews the report result in various formats. For details, see Previewing a page report.

Toolbar

Specifies whether or not to show the specified toolbars.

- **Customize**
  Opens the Customize Toolbars dialog to customize the toolbars.

Resource View

Specifies whether or not to show the Resource View panel.

Report Inspector

Specifies whether or not to show the Inspector panel.

Catalog Browser

Specifies whether or not to show the Catalog Browser panel.

Toolbox

Specifies whether or not to show the Toolbox panel.

Status Bar

Specifies whether or not to show the status bar.

Page Layout

Specifies whether or not to show the report in page layout style, which displays the page margins and enables multiple pages to be continuous.

Page Header

Specifies whether or not to show the page header.

Page Footer

Specifies whether or not to show the page footer.

Rulers

Specifies whether or not to show the vertical and horizontal rulers.

Guidelines

Specifies whether or not to show guidelines.

Margin

Specifies whether or not to show page margins.

Grids

Specifies whether or not to show grids.

Snap To Grids

Specifies whether or not to snap an object to grids when you move it by dragging and dropping. If checked, aligning objects will be made easy.

Grid Size

Opens the Set Grids dialog to set grid settings.

Display Attributes

Opens the Display Attribute dialog to specify the precision of fractional data, the unit of measurement in the report, and the resolution.

Show Hidden Objects

Specifies whether or not to show invisible objects in design mode. If you have checked this item, all hidden objects (with the Invisible property value being true) will be shown in design mode however marked with two triangles.

High Precision View

Specifies whether to display reports in high precision when shifting from the design mode to the view mode. If unchecked, low precision will be used. For detailed information about high/low precision, see Precision Sensitive.

By default, this option is enabled. You can also set its default value in the Options dialog (**File** > **Options** > **General** > **High Precision View**).

Data Buffer Information

Opens the Data Buffer Information dialog to view the information of data buffer in the current report and its subreports.

View Mode Language

Sets the language for previewing the report result in view mode.

Insert menu

The Insert menu contains the following commands:

DBField

Inserts a DBField into a report. For details, see Inserting a DBField in a report.

Formula

Inserts a formula field into a report. For details, see Inserting a formula in a report.

Summary

Inserts a summary field into a report. For details, see Inserting a summary field in a report.

Parameter

Inserts a parameter field into a report. For details, see Inserting a parameter field in a report.

Group

Creates and edits the criteria by which the fields of certain components are grouped and sorted. For details, see Grouping the data.

Label

Inserts and creates labels into a report.

Text Box

Places and creates a text box into a report.

Image

Places a BMP, JPEG, or GIF image into a report.

Barcode

Inserts a barcode into a report. For details, see Barcodes.

Rank

Inserts a rank object into a report. For details, see Ranks.

Table

Inserts a table into a report. For details, see Inserting a table.

Crosstab

Inserts a crosstab into a report. For details, see Inserting a crosstab.

Chart

Inserts a chart into a report. For details, see Inserting a chart.

Tabular

Inserts a tabular into a report. For details, see Inserting a tabular in a report.

Banded Object

Inserts a banded object into a report. For details, see Inserting a banded object.

Subreport

Inserts a subreport into a report. For details, see Subreports.

Map

- **Shape Map**
  Inserts a shape map into a report. For details, see Inserting a shape map.
- **Geographic Map**
  Inserts a geographic map into a report. For details, see Inserting a geographic map.

UDO

Inserts a user defined object into a report. For details, see Inserting a UDO into a report.

Special Fields

Inserts a special field which is not fetched from the database into a report.

Drawing Objects

Draws lines, boxes, arcs, ovals, and round boxes in a report.

Web Controls

Inserts web controls into a report. For details, see Web controls.

Multimedia Objects

Inserts multimedia objects into a report, including flash files, applets, Windows Media files, and RealMedia files. For details, see Multimedia objects.

Page Break

Inserts a page break before or after the current page in a report.

Page Panel

Inserts a page panel after the current page in a report.

Format menu

The Format menu contains the following items:

Conditional Formatting

Opens the Conditional Formatting dialog to add some conditional formats to the selected object.

Move

- **Move Backward**
  Places the selected object that overlaps one or more objects behind the current overlapping object.
- **Move To Back**
  Places the selected object that overlaps one or more objects behind all the overlapping objects.
- **Move Forward**
  Brings the selected object that overlaps one or more objects to the front of the current overlapping object.
- **Move To Front**
  Brings the selected object that overlaps one or more objects to the front of all the overlapping objects.

Align

- **Left**
  Shifts the selected object to the left.
- **Right**
  Shifts the selected object to the right.
- **Top**
  Shifts the selected object to the top.
- **Bottom**
  Shifts the selected object to the bottom.
- **Vertical Center**
  Aligns the selected object in the center vertically.
- **Horizontal Center**
  Aligns the selected object in the center horizontally.
- **Vertical Distribute Space**
  Spaces the selected object evenly in a vertical direction.
- **Horizontal Distribute Space**
  Spaces the selected object evenly in a horizontal direction.
- **Left**
  Aligns the content of the selected object to the left boundary of the object .
- **Center**
  Aligns the content of the selected object to the center of the object.
- **Right**
  Aligns the content of the selected object to the right boundary of the object.
- **Justify**
  Adjusts horizontal spacing so that the content is aligned evenly along both the left and right margins in the selected object.

Text Style

- **Bold**
  Specifies whether or not to bold the text in the selected object.
- **Italic**
  Specifies whether or not to italicize the text in the selected object.
- **Underline**
  Specifies whether or not to underline the text in the selected object.

Position

- **Static**
  Positions the selected object to the default location.
- **Relative**
  Positions the selected object to "flow" relating to its surrounding objects.
- **Absolute**
  Sets the selected object to the location you specify by dragging and dropping or setting the X and Y properties.

Report menu

The Report menu contains the following commands:

Dataset Management

Manages existing datasets. For details, see Managing datasets.

Edit Display Name

Specifies the field display names for the Sort, Filter, Search dialogs and the Drill menu items in Page Report Studio. For details, see Customizing the field display names.

Edit Sort and Filter

Predefines sort and filter conditions for banded objects/tables/charts/crosstabs. For details, see Predefining sort and filter conditions.

Bursting

Opens the Bursting dialog to enable the current report to become a bursting report.

Window menu

The Window menu contains the following commands:

Cascade

Cascades the design windows of the open reports.

Tile Horizontally

Tiles the design windows of the open reports horizontally.

Tile Vertically

Tiles the design windows of the open reports vertically.

Windows

Adjusts the open reports.

Note: Besides the above, you may see one or more items labeled file names, and the item representing the current report is marked with a check. You can click an item to activate the window of the report that it represents.

Help menu

The Help menu contains the following commands:

Tutorial

Displays the JReport Tutorial.

User's Guide

Displays the JReport Designer User's Guide.

Welcome Page

Shows the Welcome Page.

Jinfonet Software Home Page

Accesses Jinfonet Software website.

Online Support

Opens the Jinfonet online support page.

Knowledge Base

Opens the page which directs to the Jinfonet knowledge base.

About JReport

Brings out the About JReport Designer window. This window shows general product information about JReport Designer.

Toolbars

By default, JReport Designer provides the following toolbars:

- Standard toolbar
- Format toolbar
- Insert toolbar
- NLS Language toolbar
- View toolbar
- Alignment toolbar

Tips:

- You can choose which of the toolbars can be displayed in JReport Designer. To do this, right-click the space on any toolbar and check or uncheck the checkboxes ahead of the toolbar names, or click **View** > **Toolbar** and then check or uncheck the toolbars on the submenu.
- You can customize the toolbars according to your requirements. To do this, right-click the space on any toolbar and select **Customize** from the shortcut menu, then in the Customize Toolbars dialog, define the toolbars as required. You can also access this dialog by clicking View > Toolbar > Customize.
- If you want to reset the toolbars to the default, right-click the space on any toolbar and then click **Reset** on the shortcut menu.

Standard toolbar

This toolbar by default contains the following buttons:

New Page Report

Creates a new page report. For details, see Creating a page report.

Open

Brings up the Open Report dialog to open a report.

Save

Saves the current report.

Page Setup

Opens the Page Setup dialog to set page properties for the active report tab in the current page report, or for the current web report.

Print

Shows the Print dialog for you to print the active report tab in the current report. The dialog varies according to the printing method you choose.

Preview As

Previews the report result in the specified format. For details, see Previewing a page report.

Cut

Erases an object and places it on the clipboard.

Copy

Copies an object and places it on the clipboard.

Paste

Takes an object from the clipboard and places it into the report.

Delete

Deletes the selected object.

Undo

Reverses a previous action.

Redo

The counter-operation of Undo.

Find DBField

Finds specific DBFields in a report.

Freeze Field in Table

Stops the fields in a table from accepting any mouse or keyboard events, or releases the status of freezing.

Merge and Split

Merges cells in a table or tabular, or split a tabular cell, or unmerge a merged cell in a table.

Format toolbar

This toolbar contains the following buttons and drop-down list boxes by default:

Font Face

Specifies the font face of the text in the selected object.

Font Size

Specifies the font size of the text in the selected object.

Style

Specifies the style group applied to the selected object.

Bold

Specifies whether or not to bold the text in the selected object.

Italic

Specifies whether or not to italicize the text in the selected object.

Underline

Specifies whether or not to underline the text in the selected object.

Justify

Adjusts horizontal spacing so that the content is aligned evenly along both the left and right margins in the selected object.

Left

Aligns the content of the selected object to the left boundary of the object.

Center

Aligns the content of the selected object to the center of the object.

Right

Aligns the content of the selected object to the right boundary of the object.

Background Color

Specifies the background color of the selected object.

Foreground Color

Specifies the foreground color of the selected object.

Conditional Formatting

Opens the Conditional Formatting dialog to add some conditional formats to the selected object.

Insert toolbar

Below are the buttons contained on this toolbar by default:

Insert DBField

Inserts a DBField into a report. For details, see Inserting a DBField in a report.

Insert Formula

Inserts a formula field into a report. For details, see Inserting a formula in a report.

Insert Summary

Inserts a summary field into a report. For details, see Inserting a summary field in a report.

Insert Parameter

Inserts a parameter field into a report. For details, see Inserting a parameter field in a report.

Insert Group

Creates and edits the criteria by which the fields of certain components are grouped and sorted. For details, see Grouping the data.

Insert Label

Inserts a label into a report.

Insert Text Box

Creates and places a text box into a report.

Insert Image

Places a BMP, JPEG, or GIF image into a report.

Insert Barcode

Inserts a barcode into a report. For details, see Barcodes.

Insert Rank

Places a rank object into a report. For details, see Ranks.

Insert Table

Inserts a table into a report. For details, see Inserting a table.

Insert Crosstab

Inserts a crosstab into a report. For details, see Inserting a crosstab.

Insert Chart

Inserts a chart into a report. For details, see Inserting a chart.

Insert Tabular

Inserts a tabular in a report. For details, see Inserting a tabular in a report.

Insert Banded Object

Inserts a banded object into a report. For details, see Inserting a banded object.

Insert Subreport

Inserts a subreport into a report. For details, see Subreports.

Insert Shape Map

Inserts a shape map into a report. For details, see Inserting a shape map.

Insert Geographic Map

Inserts a geographic map into a report. For details, see Inserting a geographic map.

Insert UDO

Inserts a user defined object into a report. For details, see Inserting a UDO into a report.

Insert Special Field

Inserts a special field into a report. You can click the downward triangle to show all the available special fields.

Insert Drawing Object

Draws a drawing object in a report.

Insert Web Control

Inserts a web control. You can click the downward triangle to show all supported web control types. For details, see Web controls.

Insert Multimedia Object

Inserts a multimedia object into a report. For details, see Multimedia objects.

NLS Language toolbar

This toolbar contains the following items by default:

View Mode Language

Specifies the language in which to view the report result. For details, see Previewing a NLS report.

NLS Editor

Shows the NLS Editor for you to edit NLS file. For details, see NLS Editor.

View toolbar

The following items are contained on this toolbar by default:

Toolbox

Shows or hides the Toolbox panel.

Catalog Browser

Shows or hides the Catalog Browser.

Inspector

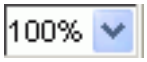
Shows or hides the Report Inspector.

Page Layout

Specifies whether or not to show the report in page layout style, which displays the page margins and enables multiple pages to be continuous.

Show Hidden Object

Specifies whether or not to show hidden objects.



View Percent Box

Lists percentages with which you can zoom in or out of a report.

Alignment toolbar

The Alignment toolbar provides the following buttons by default:

Align Left

Shifts the selected object to the left.

Align Right

Shifts the selected object to the right.

Align Top

Shifts the selected object to the top.

Align Bottom

Shifts the selected object to the bottom.

Align Vertical Center

Aligns the selected object to the center vertically.

Align Horizontal Center

Aligns the selected object to the center horizontally.

Vertical Distribute Space

Spaces the selected object evenly in a vertical direction.

Horizontal Distribute Space

Spaces the selected object evenly in a horizontal direction.

Move To Back

Places the selected object that overlaps one or more objects behind all the overlapping objects.

Move To Front

Brings the selected object that overlaps one or more objects to the front of all the overlapping objects.

Move Backward

Places the selected object that overlaps one or more objects behind the current overlapping object.

Move Forward

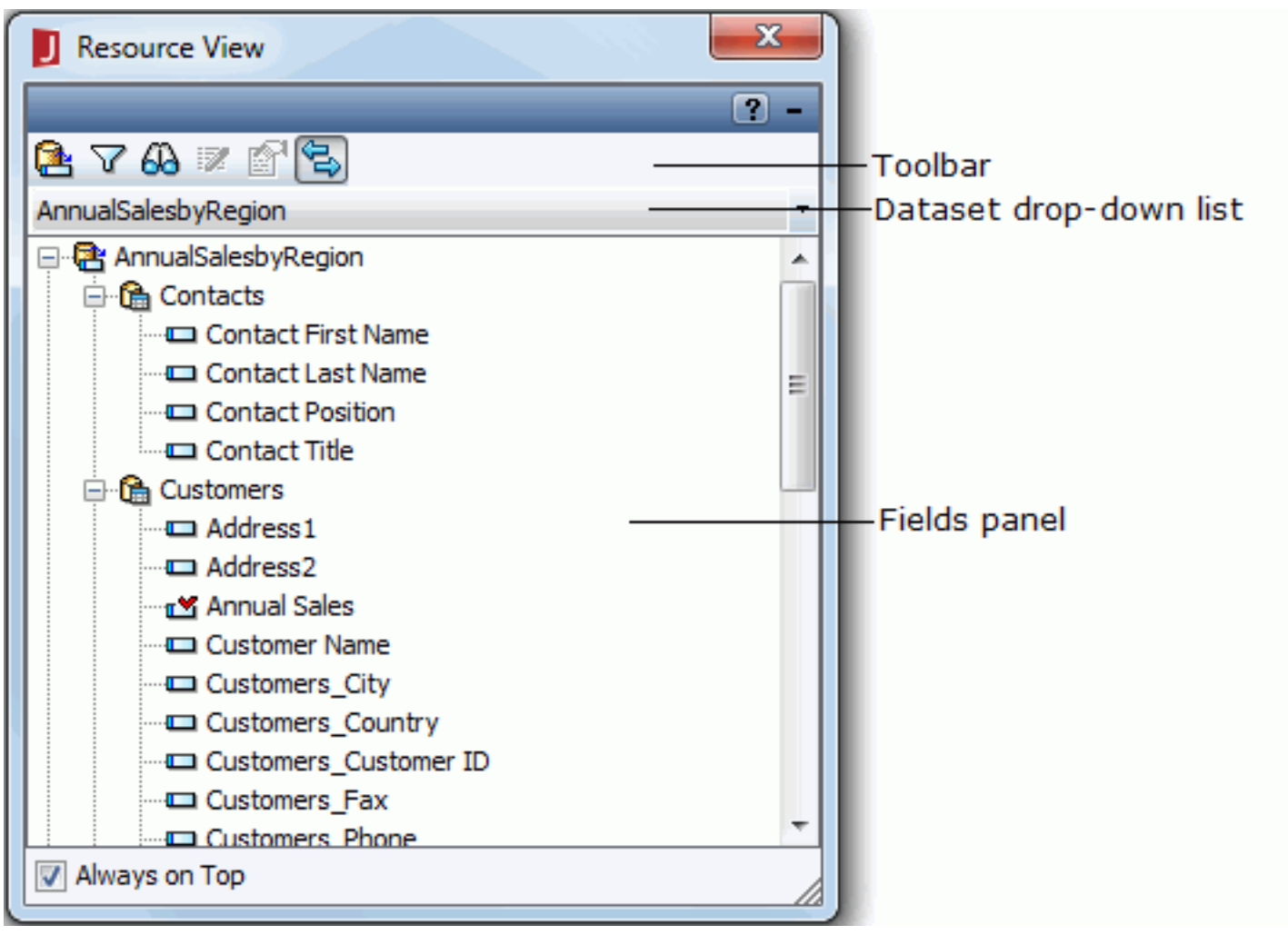
Brings the selected object that overlaps one or more objects to the front of the current overlapping object.

Change Position

Click the downward triangle to change the position of an object.

Resource View panel

The Resource View panel, which can be displayed by clicking View > Resource View on the menu bar, is an integrated interface for managing the resources that are used in the current report. It consists of a toolbar, a dataset drop-down list, and a field panel as shown below:



Toolbar

The toolbar includes the following buttons:

- **Views**
  This button varies with the view you select. It provides a shortcut menu for you to switch among the view modes (Query View, Dataset View, and Dataset List View). The Query View mode lists all available resources for the current report. The Dataset View mode also lists all these resources, however with a checkbox left to every resource for you to select (for a dataset coming from a hierarchical data source, there are no such checkboxes). The Dataset List View mode lists the resources you have selected in the dataset view mode.

- **Dataset Filter**
  Opens the Dataset Filter dialog to filter out some unnecessary data.

- **Search Resource**
  Searches the required resources in the Resource View panel. See the topic below for details.

- **Edit**
  This button is for editing the selected object if it is a query, formula, summary or parameter. According to what you have selected, clicking this button will bring out different dialogs.

- **Properties**
  Opens the Properties dialog of the selected object. You can edit the properties as desired.

- **Link with Editor**
  If this check button is selected, the Dataset drop-down list will show the dataset bound to the currently selected object; or else the drop-down list will always show the dataset you have selected from the drop-down list, until you select another one.

Dataset drop-down list

This drop-down list is available for page report only. It lists all the datasets used in the current report. The first item <Choose Data from...> provides access to creating a dataset.

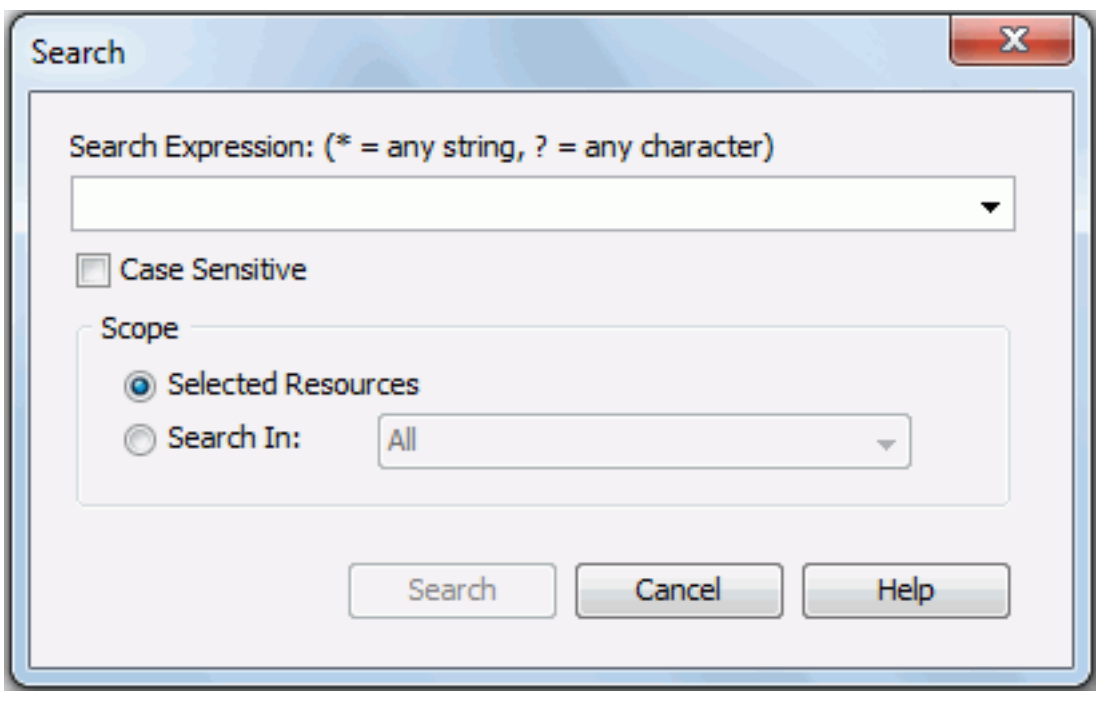
Fields panel

This panel lists all available resources for the current report in a tree structure. The DBFields, formulas, summaries, and parameters in this panel respond to right-clicking events, and thus provide you with shortcuts for viewing and editing their properties, editing formulas/summaries/parameters, creating new formulas/summaries/parameters, and operating queries.

Searching for resources in the Resource View panel

You can search the resources you want in the Resource View panel as shown below:

1. Click on the Resource View toolbar to display the Search dialog.



2. Provide a search condition in the Search Expression text box. Use "*" to stand for any string, and use "?" to stand for any character.

3. Select the **Case Sensitive** checkbox if you want JReport to distinguish between uppercase and lowercase characters when searching.

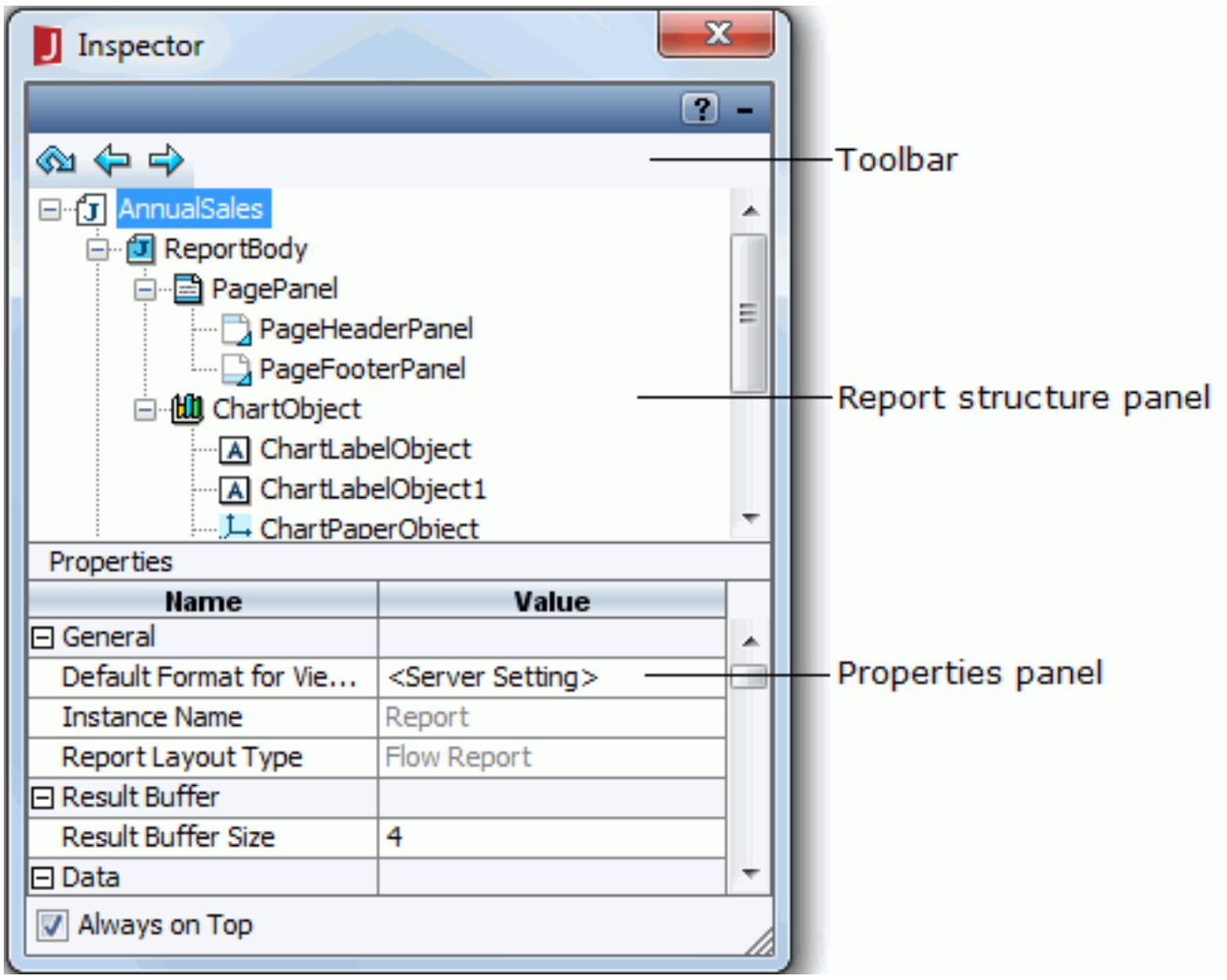
4. Specify the search scope by selecting the radio button Selected Resources or Search In.
   - **Selected Resources**: If you have selected a resource in the resource tree, when you bring out the Search in Resource Viewer dialog, this radio button will be selected and JReport will search for the resources from the selected resource.
   - **Search In**: If it is selected, JReport will search for the resource from the Resource View panel.

5. Click the **Search** button to search for the resource you want.

   The result will be listed in the Search Result panel which has three columns: Resource Name, Resource Type and Resource Path. You can sort the search result alphabetically by clicking the column headers. When you select a search result from the Search Result panel, the resource node that contains it will be expanded with that resource highlighted in the Resource View panel.

Inspector panel

The Inspector panel can be displayed by clicking View > Report Inspector on the menu bar, or the Report Inspector button on the View toolbar.



Toolbar

A page report may contain one or more report tabs, and each report tab contains a report body. You can use the toolbar to focus on one of the three levels, namely the report, report tab, and report body.

- **Home**
  Only shows the current report tab and its contents (including TOC of the report tab and datasets used in the report tab) in the Report structure panel.

- **Down**
  Goes down a level, that is, clicking this button will switch from the report level to the report tab level, or from the report tab to the report body.

- **Up**
  Goes up a level, that is, clicking this button will switch from the report body level to the report tab level, or from the report tab to the report.

Report structure panel

This panel shows all the objects in the current report tab in a tree structure.

The display name of each object in this tree can be edited. To do this, select the object in the panel, after 2 or 3 seconds, click it again and the name will become editable, then give the object a new name and press **Enter** on the keyboard to confirm.

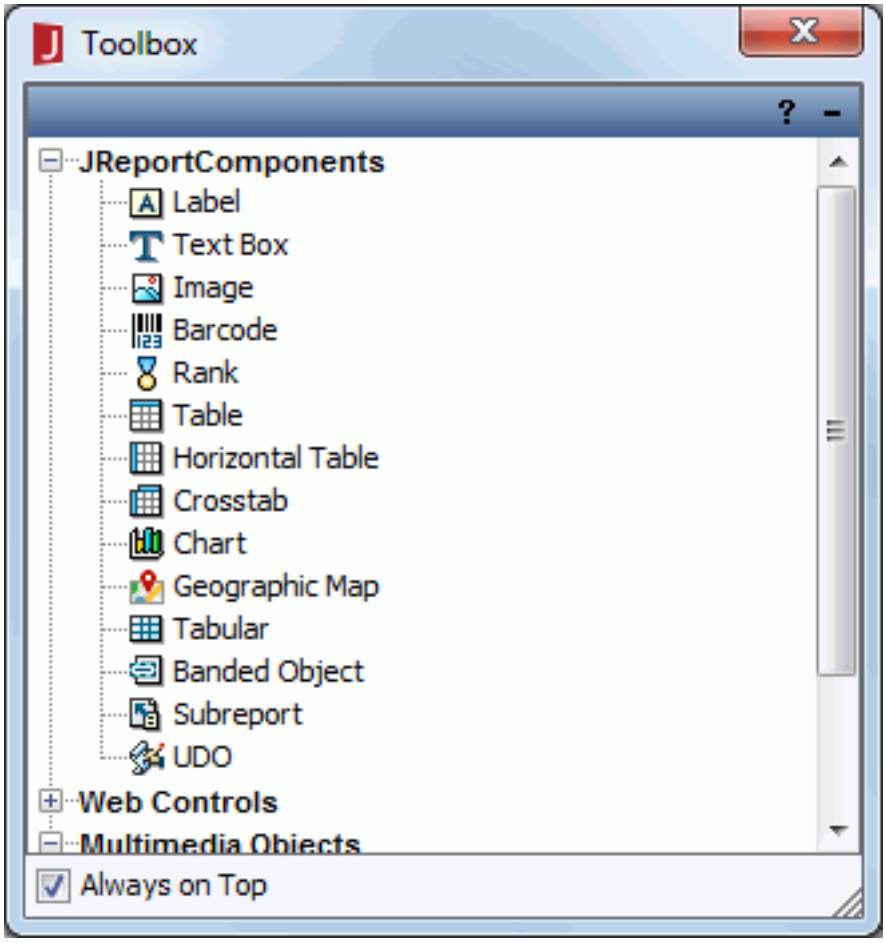
Properties panel

This panel lists all the properties of the object selected in the report structure panel. Each type of object has its own set of properties. Depending on the object type, the properties that a certain object holds may greatly differ from those of another. The Properties sheet has two columns: Name and Value. You can change how an object appears and behaves by changing its value in the Value column.

In addition, you can also click the button on the title bar to get detailed explanation about each property. For details about properties of each object, see Properties in the Report Inspector.

Toolbox panel

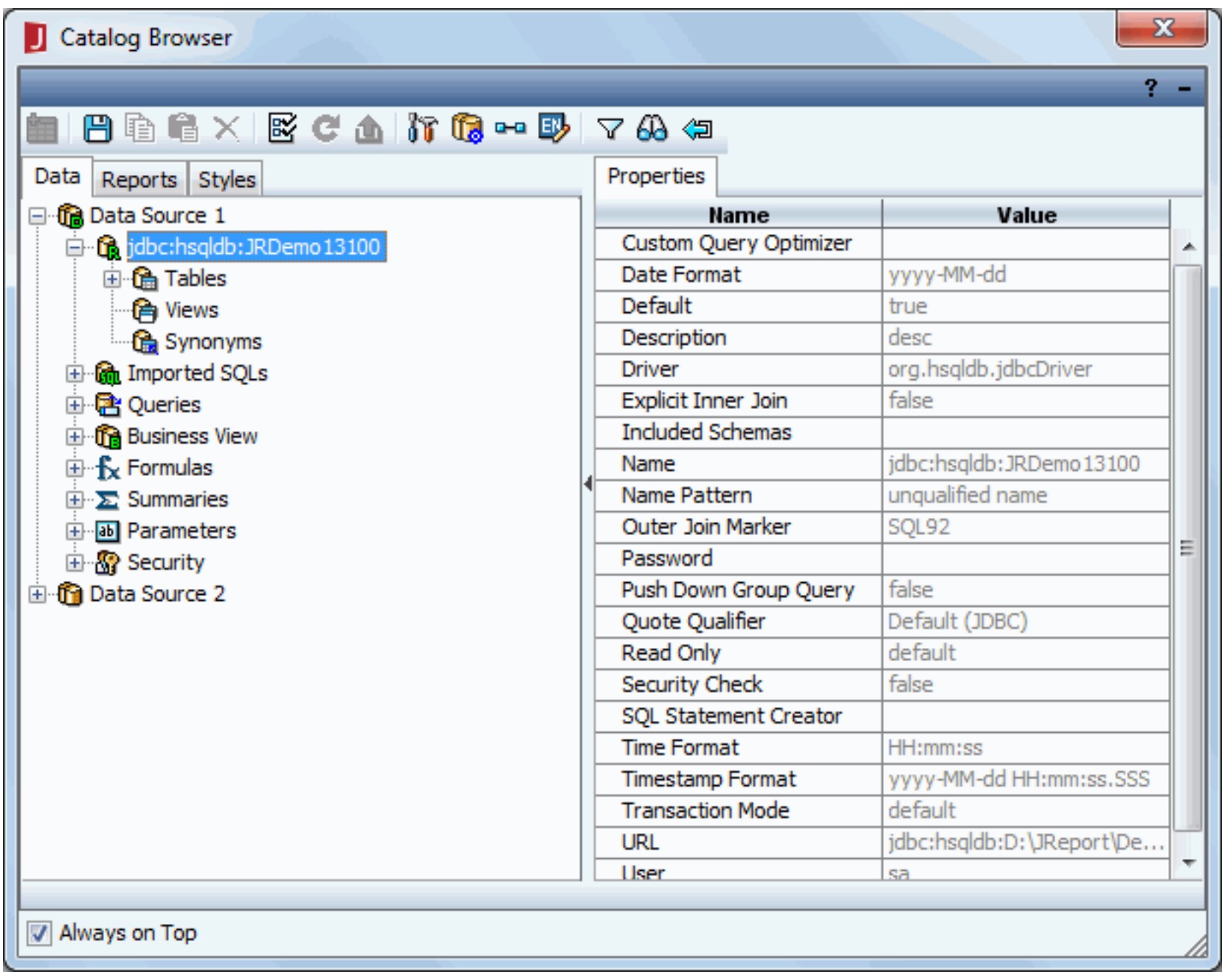
You can click View > Toolbox on the menu bar, or the Toolbox button on the View toolbar to show the Toolbox panel.



This panel lists all the objects that can be inserted into a report, which are classified as JReport Components, Web Controls, and Multimedia Objects. In fact, it is a shortcut to some items on the Insert menu. The method to create an object using the Toolbox panel is to drag a button representing that object from the Toolbox panel to the desired location in the report design window.

Catalog Browser panel

Clicking View > Catalog Browser on the menu bar, or the Catalog Browser button on the View toolbar will show the Catalog Browser panel.



This panel lists all data sources, reports, and styles belonging to the current catalog file (.cat) in three tabs: Data, Reports, and Styles. This is a useful interface which simplifies the work of report design and the management of the actual resources you use.

Data tab

In this tab, you can manage data sources and their contents used by the catalog.

Reports tab

Manages your page reports in folder view. You can divide them up into different types. In this way, you can manage them just like processing files in a file system. You can add, open, delete, move, rename and search for items. For details, see Page Report Manager.

Styles tab

From the Styles tab, you can create and edit styles. For details, see XSD styles.

Properties sheet

Lists the properties of the selected object. This sheet is hidden by default. To display it, click the **Expand** button on the Catalog Browser toolbar, or right-click any object in the left tabs and select **Properties** from the shortcut menu. For details about the properties, see Properties in the Catalog Browser.

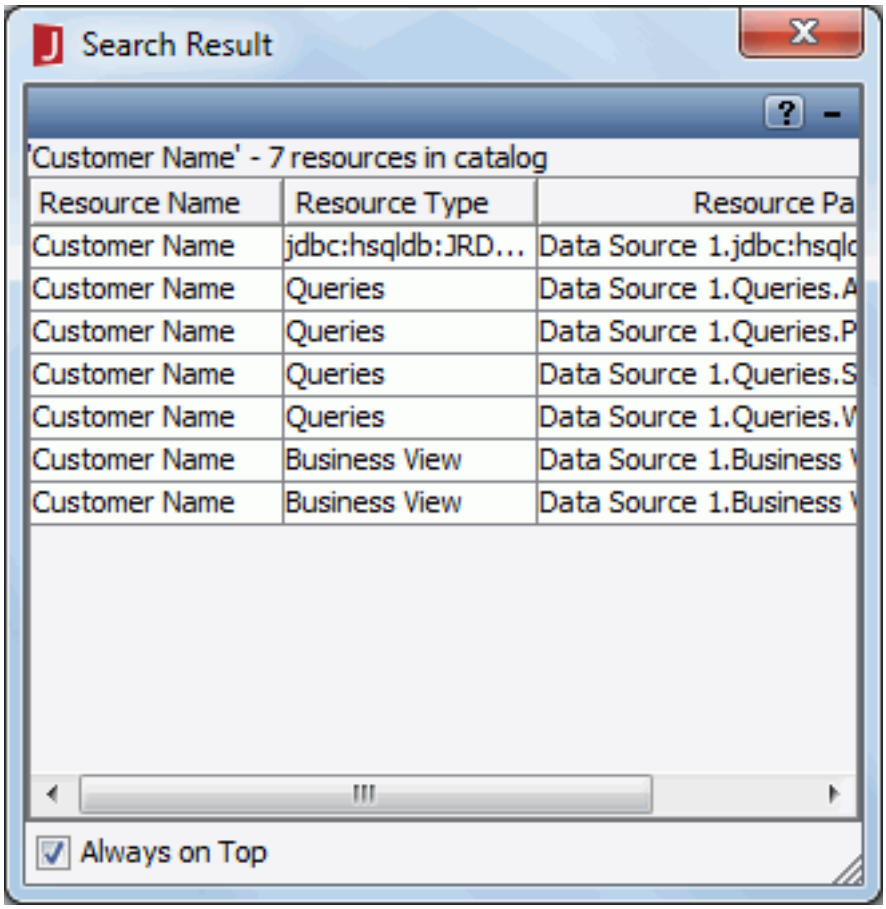
Toolbar

The following are buttons on the Catalog Browser toolbar:

- **New**
  The first button on the toolbar, which may change according to the currently selected object. Usually, this button contains a plus sign. Click this button to create a new object of the selected type. For example, if a query is currently selected, you can click this button to create a new query.
- Adjusts the order of parameters displayed in the parameter value specifying page. This button is available when the Parameters node or a parameter is selected. For details, see Editing the display sequence of parameters.
- **Save Catalog**
  Saves the current catalog.
- **Copy**
  Copies the selected entity to the clipboard.
- **Paste**
  Fetches the entity from the clipboard and pastes it in a suitable node.
- **Delete**
  Deletes the selected entity.
- **Configure Reference**
  Allows you to select data sources and resource types that you want JReport Designer to monitor. For details, see Using a reference table to clarify resource relationships.
- **Refresh Reference**
  Refreshes the reference relationship information in order to make it consistent with the catalog resources.
- **Export Reference**
  Exports the reference relationship information to a text file.
- **Options**
  Opens the Options dialog to set options for editing a catalog or reports.
- **Data Source Driver Manager**
  Manages cached query results. For details, see Managing cached query results.
- **Pre-join**
  Defines the pre-join paths formed by the relationships among the tables in the data source you specified. For details, see Defining the join relationships between tables.
- **Data Mapping Editor**
  Maps the data values of resources in a catalog to a certain language. For details, see Data Mapping Editor.
- **Filter View**
  Excludes some elements from the view in the Catalog Browser. For details, see Filtering resources.
- **Search**
  Searches the required resources in the current catalog. For details, see Searching for resources.
- **Expand** / **Fold**
  Expands or folds the Properties sheet of the Catalog Browser.

Search Result panel

This panel lists the result of searching for a particular resource.



Resource Name

Lists the resource names.

Resource Type

Shows the resource types.

Resource Path

Displays the resource paths.

Design/View area

The Design/View area, or report-designing window, comprises of the following parts:

Title bar

The title bar shows the report file name, and three buttons for you to minimize, maximize and close the active report.

Design tab

The Design tab is a WYSIWYG area for laying out your report. You can make use of rulers, grids and guidelines to position objects.

View tab

The View tab is for previewing the report layout and result. In this tab there is a toolbar which consists of the following buttons:

- **First Page**
  Displays the first page of the report.
- **Previous Page**
  Displays the previous page of the report.
- **Next Page**
  Displays the next page of the report.
- **Last Page**
  Displays the last page of the report.
- **Toc**
  Shows the table of contents.
- **Refresh Data**
  Updates any changes you created in your data.
- **Refetch Data**
  Re-fetches data to re-run the report.
- **Stop**
  Click this button if you want JReport to stop generating the report.
- **Back**
  Navigates to the last view.
- **Forward**
  Navigates forward.
- **Drag Tool**
  Drags the page up and down.

  Click this button, and place your mouse pointer on any blank area of the report. When the mouse pointer becomes a small hand, you can drag the page.
- **Zoom In Tool**
  Enlarges the page.
- **Zoom Out Tool**
  Click to view the page in a smaller size.

Report tab bar

The report tab bar is for switching among report tabs in a page report. It also provides a shortcut menu that allows you to rename, duplicate, or remove a report tab. For details, see Managing report tabs in a page report.

Catalogs

A catalog is a report repository. Every report must exist within the context of a catalog. A catalog contains two types of data: a physical file that records data sources and object definitions used by the reports in the catalog, and the report files themselves. These two types of data, the report files and the catalog file, are stored in the same directory.

JReport Designer supports two formats of catalog files. One is the .cat file, which is in binary, and the other is the .cat.xml file, which is in XML format. The .cat format catalog can be manipulated (opened, saved, and edited) only by JReport Designer since it is a proprietary binary file. The XML format catalog has many advantages. It has more readability and controllability. You can modify the catalog freely according to your requirements without using any other special editor. Howerver, the operation of .cat.xml format catalog in JReport Designer is the same as that of .cat format catalog. The difference between these two is the underlying structure in which they are organized. By default, the XML format catalog feature is not enabled. Contact Jinfonet Support (support@jinfonet.com) to upgrade your license key if desired.

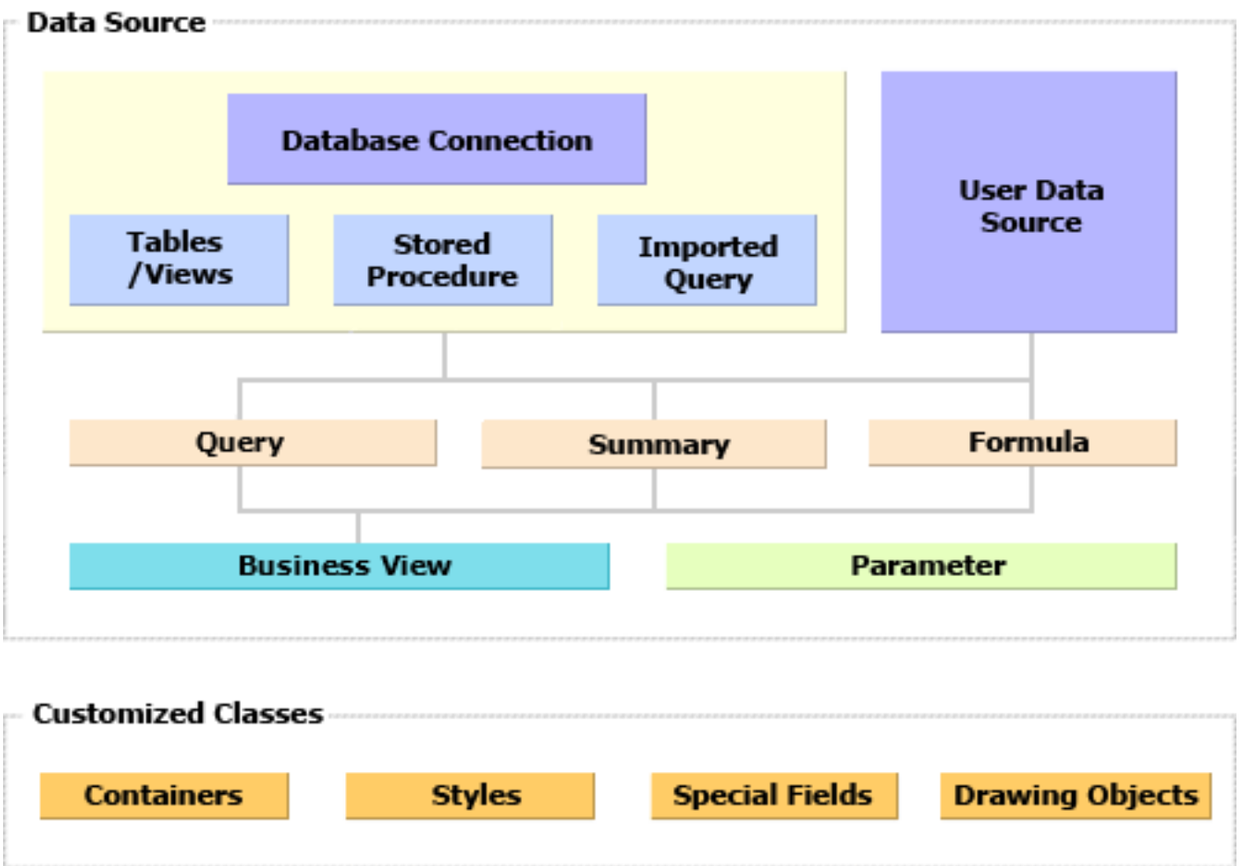
JReport Designer comes with four catalogs: JinfonetGourmetJava, SampleComponents, SamepleReports and TutorialReports. The JinfonetGourmetJava catalog is used for the JReport Tutorial manual, and the rest are samples for you to learn JReport Designer.

This chapter describes the following topics and tasks related to catalogs:

- What is in a catalog file?
- Advantages of catalogs
- Creating a catalog
- Saving a catalog
- Opening a catalog
- Exporting catalog security into a security file
- Merging catalogs
- Managing resources in a catalog
- Maintaining catalogs

What is in a catalog file?

A catalog file stores all of the object definitions that you have created while developing the reports that are also stored in the catalog. This includes data source definitions, component customizations, style definitions, and more. The following illustration summarizes the content of a catalog file:



A database connection defines the database information from which the catalog retrieves its raw data. For example, it specifies the driver such as JDBC driver (for JDBC), or the data source name (for JDBC-ODBC bridge), and the user ID and the password for connecting to the raw data. In this way, a connection is the gateway to the raw database.

Tables/Views, stored procedures, imported queries (SQL files) and queries are all based on the JDBC connection. Queries are built using the interactive query designer and are based on the tables/views from the database. Only after you have added the tables/views to the catalog, can you build queries.

Stored procedures are defined in the database. They function like queries, and can be used as data sources for reports.

Imported queries, that are SQL statements in an external file, can be added to the catalog and also function as queries.

Functionally, imported queries (SQL files), stored procedures and queries are all the same. However, queries depend on the tables/views added, while stored procedures and imported queries (SQL files) only need to connect with your database (you do not necessarily have to add the database tables).

User data sources (UDS) and hierarchical data sources (HDS) are also like queries, but users write the code for fetching the records. They are independent from the JDBC connection.

Business views are a semantic layer - a view of the data source that you create for a business analyst to use to create analytic reports.

Parameters are used to control the report content at runtime. They are most often used for entering data selection criteria.

Formulas and summaries are objects that are computed at runtime.

Containers, styles, special fields, and drawing objects can each be customized by setting a wide range of object properties.

Advantages of catalogs

The main advantage to catalogs is that they provide a way to organize related reports and make it easy for these reports to share objects, have a consistent look and feel, and can be moved to different systems. Catalogs also provide these additional benefits:

- **Visual database - working off-line**
  From a system that is offline from the database, you can still continue to develop the reports that will access it. This is because the architecture of your database (tables/views, queries, and parameters) is stored in the catalog.
- **Sharing defined data fields and formulas among reports**
  Multiple reports can use data fields and formulas that have already been defined and saved in the catalog.
- **Reference table**
  JReport provides a reference table that can keep track of all the objects used in all the reports in a catalog. In addition to helping you build well-designed reports, this feature makes object names changes easier to manage.
- **Default values for objects**
  You can set the default values for certain catalog objects. Once set, their values do not need to be individually set in the report after they have been inserted.

Creating a catalog

A report can only exist within a catalog. Therefore you must create and open a catalog before you create or edit a report.

A catalog is represented at the operating system level as a directory that contains the catalog file (.cat) as well as the other report objects.

To create a catalog:

1. In JReport Designer, click **File** > **Catalog Management** > **New Catalog** on the menu bar.

   If necessary, JReport Designer will prompt you to save changes to the current open catalog. Only one catalog can be open at a time.

2. In the New Catalog dialog, enter the name for the catalog in the Name field.

3. In the Data Source Name field, input the name for the data source that will be created along with the catalog (when creating a catalog, by default, a data source will be created in the catalog at the same time). Make sure the name you give follows the Java class naming rule.

4. Specify the path where to save the catalog in the Directory field. You can also click the button ... to browse to and select the directory in which to create the catalog.

5. Click **OK** to create the catalog.

   The Catalog Browser displays with a data source created for the catalog. You can then set up the required connections to connect with the data source (for details about setting up connections, see Data Source Connections).

   A data source in a catalog can be connected with multiple connections, then when you create queries in such data source, you can mash-up multiple data resources in the queries, which come from all the connections the data source is connected with.

Note: The catalog name must include the extension (.cat or .xml) and the directory you choose must not already contain a catalog file.

JReport Designer supports the Multiple Data Sources feature, which means in one catalog, you can add as many data sources as you need.

To add another data source to a catalog, follow the steps below:

1. Do either of the following:
   - In the Catalog Browser, select the data source node, then click the **New Data Source** button on the Catalog Browser toolbar.
   - Click **File** > **New** > **Data Source** on the menu bar.

2. In the New Data Source dialog, specify a name for the data source in the Data Source Name field, select the connection type, then click **OK**.

3. Set up the first connection in the data source according to your requirements.

When multiple data sources are created in a catalog, you need to specify which data source will be the default one for the catalog (by default, the data source that is created along with the catalog is the default data source of the catalog). To do this, right-click the data source which you want to use as the default data source, then click **Set as Default** on the shortcut menu. A catalog must have one and only one default data source.

In addition, you can rename and delete the data sources in a catalog if required.

- To rename a data source, right-click the data source and select **Rename** from the shortcut menu. In the renaming box, enter a new name for the data source, then click outside the box to confirm the change.

- To delete a data source, right-click the data source, on the shortcut menu, click **Delete**. The default data source cannot be deleted.

Saving a catalog

To save a catalog, click **File** > **Catalog Management** on the menu bar, then select from the following commands:

- **Save Catalog**
  Saves the open catalog. The type, name and location were specified on the Input New Catalog Name dialog when you first created the catalog.
- **Save Catalog As**
  Saves the open catalog in the selected alternate format. Options are .cat or .cat.xml. The .cat.xml file can be in the same directory as the .cat file.

A directory can contain only one catalog file other than the .cat and .cat.xml of the same catalog. To merge multiple catalog files, see Merging catalogs.

Sharing catalog files among multiple report developers

By default, the report file and resources referenced by this report are created and saved in the current catalog. You can also save a report or other resource to a catalog other than the one in which it is created. All the catalog resources related to this report will then be merged into that catalog. This can allow a team of report developers to share resources. Each report developer can work on a local version of the catalog, and then use the Save To command to have his catalog "uploaded" to a universal catalog. The Save To command saves not only the report files, but also the resources (query, formulas, and parameters) that are referenced by this report.

For details about how to share catalog files, see Saving a page report to a different catalog.

Opening a catalog

You can open a catalog explicitly, by using the Open Catalog command, or you can open it implicitly, by opening one of the reports in the catalog.

To open a catalog explicitly:

1. Click **File** > **Catalog Management** > **Open Catalog** on the menu bar.

   If necessary, JReport Designer will prompt you to save changes to the current open catalog. Only one catalog can be open at a time.

2. In the Open Catalog File dialog, browse to and select the catalog you want to open and then click the **Open** button.

   The Catalog Browser displays, which lists the resources in the specified catalog.

Exporting catalog security into a security file

The security policies defined in catalogs can be exported to security files. The security files can then be used to create dynamic security for catalogs on JReport Server.

A catalog security file stores catalog internal security definitions, which include the following:

- Business view security
  - The Visible permission on business views and their elements
  - Member-level security
- Security entry
  - Record-level security and column-level security

To export the security of a catalog into a file:

1. Open the catalog.
2. Click **File** > **Catalog Management** > **Export Security** on the menu bar.
3. In the Save Catalog Security dialog, use the default name or provide a new one in the File Name text box and then click **Save**. The security file will be stored in the same location as the current catalog.

Merging catalogs

Two catalogs can be merged. The merge operation occurs when you save a report to another directory in which a catalog of the same name already exists.

Specifying the checking level

When merging catalogs, JReport Designer will check for differences between the two catalogs. You can specify the checking level to suit your requirements.

To specify the checking level:

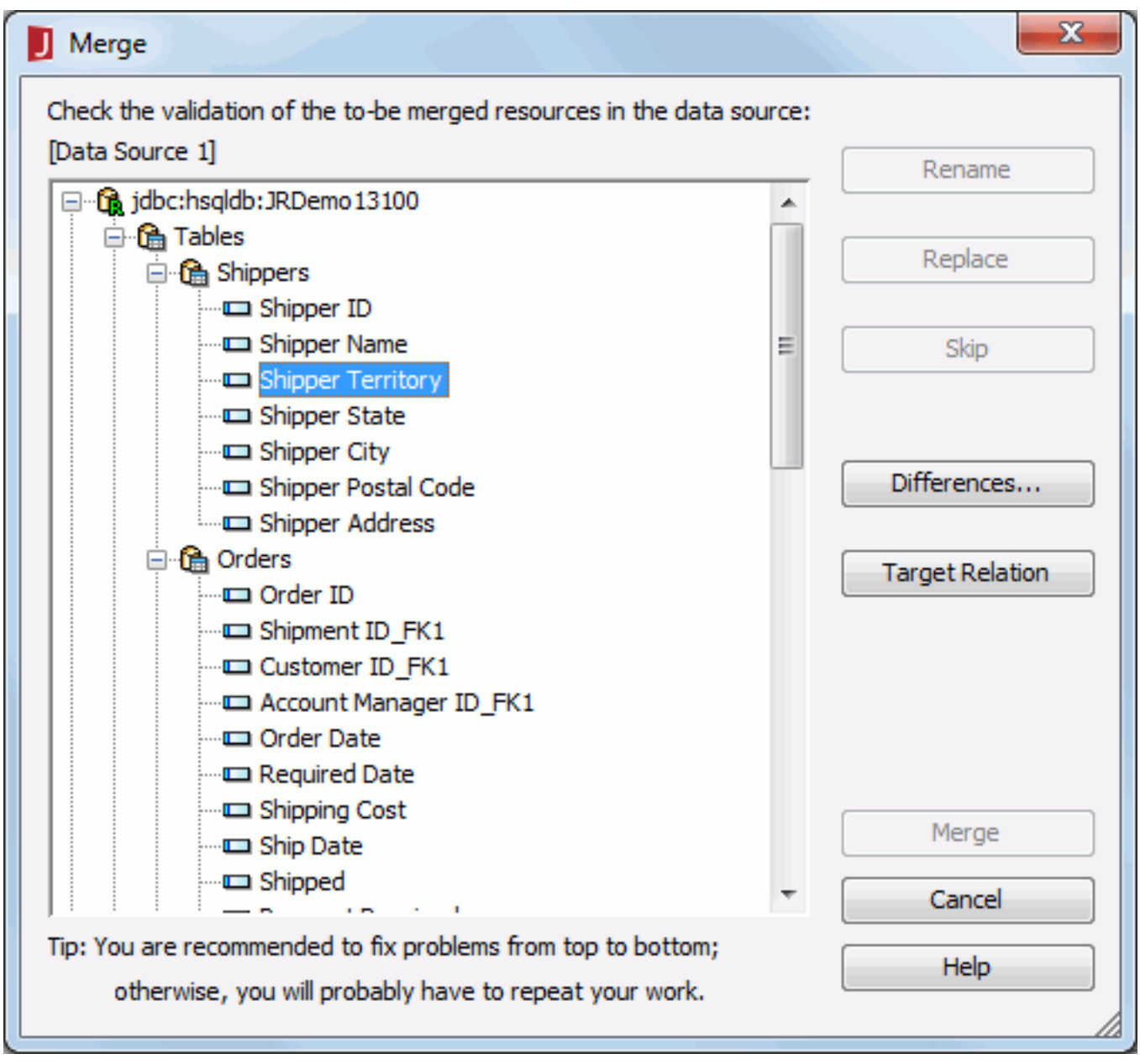
1. Click **File** > **Options**. The Options dialog displays.
2. In the Catalog category, select the required level in the Merge Catalog box. Options are:
   - **Identify All Differences**
     JReport Designer will check for all differences between the two catalogs. All the resources of the report will be displayed in the Merge dialog with all the conflicting resources marked.
   - **Identify Critical Differences**
     JReport Designer will check for differences that can cause the engine to fail when running the report. All the resources of the report will be displayed in the Merge dialog with any critical conflicting resources marked. As a result, when this option is selected, some different values may be returned when you run the saved report later.
   - **Ignore Differences**
     Does not show differences between the two catalogs. If there are conflicts, they will remain in the resources of the target catalog. As a result, when this option is selected, a saved report might not run later.
3. Click **OK** in the Options dialog to accept the settings.

When you merge the two catalogs, resources that have the same mapping names in the two catalogs may conflict with each other (have different property values). JReport Designer will check the differences for you according to the level specified.

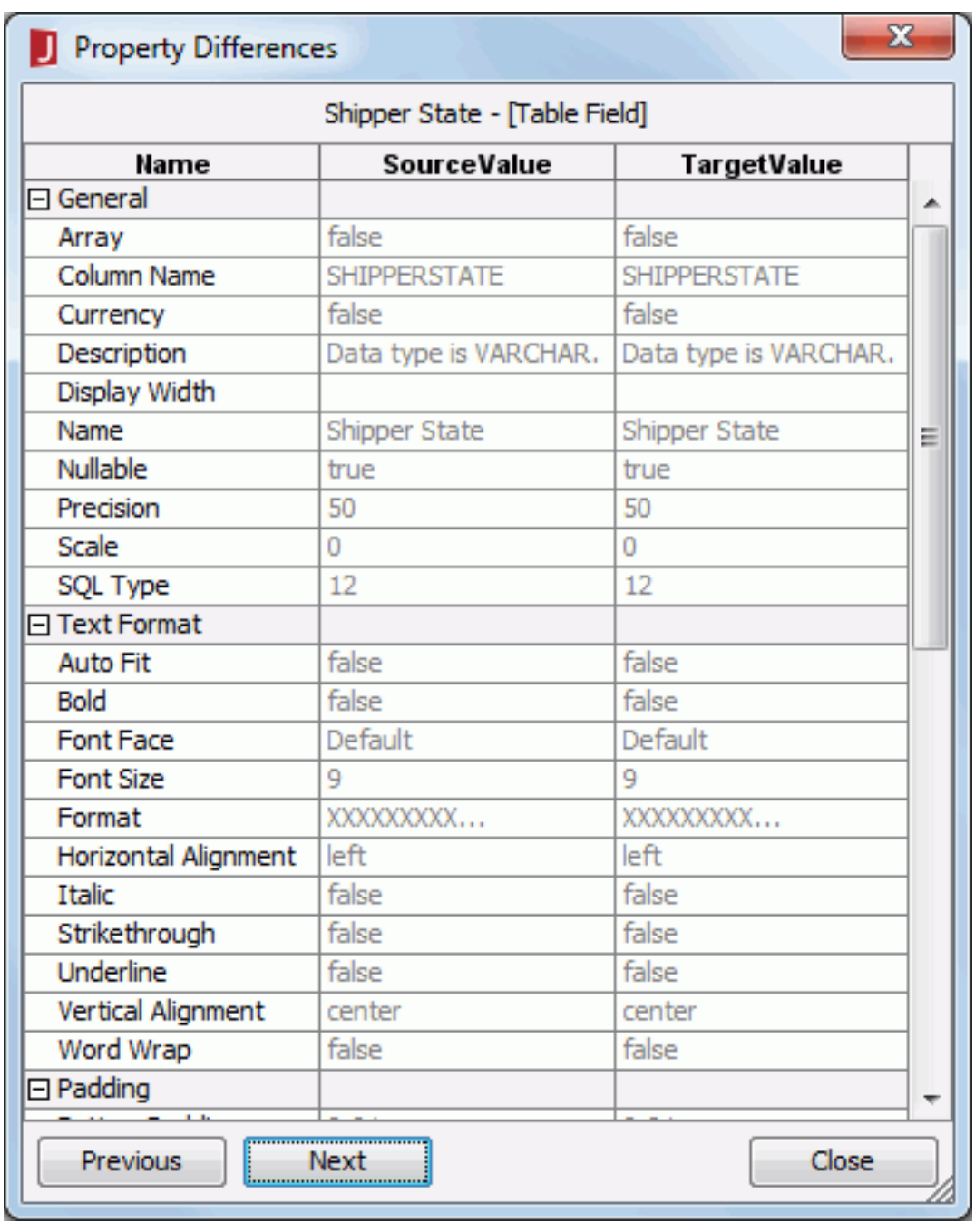
Merging catalogs

With the checking level specified, you can then start to merge catalogs.

1. Open a report in the catalog that you want to merge with another catalog.
2. Click **File** > **Save To** on the menu bar. The Save To dialog appears.
3. Choose a directory where a catalog of the same name exists, and then save the report. If you specified the checking level as Identify All Differences or Identify Critical Differences, the Merge dialog appears for dealing with conflicting resources.



4. Highlight one conflicting resource in the resources to be merged box, and then select the operations:
   - **Rename** - Renames the selected conflicting resource, and copies it to the target catalog with a new name.
   - **Replace** - Replaces the target catalog with the selected conflicting resource from the source catalog. In this case, the reports that use this object in the target catalog will be impacted.
   - **Skip** - Keeps the value in the target catalog for the selected conflicting resource. In this case, the report that uses this object in the source catalog will be impacted.
5. Click the **Differences** button and the Properties Differences dialog appears.



Name	SourceValue	TargetValue
General		
Array	false	false
Column Name	SHIPPERSTATE	SHIPPERSTATE
Currency	false	false
Description	Data type is VARCHAR.	Data type is VARCHAR.
Display Width		
Name	Shipper State	Shipper State
Nullable	true	true
Precision	50	50
Scale	0	0
SQL Type	12	12
Text Format		
Auto Fit	false	false
Bold	false	false
Font Face	Default	Default
Font Size	9	9
Format	XXXXXXXXX...	XXXXXXXXX...
Horizontal Alignment	left	left
Italic	false	false
Strikethrough	false	false
Underline	false	false
Vertical Alignment	center	center
Word Wrap	false	false
Padding		

   The properties values of the conflicting resource from the two catalogs will be displayed in this dialog with any differences highlighted. Click the **Previous** / **Next** button in the dialog, and it will go to the previous/next conflicting resource. Click **Close** to close the dialog.
6. Click the **Target Relation** button in the Merge dialog.

   The name of the resource and its parent node in the target catalog will then be displayed in the Target Relation dialog.

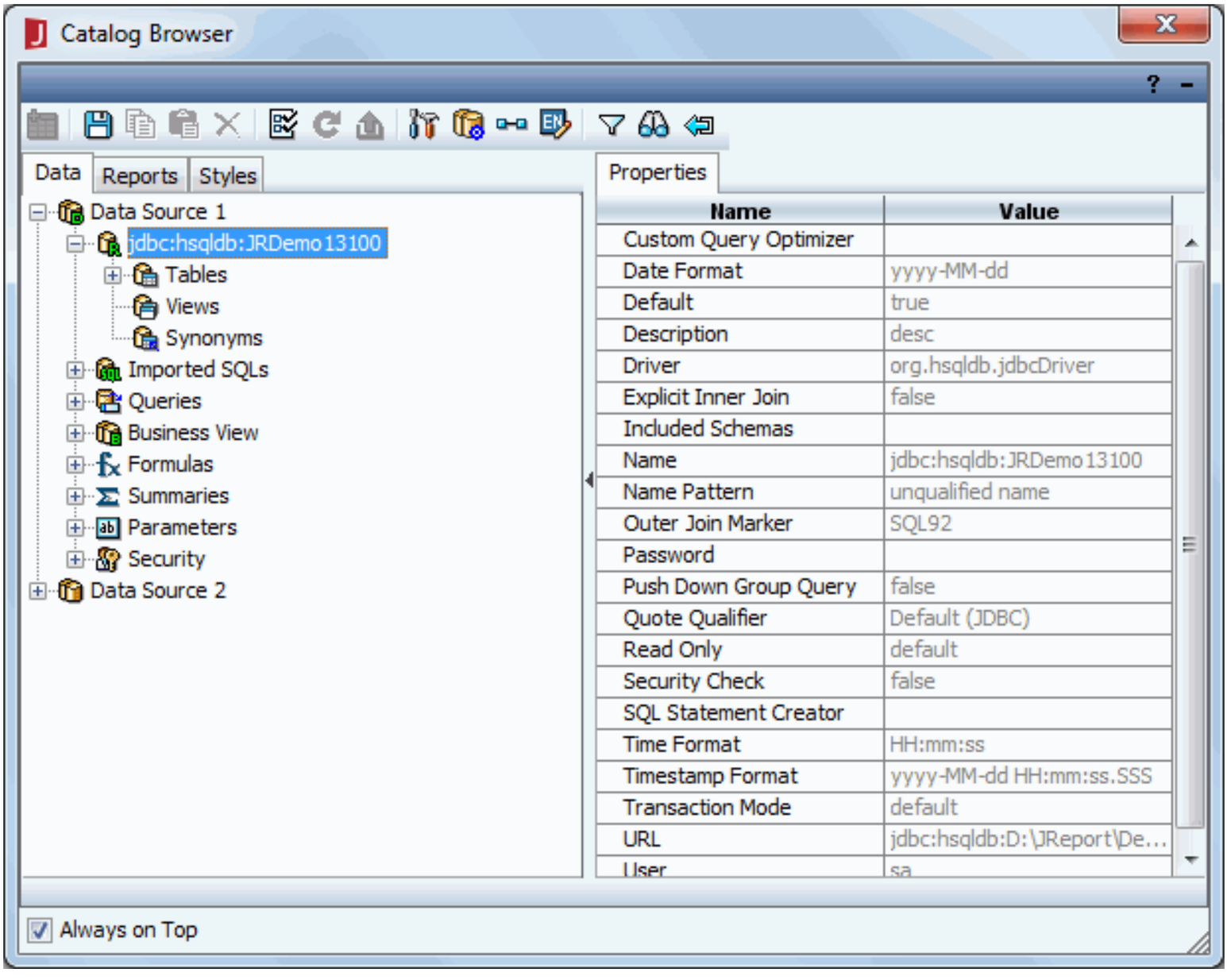
   For example, sometimes a resource is marked in the Merge dialog, but in the Properties Differences dialog, no difference is shown between the two resources. In this case you can use this button to check whether the parent nodes of the two resources are different.
7. Click **Close** to close the Target Relation dialog box and return to the Merge dialog.
8. Click the **Merge** button when you finish. JReport Designer will save the report according to the changes that you have made.
9. Publish the catalog and the reports and test to ensure all the reports are working properly.

Notes:

- Sometimes, though some objects are marked in the Merge dialog, in the Properties Differences dialog, you will not find these differences. The reason is that some children of the two objects are different.
- JReport Designer checks parameters based on the whole catalog. For example, assume that there is a parameter named Param1 in DataSource1 of the source catalog, and there is also a parameter named Param1 in DataSource2 in the target catalog. These two parameters in different connections will be checked as a conflicting resource. You must rename the parameter in order to get a correct result.

Managing resources in a catalog

You use the Catalog Browser to add, remove, or modify the resources in a catalog. The Catalog Browser provides a hierarchical view of the resources, reports, and styles in the current catalog:



To access the Catalog Browser, click **View** > **Catalog Browser** on the menu bar. See Catalog Browser panel for information about the panel.

This section describes the following tasks related to managing resources in a catalog:

- Adding and editing resources
- Copying, deleting and renaming resources
- Setting resource properties
- Searching for resources
- Filtering resources
- Using a reference table to clarify resource relationships
- Defining the join relationships between tables

Adding and editing resources

It is easy for you to create or add resources, and to edit the existing resources in a catalog using the Catalog Browser.

Adding a resource

Although there are a variety of resources in a catalog, the ways you create them are fairly similar - select a resource node of the type that you want to create, and then create them using either toolbar buttons or context menu.

To add a resource:

1. Open the catalog to which you want to add the resource.
2. In the Catalog Browser, expand the data source where the resource will be added, then right-click a resource node or object of the type that you want to create and select **Add XXX**. The appropriate object editor or dialog will then be displayed.

   For example, if you want to create a formula, right-click the **Formulas** node and select **Add Formula**. The Formula Editor window appears. Create the formula in the window.

Editing a resource

To edit any of the existing resource in a catalog, locate the resource in the Catalog Browser resource tree, right-click the resource and select **Edit XXX** from the shortcut menu. Then, the corresponding editor or dialog will appear for you to edit the resource.

Copying, deleting and renaming resources

You can copy, delete, and rename catalog resource objects in the Catalog Browser.

- To copy a resource object, select the resource object, on the toolbar of the Catalog Browser, click the **Copy** button . To paste it, click the **Paste** button .
- To delete a resource object, right-click the resource object, on the shortcut menu, click **Delete**.
- To rename a resource object, double-click the resource object, edit the resource name, and then click outside the renaming box to confirm the change.

  You can also rename objects by using the Properties sheet:

  1. Select the resource object, and click the **Expand** button on the Catalog Browser toolbar to show the Properties sheet.
  2. Select the property **Name**, type a new name in the Value field. You need to enable the editable mode for the Properties sheet before you can do this, that is, you need to uncheck the option **Forbid editing data object properties** in the Options dialog.

     Note: Renaming an object may impact the reports that use it. You must be aware of the reports currently using this object so that you can update those reports.

Setting resource properties

When you click the Expand button on the Catalog Browser toolbar, the property list of the selected resource will be displayed. See Properties in the Catalog Browser for detailed explanation about properties of each resource.

By default, the properties in the Catalog Browser cannot be edited. To make the properties editable:

1. On the Catalog Browser toolbar, click the **Options** button (or click **File** > **Options**).
2. In the Options dialog, select **Catalog** in the Category box.
3. Uncheck the option **Forbid editing data object properties**, then click **OK** in the dialog to confirm the change.

Example: Specifying the display width of a field

You can define the default width of a field (DBField, formula, summary or parameter) by setting the Display Width property in the Catalog Browser. To do this:

1. Make sure that the property values are editable in the Catalog Browser.
2. Highlight the field in the Catalog Browser, right-click it and then select **Properties** from the shortcut menu.
3. In the Properties sheet, set the value for Display Width as required.

Notes:

- If the Display Width value is not set in the Catalog Browser, it will be inserted with the width defined in the template. This is the width you selected when creating the report.
- Assume that you have set the Display Width of a field, and you insert the field using the wizard or by dragging and dropping, its width may be adjusted according to the paper size. However, if you insert the field in design mode, the defined width will not be changed.

Setting default values for resources

You can set the default values for the properties of resources in the Catalog Browser, so that you do not need to set them each time you insert them into your report. To do this:

1. Select the resource with the properties you want to set.
2. Click on the Catalog Browser toolbar to show its properties.
3. In the Properties sheet, change the values of the properties to suit you requirements.

The properties listed in the Properties sheet are the default values for the selected resource. You can change the property values here, and the next time you insert the resource, the value you specify here will be used as a default value.

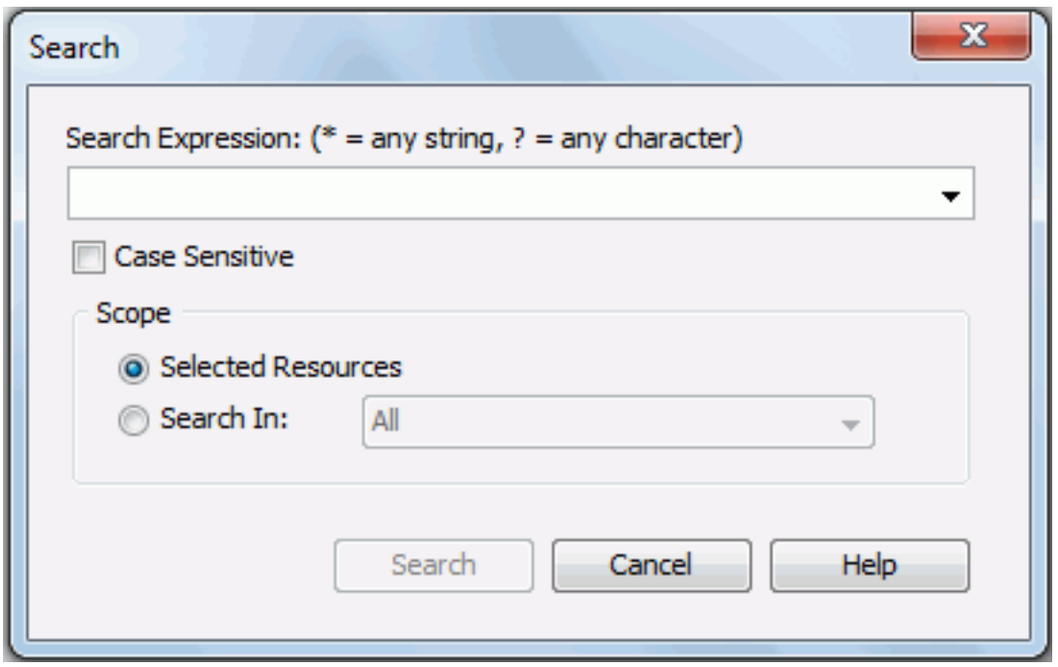
The resources that can have default values are table/view columns, formulas, parameters, summaries, SQL columns, procedure columns and user data source columns. The query column does not support this feature.

Searching for resources

When there are too many resources in a catalog, sometimes you may find that it is difficult to locate where a resource is from the Catalog Browser. JReport provides you with a searching function which enables you to easily find the required resources.

To search resources in a catalog:

1. Open the catalog where you want to search resources.
2. In the Catalog Browser, click the **Search** button on the toolbar to display the Search dialog.



3. Provide a search condition in the Search Expression text field. Use "*" to stand for any string, and use "?" to stand for any character.

   Note: JReport can only search resources in the leaf node of the catalog resource tree.

4. Select the **Case Sensitive** checkbox if you want JReport to distinguish between uppercase and lowercase characters when searching.
5. Specify the search scope by selecting the radio button **Selected Resources** or **Search In**.
   - **Selected Resources**: JReport will search from the resource you have selected in the Catalog Browser before you bring out the Search in Catalog dialog.
   - **Search In**: Search for the resources from the specified data source. If there are multiple connections in this catalog, select the data source you want from the drop-down list.
6. Click **Search** to search for the resource you want.

   All the matched results will be listed in the Search Result panel which has three columns: Resource Name, Resource Type and Resource Path. You can sort the search result alphabetically by clicking the column headers. Select a search result from the Search Result panel, the resource node that contains it will be expanded and with that resource highlighted in the Catalog Browser.

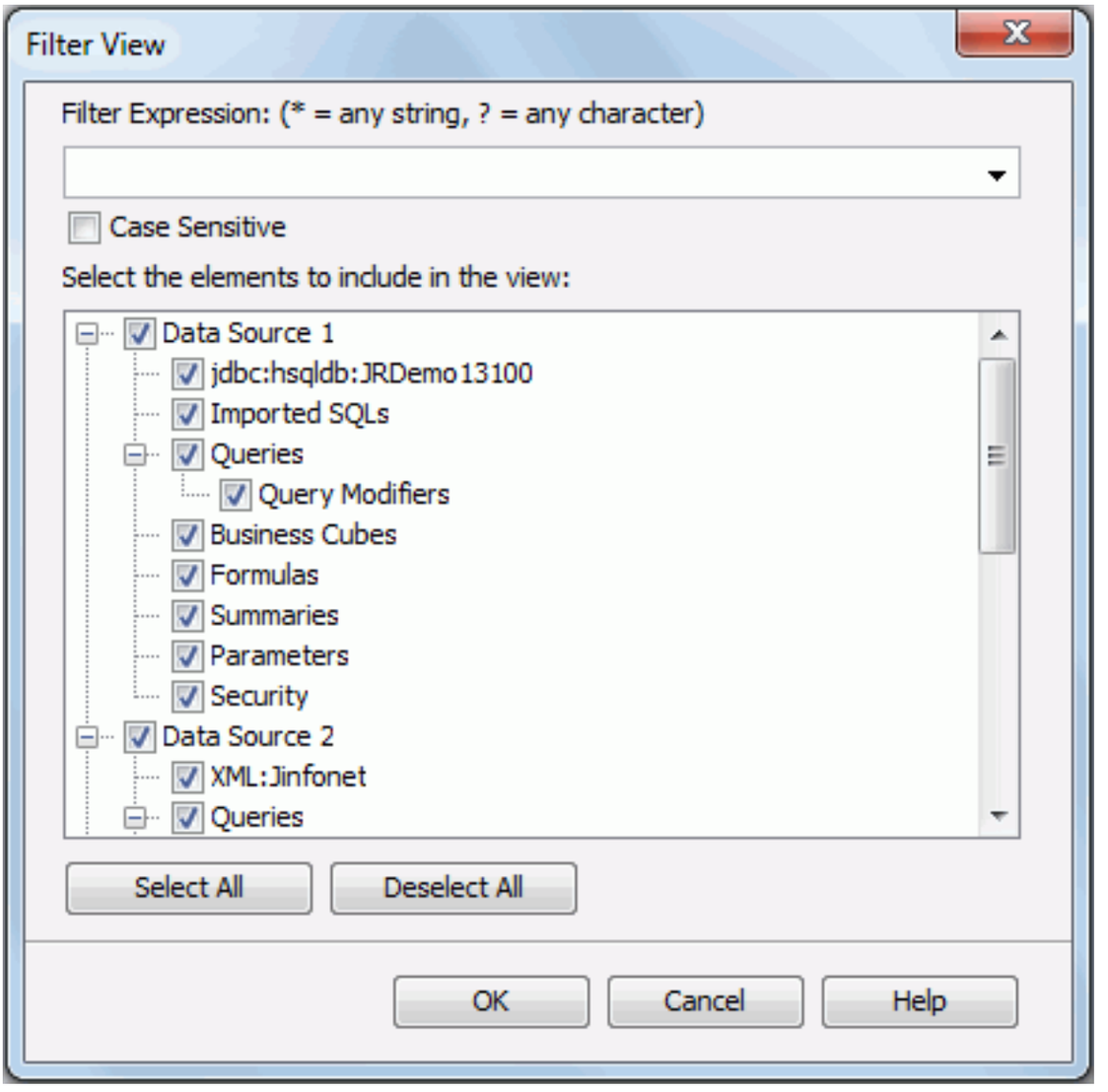
For example, if you want to find the DBField Customer Name in the Queries node, first select the node in the Catalog Browser, click , then in the Search dialog, input **Customer Name** in the Search Expression field, check the scope as **Selected Resource** and click the **Search** button. The results will then be displayed in the Search Result panel. You can click any of the results to highlight it in the catalog resource tree.

Filtering resources

JReport allows you to filter the data resources in a catalog and remain only the useful resources for you.

To filter resources in a catalog:

1. In the Catalog Browser, click the **Filter View** button on the toolbar and the Filter View dialog appears.



2. Provide a filter condition in the Filter Expression text box. Use "*" to stand for any string, and use "?" to stand for any character. Also you can leave this text box blank.

3. Select the **Case Sensitive** checkbox if you want JReport to distinguish between uppercase and lowercase characters when filtering.

4. In the Select the elements to include in the view box, select the resources you want to remain in the Catalog Browser by checking the checkbox in front of each resource.
   - If you want to select/deselect all the resources under one certain data source, check/uncheck the checkbox of the data source node.
   - If you want to select/deselect all the resources in the box, click the **Select All**/**Deselect All** button.

5. Click the **OK** button to filter the resources.

   The filter result will be the union of the resources that match the filter condition and the selected resources from the Select the elements to include in the view box.

Using a reference table to clarify resource relationships

With the development of reports and catalogs, the relationships among the catalog resources may become increasingly complicated. You might be wondering if and where an SQL or a formula is used by a resource in a catalog or report. You might hesitate to rename a resource because you don't exactly know what other resources are currently using it and, if you rename it, what result it would cause. To help you manage catalog resources, JReport provides you with a tool - the reference table.

JReport monitors the resource reference information in the catalog and stores this information in the reference table. You can select the resource types that can be registered in the reference table.

With the reference table, when you rename a resource object, JReport Designer will scan the reference relationships between it and other resource objects. It will then prompt you to update the changes in the resources that are referencing the object, helping you reduce the risk of resource not applicable.

Most resources in the catalog can be included in the reference table, such as connections (tables, views, synonyms, stored procedures, SQLs, queries, and business views), user data sources, hierarchical data sources, formulas, summaries, parameters, and reports.

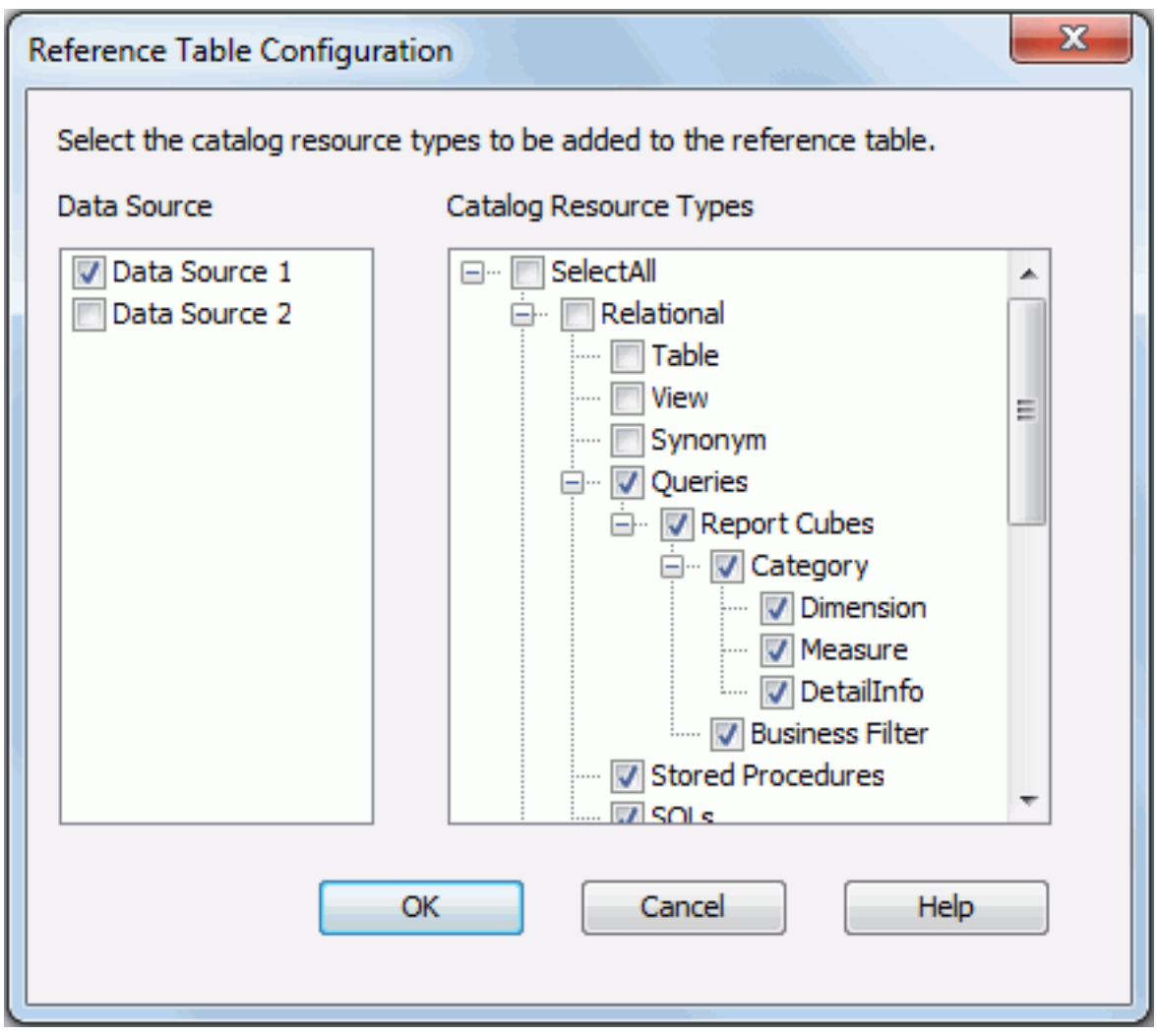
This section describes the usage of the reference table as follows:

Configuring the reference table

Before JReport can start to monitor and record the resource reference information in the reference table, you must first have the reference table configured. In the Reference Table Configuration dialog, you can choose the resource types that are to be added to the table. Then, JReport will monitor the reference relationships among objects of these resource types, and store the relationship information in the database.

To configure the reference table:

1. On the Catalog Browser toolbar, click the **Configure Reference** button to open the Reference Table Configuration dialog.



2. Check the data sources you want JReport to monitor in the Data Source box.
3. In the Catalog Resource Types box, check the resource types that are to be added to the reference table.

   In order to improve product performance and provide you with more flexibility, JReport allows you to choose the resource types that you want to monitor by yourself. Note that JReport will only monitor the referencing relationships among the resource types that you have selected. The resource types that have not been selected will not be monitored, even if they may have referencing relationships to the selected resource types. Therefore, if you are not sure that the resource types you have selected hold all the possible reference relationships within themselves, it is recommended that you select as many resource types as possible to avoid incomplete information.

   For example, in the Reference Table Configuration dialog, selecting only the formula resource type means that for a formula, only the formulas referencing this formula will be recorded in the reference table, although other resources, such as queries, summaries, and reports which are also using this formula will not be shown in the reference table. If you rename this formula, you will only be prompted to update the change in the formulas using it, ignoring any other resources also using it.
4. Click **OK**. The referencing relationships among the objects of the checked resource types will then be monitored.

Viewing a resource object's referencing relationships

After you have configured a reference table, you can view the referencing relationships for a specific resource object. To do this:

1. In the Catalog Browser, browse to the resource object with the reference relationships that you want to view.
2. Right-click the resource object, and then click **Reference Entities** on the shortcut menu.
3. The Reference Entities dialog will then appear, listing all the catalog entities that are using the selected resource object.

Note: In the Reference Entities dialog, you can right-click any of the query, formula, parameter or summary entity which uses the specified resource object to edit it, and right-click any report entity to open the report.

Refreshing and exporting the reference table

The reference table information is stored in a database, which provides reliable stability. However, in some circumstances you may need to refresh the reference table to retrieve the correct reference relationship information. To refresh the reference table, on the Catalog Browser toolbar, click the **Refresh Reference** button .

You can also export the reference table information to a text file. To do this:

1. On the Catalog Browser toolbar, click the **Export Reference** button .
2. In the Save Reference dialog, specify the location where you want to save the file.
3. Click **Save** to save the file.

Defining the join relationships between tables

After tables have been added to a catalog, you can now use them to build reports. However, in your daily work, you may often be dealing with a set of tables in your queries. You will have to define relationships and create joins among different tables every time you create a new report.

JReport Designer provides you with the pre-join feature which can be used for creating/editing queries. The pre-join information is one kind of resource information stored in the catalog. It is saved in a standalone file with the extension .pre, and shares the same prefix as the catalog file. It is not used at report runtime. In JReport Designer, when you open a catalog, the pre-join file is opened at the same time. When you save the catalog, the pre-join file is also saved.

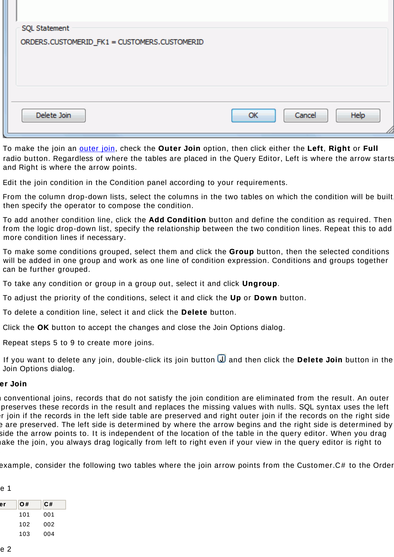
Creating joins between tables

The Pre-join Editor is a convenient tool for you to predefine the relationships between tables all at once.

To predefine joins between tables in a catalog, follow the steps below:

1. In JReport Designer, open the catalog on which you want to define join relationships.
2. In the Catalog Browser, select a data source to activate the Pre-join button on the toolbar if it is disabled.
3. Click the button, then in the Data Source dialog, select the date source on which the joins will be created and click **OK**. The Pre-join Editor window appears.
4. Click the **Arrange** button to organize the tables in the selected data source. If the data source is connected with multiple connections, all the tables obtained from these connections will be displayed in the Pre-join Editor. The queries created within the selected catalog data source will also be displayed as tables in the editor.
5. Now you can make joins among tables by using mouse drag and drop. Point to one column in the source table, press and hold the mouse button, then move the pointer to the other column in the target table and release the mouse button. A green line with a join button will be shown, linking the two columns. This represents that a join has been created.

   For tables obtained from an XML connection, if the parent and children nodes are transformed to different tables, joins will be embodied by the parent-child relationship by default, which is maintained by primary key and foreign key in the tables.
6. If you want to further edit the join, double-click its join button. The Join Options dialog appears.



7. To make the join an outer join, check the **Outer Join** option, then click either the **Left**, **Right** or **Full** radio button. Regardless of where the tables are placed in the Query Editor, Left is where the arrow starts and Right is where the arrow points.
8. Edit the join condition in the Condition panel according to your requirements.

   From the column drop-down lists, select the columns in the two tables on which the condition will be built, then specify the operator to compose the condition.

   To add another condition line, click the **Add Condition** button and define the condition as required. Then from the logic drop-down list, specify the relationship between the two condition lines. Repeat this to add more condition lines if necessary.

   To make some conditions grouped, select them and click the **Group** button, then the selected conditions will be added in one group and work as one line of condition expression. Conditions and groups together can be further grouped.

   To take any condition or group in a group out, select it and click **Ungroup**.

   To adjust the priority of the conditions, select it and click the **Up** or **Down** button.

   To delete a condition line, select it and click the **Delete** button.
9. Click the **OK** button to accept the changes and close the Join Options dialog.
10. Repeat steps 5 to 9 to create more joins.
11. If you want to delete any join, double-click its join button and then click the **Delete Join** button in the Join Options dialog.

Outer Join

With conventional joins, records that do not satisfy the join condition are eliminated from the result. An outer join preserves these records in the result and replaces the missing values with nulls. SQL syntax uses the left outer join if the records in the left side table are preserved and right outer join if the records on the right side table are preserved. The left side is determined by where the arrow begins and the right side is determined by the side the arrow points to. It is independent of the location of the table in the query editor. When you drag to make the join, you always drag logically from left to right even if your view in the query editor is right to left.

For example, consider the following two tables where the join arrow points from the Customer.C# to the Order.C#:

Table 1

Order	O#	C#
	101	001
	102	002
	103	004

Table 2

Customer	C#	Name
	001	GE
	002	IBM
	003	DELL

The inner join of `Customer.C# = Order.C#` will produce the following result:

JoinResult	O#	C#	Name
	101	001	GE
	102	002	IBM

The `Customer LEFT JOIN Order ON (Customer.C# = Order.C#)` will produce the following result:

JoinResult	O#	C#	Name
	101	001	GE
	102	002	IBM
	<null>	003	DELL

The `Customer RIGHT JOIN Order ON (Customer.C# = Order.C#)` will produce the following result:

JoinResult	O#	C#	Name
	101	001	GE
	102	002	IBM
	103	<null>	<null>

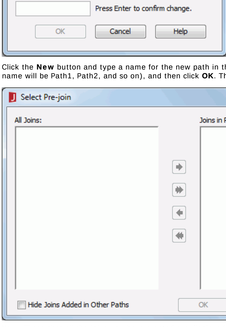
The `Customer FULL OUTER JOIN Order ON (customer.C# = order.C#)` will produce the following result:

JoinResult	O#	Order.C#	Customer.C#	Name
	101	001	001	GE
	102	002	002	IBM
	<null>	<null>	003	DELL
	103	004	<null>	<null>

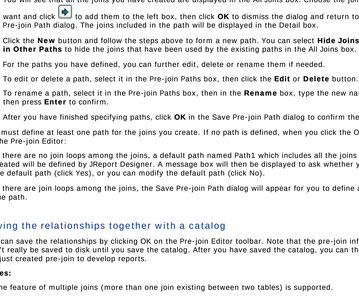
Defining join paths

After you have created joins between the tables in the Pre-join Editor window, you can then use them to define paths. To do this:

1. Click the **Paths** button on the toolbar of the Pre-join Editor window. The Save Pre-join Path dialog appears.



2. Click the **New** button and type a name for the new path in the Input Path Name dialog (by default the name will be Path1, Path2, and so on), and then click **OK**. The Select Pre-join dialog appears.



3. You will see that all the joins you have created are displayed in the All Joins box. Choose the joins you want and click to add them to the left box, then click **OK** to dismiss the dialog and return to the Save Pre-join Path dialog. The joins included in the path will be displayed in the Detail box.
4. Click the **New** button and follow the steps above to form a new path. You can select **Hide Joins Added in Other Paths** to hide the joins that have been used by the existing paths in the All Joins box.
5. For the paths you have defined, you can further edit, delete or rename them if needed.

   To edit or delete a path, select it in the Pre-join Paths box, then click the **Edit** or **Delete** button.

   To rename a path, select it in the Pre-join Paths box, then in the **Rename** box, type the new name and then press **Enter** to confirm.
6. After you have finished specifying paths, click **OK** in the Save Pre-join Path dialog to confirm the settings.

You must define at least one path for the joins you create. If no path is defined, when you click the OK button on the Pre-join Editor:

- If there are no join loops among the joins, a default path named Path1 which includes all the joins you have created will be defined by JReport Designer. A message box will then be displayed to ask whether you accept the default path (click Yes), or you can modify the default path (click No).
- If there are join loops among the joins, the Save Pre-join Path dialog will appear for you to define at least one path.

Saving the relationships together with a catalog

You can save the relationships by clicking OK on the Pre-join Editor toolbar. Note that the pre-join information won't really be saved to disk until you save the catalog. After you have saved the catalog, you can then use the just created pre-join to develop reports.

Notes:

- The feature of multiple joins (more than one join existing between two tables) is supported.
- You cannot make outer joins for XML data sources.
- Not all forms of joins are supported by all database systems. For example, MySQL does not support Full Outer Join so be sure to check your DBMS manuals.
- In JReport Designer, the joins in one path should never form a loop (any table in this path will have direct or indirect joins with all the other tables). If you specify a path which forms a loop, JReport Designer will prompt you to re-select the joins.
- You cannot define a path completely having all the same joins as with existing paths. If you have specified a path that is the same as an existing path, JReport Designer will prompt you to re-select the joins.

Maintaining catalogs

JReport Designer provides a tool - JReport Catalog Doctor for you to maintain catalogs and diagnose problems with catalogs. After reading through this chapter, you will understand how to use the Catalog Doctor to diagnose the specific components of catalogs: connections, tables and fields, formulas and summaries, and queries.

The following topics describe how to use the Catalog Doctor:

- Errors the Catalog Doctor can detect
- Using the Catalog Doctor
- Diagnosing and fixing problems

Errors the Catalog Doctor can detect

With the Catalog Doctor you can diagnose and validate data objects in a catalog. Since the data objects in a catalog are constructed independently from each other, the Catalog Doctor diagnoses each type of data object independently. For example, when it checks formulas, it only looks into the definition of parameters and DBFields in the current catalog. The system does not check whether the DBFields exist in the connected database. When the system checks the queries, it will only look for the mapping fields which are currently in the catalog, and does not confirm whether the formulas are valid.

For your convenience, in the following statements database refers to a raw database connected via JDBC or ODBC, mapped table/view refers to a table/view in a catalog, raw table/view refers to a table/view in a raw database, and raw column refers to the columns in a raw database.

Mapped table/view and DBField

- Checks whether the raw table/view of a mapped table/view exists in the database.
- Checks whether the raw column of a DBField exists in the database. If it exists, then it checks to see whether the stored information such as SQL Date type matches the raw column in the database.

Formulas and summaries

- Checks to see whether the formulas and summaries have correct syntax. It checks whether the remapped parameters and DBFields need to be remapped again.

Queries

- Checks whether the mapped tables/views referenced by the query exist in the catalog.
- Checks whether or not the mapping fields are valid.
  - Whether the DBFields of the table/view match those saved in the query.
  - Whether or not the formulas and parameters can be found in the catalog.

Using the Catalog Doctor

The following topics explain how to use the Catalog Doctor.

Starting the Catalog Doctor

In `<install_root>\bin`, there is a batch file named CatDr.bat. The Catalog Doctor can be started by running this batch file.

You can also start it by issuing the following commands from an MS-DOS command prompt:

```
>cd c:\jreport\designer\bin
c:\jreport\designer\bin\>catdr
```

Loading a catalog

To load a catalog into the Catalog Doctor for diagnosing, take the following steps:

1. Click **File** > **Open** to bring up the Open Catalog File dialog.
2. Select a catalog file, and click **Open**.

Saving a catalog

- To save the open catalog to the current file, click **File** > **Save**.
- To save the open catalog to another file, click **File** > **Save As**.

Merging catalogs

With the Catalog Doctor, you can also merge the resources from two catalogs into one. However, before merging, you need to first specify at which level the differences between these two catalogs will be checked by the Catalog Doctor.

To specify the checking level:

1. Click **File** > **Options**.
2. In the Merge Catalog box of the Options dialog, specify the checking level.
   - **Identify all differences**
     The Catalog Doctor will check for differences between the two catalogs. All the resources of the report are displayed in the Merge dialog with all conflicting resources marked.
   - **Identify critical differences**
     The Catalog Doctor will check for any differences that may cause the engine to fail to run a report. All the resources of the report are displayed in the Merge dialog with any critical conflicting resources marked.
   - **Ignore differences**
     Does not show the differences between the two catalogs. Any conflicts will remain in the target catalog.
3. Click **OK** to accept the setting.

With the checking level specified, you can then start to merge catalogs.

To merge catalogs:

1. Open the target catalog in the Catalog Doctor by clicking **File** > **Open**.
2. Click **File** > **Import**, and then select the catalog that is to be imported.
3. In the Import From Catalog dialog, select the resources from the catalog, and click **OK**.
4. The selected resources may conflict with those in the target catalog. The conflicting resources will be marked with red question marks.
5. Click the **Rename** button to rename, or the **Replace** button to replace. You can also click the **Skip** button to skip.
6. Click the **Differences** button to see the property values of the objects in the target catalog and the sources catalogs. Different property values will be highlighted.
7. Click the **Target Relation** button to see the relationship between the object and its parent node.
8. Repeat the steps above to manage conflicting resources, and then click the **Merge** button to merge the two catalogs. If the connections in the two catalogs connect to different data sources, you will be asked to modify the existing data source in the target catalog. For safety purposes, it is better to create a new data source.
9. Click **File** > **Save** to save the changes and exit the Catalog Doctor.
10. Launch JReport Designer, and open the merged catalog. The resources you just imported are now available and new data sources have been created.

Diagnosing and fixing problems

After opening a catalog, you will then be shown the Catalog Doctor. All the resources in the open catalog will be listed in the left panel of the Catalog Doctor window. Select a resource from the resource tree. In the right panel, you will see the information of the selected resource.

You should diagnose the resources in the following order: connection > tables and fields > formulas and summaries > queries.

Connection

The Catalog Doctor can be used to diagnose connections in the following aspects:

- **Finding information of the current connection**
  Select a connection from the resource tree. The connection information will be displayed in the left panel.
- **Modifying the current connection**
  Right-click the connection in the resource tree, and then click **Modify** on the shortcut menu. The Get JDBC Connection Information dialog will appear. You can now modify the connection.

  For a JDBC driver, fill in the class name of the driver, the URL, user ID and password (if necessary).
- For a JDBC-ODBC bridge, uncheck the **Driver** checkbox and check **Use ODBC Data Source**, then give the ODBC data source name, and the user ID and password (if necessary).

  You can also change the properties values in the Properties panel directly. Select a property value, and then choose or type in the value.

Tables and fields

The Catalog Doctor can be used to diagnose tables and fields in the following aspects:

- **Finding all the mapping tables, views, synonyms and DBFields in the catalog**
  In the resource tree, expand the **Relational** node, then go to the **Tables**, **Views** or **Synonyms** node and reveal it. All the mapping tables, views, synonyms and DBFields in the catalog will then be listed. The icon in front of each object has a specific meaning, which indicates whether the object is valid or not.

  The following are descriptions of the icons in front of each object:

  : Table

  : Table which does not exist in the database

  : Table which has invalid fields

  : View

  : View which does not exist in the database

  : View which has invalid fields

  : Synonym

  : Field

  : Field which does not exist

  : Field with a name which can be found in the database, but has a type mismatch.
- **Finding the stored information of a mapping table/view/synonym or a DBField**
  Select a table/view/synonym or a DBField in the resource tree to find its stored properties. The information will be listed in the Properties panel.
- **Finding all the raw tables/views/synonyms**
  Select a table/view/synonym in the Tables and Fields node. All the raw tables/views/synonyms will be shown in the Choose from following to Re-Map panel. Each item there shows the name of a table/view/synonym.
- **Getting the raw columns in a raw table/view/synonym**
  First select a mapping table/view/synonym, then choose a raw table/view/synonym from the Choose from following to Re-Map panel. The raw columns will be shown in the bottom right panel.

  > **Note:** If you select a mapping table/view/synonym with an existing raw table/view/synonym, the raw table/view/synonym will automatically be selected.
- **Deleting a set of mapping tables, views, synonyms or DBFields**
  To delete the mapping object, you can use one of the following methods:
  - Select an object from the resources tree, right-click and on the shortcut menu click **Delete**.
  - Click the **Delete** button on the toolbar after selecting the object.
  - Select the object and press **Delete** on your keyboard.

  To delete a set of mapping tables, views, synonyms or DBFields, select the objects you want to delete, and click on the toolbar (or press **Delete** on your keyboard).
- **Re-mapping (Updating) a mapping table/view/synonym**
  First select the mapping table/view/synonym that is to be updated from the resource tree. Then, select the raw table/view/synonym which the mapping table/view/synonym refers to from the Choose from following to Re-Map panel, and click the **Update** button on the toolbar.

  > **Note:** The tree is sorted by the validation status of each object (invalid objects are listed ahead). So, after you have updated a mapping table/view/synonym, the order of the objects will probably change. If the table was invalid and becomes valid after updating, it will be listed after the invalid objects.
- **Re-mapping (Updating) a DBField**
  Before updating a DBField, you must make sure that the mapping table/view/synonym of the DBField is valid. That is, the raw table/view/synonym that the mapping table/view/synonym refers to exists in the database.

  Select the DBField you want to update from the resource tree, choose the raw column you want the DBField to refer to in the bottom right panel, and then click the Update button on the toolbar.
- **Adding tables/views/synonyms**

  Click the **Add Table** button • the **Add View** button •or the Add Synonym button on the toolbar, or right-click the Tables, Views, or Synonyms node, or a table, view or synonym and select **Add Table**, **Add View** or **Add Synonym** from the shortcut menu. In the Add Table/View/synonym dialog, select the tables/views/synonyms you would like to add, and click **Add**. To finish adding, click **Done**.
- **Adding a DBField**
  Select a table or a synonym from the resource tree, right-click and on the shortcut menu click **Insert Column** to bring out the Add Column dialog. Then select the column you want and click **Add**. When you have finished adding columns, click **Done** to close the dialog.

Formulas and summaries

The Catalog Doctor can be used to diagnose formulas and summaries in the following aspects:

- **Finding all the formulas and summaries in the catalog**
  Expand the Formulas and Summaries node on the resources tree. The formulas and summaries in the catalog will be listed in the tree. They will be sorted by their validity and names. If the icon of a formula/summary is marked by a cross, it means it is invalid (it fails on compiling).
- **Finding the compile Error message of a formula/summary**
  Select the formula/summary that is to be checked. The compile error messages will then be displayed in the Error panel.
- **Replacing a mapping field with another in formulas and summaries**
  A frequent error is caused when the mapping fields are not found in a catalog. In which case, you will need to replace the mapped field in formulas/summaries with a valid one.
  1. Select the formula/summary that contains unknown fields on the resource tree.
  2. Click the **Replace All** button in the Error panel. The Replace dialog appears.
  3. Input the field name that is to be replaced in the Mapping Name text box.
  4. Click the **Browse** button, the Mapping Fields Browse dialog appears.
  5. In the Tree View tab, you can select an existing field to replace the unknown field.
  6. In the List View tab, you will see the mapping fields in alphabetical order and the type for each mapping field. If the mapping field is a DBField, you will see the tables/views/synonyms it belongs to. Select the field you want to use as a replacement and click **OK**.
  7. Click the **Replace** button in the Replace dialog.
- **Editing a formula/summaries**
  Select the formula/summary to be edited, and click the **Edit** button in the Error panel.
- **Removing formulas/summaries**
  Select the formulas/summaries that are to be removed, then, do one of the following:
  - Right-click and on the shortcut menu, click **Delete**.
  - Click the **Delete** button on the toolbar.
  - Press the **Delete** key on your keyboard.

Queries

The Catalog Doctor can be used to diagnose queries in the following aspects:

- **Finding all the queries in the catalog**
  Expand the Queries node in the Relational node. All the queries will be listed in the resources tree.
- **Finding the tables/views selected in a query**
  Select a query and expand it. All the tables/views will be listed.
- **Finding the DBFields selected from a table/view/synonym in a query**
  Select a query and expand it, then click on a table/view/synonym and expand it. All the DBFields in the table, view or synonym will be listed.
- **Finding the formulas or computed columns selected in a query**
  Formulas and computed columns are listed before mapping tables/views/synonyms.
- **Editing a query**
  Select the query that is to be edited, and click the **Edit** button in the Error panel (or right-click the query, and select **Edit Query** from the shortcut menu).
- **Deleting queries, DBFields, formulas, computed columns and a table/view/synonym selected in a query**
  To delete the object, you can use one of the following methods:
  - Select an object from the resources tree, right-click and on the shortcut menu click **Delete**.
  - Click the **Delete** button on the toolbar after selecting the object.
  - Select the object and press **Delete** on your keyboard.
- **Saving a query as a cached query result**
  Select the query that is to be saved as a cached query result, and click **Run** (or right-click a query, and on the shortcut menu, click **Create Cached Query Result**).

Data Source Connections

JReport Designer requires access to your data source so that it can import the database schema, which allows JReport to build the queries that provide the data for reports. Data source connections are defined, named, and stored in the catalog along with the reports and other resources that use them.

JReport is extremely efficient when working with data sources that are in a relational database software. The Java Database Connectivity (JDBC) and Java Database Connectivity - Open Database Connectivity (JDBC-ODBC) connections are the basic approaches for connecting to a database for data. Other data source types can also be accessed.

This chapter describes the following categories of data source connection-related tasks:

- JDBC connections: Connects to a relational database via a JDBC driver or JDBC-ODBC bridge.
- XML connections: Connects to and transforms an XML hierarchy model to a relational model.
- User defined data sources: Through the UDS API, JReport Designer can access data from an external data source, such as a text file or Lotus Notes, which is not stored in a database or when there is no JDBC driver available.
- Hierarchical data sources: JReport Designer directly supports XML format data source by wrapping the provided HDS API. JReport Designer's built-in classes can implement the XML format hierarchical data source interface. You can directly import an XML data source to a catalog using the Catalog Browser.
- Web service data sources: JReport Designer supports SOAP web services defined by WSDL 1.1 or WSDL 2.0 as a data source. You can directly add a SOAP web service data source to a catalog by importing a WSDL file.
- MongoDB connections: Connects to a MongoDB data source and transforms the collection schemas in the data source to relational schemas.
- HIVE connections: Connects to a relational database stored in a HIVE data source via a JDBC connection.
- JSON connections: Connects to a JSON data source and transforms the schemas in the data source to relational schemas.

See an example: The SampleComponents catalog, included with JReport Designer, contains reports that have examples of different data source connections. Open the following report to see the connection examples: `<install_root>\Demo\Reports\SampleComponents\DataSourceConnections.cls`.

JDBC connections

JDBC is a Java API for executing SQL statements. It consists of a set of classes and interfaces written in the Java programming language which enables SQL statements to be executed against virtually any relational database.

The JDBC-ODBC bridge allows ODBC drivers to be used as JDBC drivers. This bridge is a JDBC driver which implements JDBC operations by translating them into ODBC operations. To ODBC it appears as a normal application program.

Notes:

- In JReport Designer, not all the data types supported by JDBC can be used. The data types that supported in JReport Designer are: Bit, Tinyint, Smallint, Integer, Bigint, Float, Real, Double, Number, Decimal, Char, VarChar, LongVarChar, Date, Time, TimeStamp, Binary, LongVarBinary, Blob, Clob, Array, nVarChar, and Distinct.
- For the Blob, Clob, and Array types, you cannot browse their data in the Catalog Browser. And for the Distinct type, due to a limitation of JDK, the based SQL Type cannot be obtained. So you need to indicate it manually after you import these types of data.
- The JDBC-ODBC bridge is not supported by JDK 8.

This chapter describes the following tasks related to JDBC connections:

- Setting up the JDBC driver
- Setting up a JDBC connection
- Setting up OOJDBC connections
- DBMS objects in a JDBC connection

Setting up the JDBC driver

JReport supports all of the current mainstream databases as well as most databases which support ODBC or JDBC drivers. For more information, refer to Supported report databases.

To connect to a database via a JDBC driver:

1. Install the JDBC driver according to the instructions provided by the JDBC driver supplier.
2. Edit setenv.bat for Windows or setenv.sh for Unix in `<designer_install_root>\bin`, by appending the class path for the JDBC driver.

Note: The step for appending the class path is very important. The same changes made to JReport Designer's class path must be made to the class path for JReport Server too. A missing JDBC driver in the JReport start-up batch file or command line will result in a "ClassNotFoundError message" when you try to run a report. Meanwhile, if you want to use the Preview as Page/Web Report Result feature in JReport Designer, you also need to append the class path for the JDBC driver to setenv.bat/setenv.sh in `<designer_install_root>\server\bin`.

The following are two examples of setting up the JDBC driver.

Example 1: Installing a Sybase JDBC driver

1. Install Sybase JDBC driver jconn2.jar.
2. Modify setenv.bat by appending the archive file path and file name of the driver to the set ADDCLASSPATH system variable:

   `set ADDCLASSPATH=%JAVAHOME%\lib\tools.jar;`**`c:\jconnect\jconn2.jar;`**

Example 2: Installing an Oracle JDBC driver

1. Install Oracle JDBC driver ojdbc6.jar.
2. Modify setenv.bat by appending the archive file path and file name of the driver to the set ADDCLASSPATH system variable:

   `set ADDCLASSPATH=%JAVAHOME%\lib\tools.jar;`**`c:\orc\thin\lib\ojdbc6.jar;`**

Limiting the size of the fetch data buffer

If you are using a database which supports the JDBC method Statement.setFetchSize(), you can request the database to retrieve a specified number of rows in each read instead of all rows which minimizes memory usage. This can improve performance. It can also avoid Java heap out of memory errors when doing large queries. JReport provides a property setFetchSize in the JdbcDriversConfig.properties file for you to do this, which is located in the `<install_root>\bin` directory.

The following is an example of setting the setFetchSize property in the JdbcDriversConfig.properties file and detailed description of each property:

```
DriverName_D1 = MySQL-AB JDBC Driver
Vendor_D1 = MySql.com
Version_D1 = 3.0.8-stable ($Date: 2003/05/19 00:57:19 $, $Revision: 1.27.2.18 $)
setFetchSize_D1 = 5000
```

- **DriverName_[Name]**
  Database driver name that should be used to establish a connection. You can use connection.getMetaData().getDriverName() to obtain the value.
- **Vendor_[Name]**
  The vendor of current JDBC driver. It is a string value.
- **Version_[Name]**
  The version of current JDBC driver. You can use connection.getMetaData().getDriverVersion() to obtain the value.
- **setFetchSize_[Name]**
  The number of rows retrieved in each buffer from the database. It is an Integer value that should be larger than 0 and less than or equal to getMaxRows(). The default value is different according to the driver in use. You can refer to your own database driver specification.
- **_[Name]**
  It is a user-defined name used to mark a group of driver settings different from the other groups.

Specifying where to implement the maximum query run time and number of records

In the JdbcDriversConfig.properties file, you can also specify where the properties, Maximum Rows and Maximum Duration, will be implemented, on JReport side or on the database side. For details about usage of these two properties, see Limiting the query run time and number of records.

The following is an example of setting the two properties:

```
DriverName_D2 = Oracle JDBC driver
Vendor_D2 = Oracle
Version_D2 = 9.2.0.3.0
supportMaxRowPushDown_D2 = true
supportMaxDurPushDown_D2 = true
```

- **DriverName_[Name]**
  Database driver name that should be used to establish a connection. You can use connection.getMetaData().getDriverName() to obtain the value.
- **Vendor_[Name]**
  The vendor of current JDBC driver. It is a string value.
- **Version_[Name]**
  The version of current JDBC driver. You can use connection.getMetaData().getDriverVersion() to obtain the value.
- **SupportMaxRowPushDown_[Name]**
  Specifies where the property Maximum Rows will be implemented, on JReport side or on the database side. By default, it is set to true, which means the property will take effect on the database side. If it is set to false, the property will take effect on JReport side.
- **SupportMaxDurPushDown_[Name]**
  Specifies where the property Maximum Duration will be implemented, on JReport side or on the database side. By default, it is set to true, which means the property will take effect on the database side. If it is set to false, the property will take effect on JReport side.
- **_[Name]**
  It is a user-defined name used to mark a group of driver settings different from the other groups.

Notes:

- If there are more than one group with the same group marking name, the last group will be adopted.
- If the sign # is seen before "DriverName" of a group, or if setFetchSize is given a negative value, the whole group will be disabled.

Setting up a JDBC connection

This procedure assumes you have set up the JDBC driver as described in the previous task.

Note: The URL format is regulated by the driver itself. That is, different drivers have different URL formats. The format of the driver template in JReport Designer jdbcdrivers.properties is `jdbc.drivers=JDBCDriverName:JDBCDriverName:...`

where, JDBCDriverName is the JDBC driver name that can be auto-loaded when JReport starts up and "**:**" is the delimiter between two driver names. JReport will search from the first and find one that can work with the driver, URL, and user name and password in the catalog.

Below is an example of the jdbcdrivers.properties file which specifies an Oracle thin driver and an Interbase thin driver:

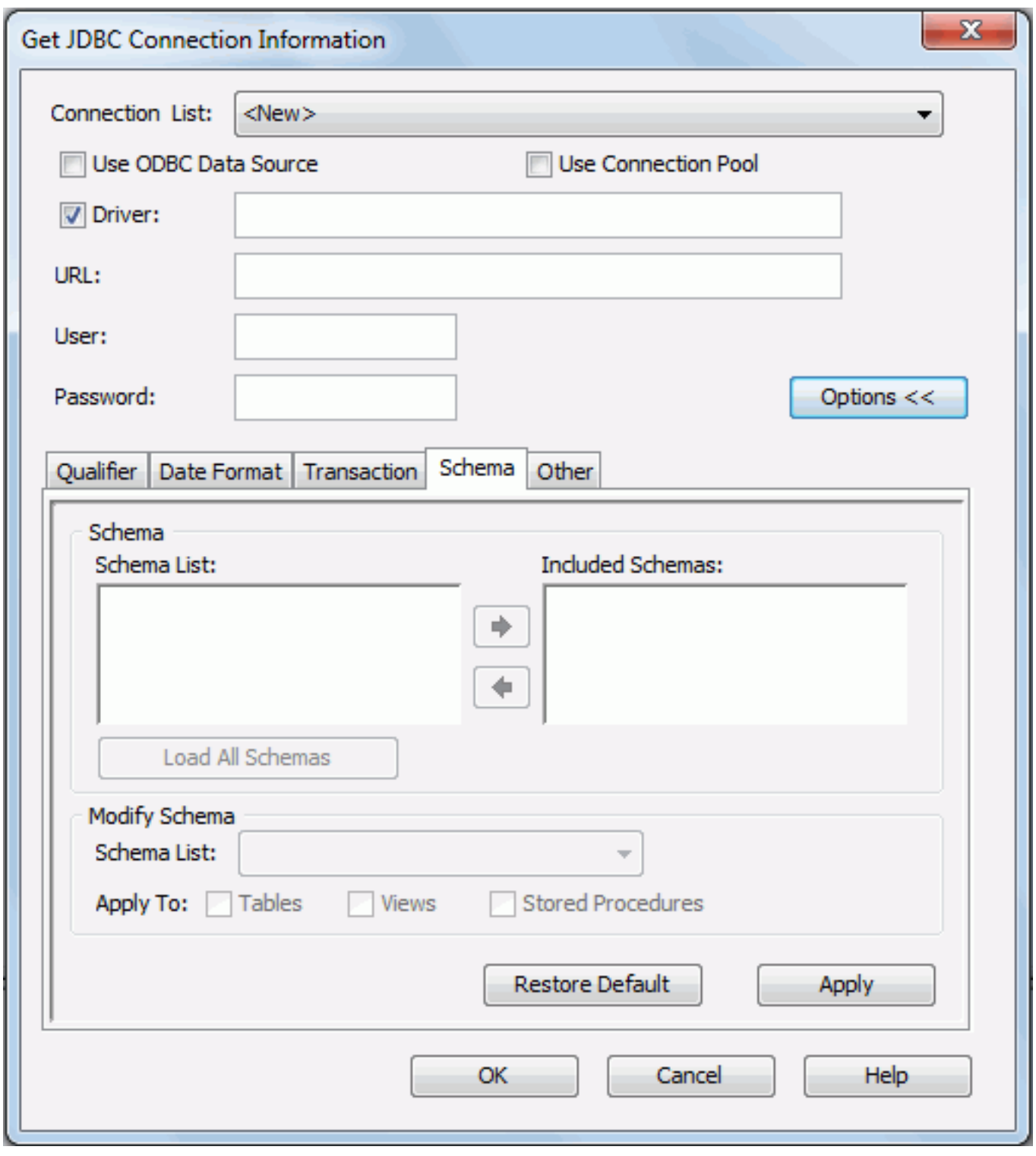
```
jdbc.drivers=oracle.jdbc.driver.OracleDriver:interbase.interclient.Driver
```

To set up a JDBC connection:

1. Create a catalog or open a catalog, then in the Data tab of the Catalog Browser, right-click the node of a data source and choose **Add JDBC Connection** from the shortcut menu.

   If you want to set up the connection by adding a data source to an existing catalog and then defining the connection on the data source, open the catalog, select any data source contained in it, click the **New Data Source** button on the Catalog Browser toolbar, then in the New Data source dialog, specify the name of the data source, select the **JDBC** connection type and click **OK**.

2. In the Get JDBC Connection Information dialog, specify the options as required.



3. Click **OK** to set up the connection, and you will be prompted with a message box showing the status of connecting to the JDBC driver. Then you can add tables stored in the database to the catalog in the displayed Add Table dialog.

You have four ways to set up a JDBC connection:

- **Setting up by adding an existing connection**
  1. In the Get JDBC Connection Information dialog, select the connection from the Connection List. This list contains previously added connection information.
  2. The JDBC URL and JDBC driver name will then appear in the corresponding text fields. Click **OK**.

     Note: The format of the connection information should be JDBC URL/(JDBC Driver Name). For example: `jdbc:odbc:jinfonet/(sun.jdbc.odbc.JdbcOdbcDriver)`.

- **Setting up an ODBC connection**
  1. In the Get JDBC Connection Information dialog, uncheck the **Driver** checkbox and check **Use ODBC Data Source** box.
  2. Enter the ODBC data source name in the DSN Name field.
  3. Input the user name and password to enable accessing the database through the ODBC data source, and then click **OK.**

     Note: The JDK that JReport uses must match the ODBC data source that the operating system uses. For example, 32-bit ODBC data source can be connected by 32-bit JDK only.

- **Setting up a JDBC connection**
  1. In the Get JDBC Connection Information dialog, specify the driver in the Driver text field.
  2. In the URL text field, specify the URL of the JDBC driver.
  3. Input the user name and password respectively, and then click **OK**.

     Tip: After entering the JDBC URL, user name and password, you can leave the Driver text field empty and instead supply it in the property file - jdbcdrivers.properties.

- **Setting up a connection via the WebLogic 6.1 connection pool**
  1. In the Get JDBC Connection Information dialog, check the **Use Connection Pool** checkbox. The string weblogic.jinfonet.pool.Driver will automatically be displayed in the Driver text field.
  2. In the URL text field, specify the URL of the JDBC driver.
  3. Input the user name and password respectively, and then click **OK**.

Tips:

- Upon the finishing of setting up the connection, the Add Table dialog will be displayed, prompting you to add tables from the database to the connection. If you want to add tables at a later date, click Done.
- When creating any of the above connections, if your database has some special requirements, you can click the Options button to modify the options according to your requirements.

The following examples show how to set up connections via specific JDBC drivers:

- Example 1: Setting up a connection via the Oracle JDBC driver
- Example 2: Setting up a connection via the Sybase JDBC driver
- Example 3: Setting up a connection via WebLogic 6.1 Connection Pool
- Example 4: Setting up a connection via Vertica
- Example 5: Setting up a connection via Amazon RDS
- Example 6: Setting up a connection via RedShift

Example 1: Setting up a connection via the Oracle JDBC driver

Assume that:

- You have already installed the Oracle JDBC driver, and have appended the archive file (ojdbc6.jar) of the thin driver to the ADDCLASSPATH variable in the file setenv.bat in `<designer_install_root>\bin`.
- The Oracle database server is with the name p02_wuhb, port number 1521, database orcl, user name scott and password tiger.
- A catalog has been created with a default data source.

Take the following steps to set up a connection which connects JReport Designer to a database via the Oracle JDBC driver:

1. Start JReport Designer and open the catalog.
2. In the Data tab of the Catalog Browser, right-click the node of the default data source and select **Add JDBC Connection** from the shortcut menu.
3. In the Get JDBC Connection Information dialog, enter the JDBC driver class name **oracle.jdbc.OracleDriver** in the Driver text field.
4. In the URL text field, specify the URL in the format jdbc:oracle:thin:@<hostname>:<port>:<SID>. In this example, enter **jdbc:oracle:thin:@p02_wuhb:1521:orcl**.
5. Input the user name **scott** and password **tiger** respectively.
6. Click **OK** to set up the connection.

Example 2: Setting up a connection via the Sybase JDBC driver

Assume that:

- You have already installed the Sybase JDBC driver and have appended the archive file of the driver to the ADDCLASSPATH variable in the file setenv.bat in `<designer_install_root>\bin`.
- The Sybase database server is with the name SybSever, port number 5000, database pubs2, user name sa and password 123456.
- A catalog has been created with a default data source.

Take the following steps to set up a connection which connects JReport Designer to a database via the Sybase JDBC driver:

1. Start JReport Designer and open the catalog.
2. In the Data tab of the Catalog Browser, right-click the node of the default data source and select **Add JDBC Connection** from the shortcut menu.
3. In the Get JDBC Connection Information dialog, enter the JDBC driver class name **com.sybase.jdbc.SybDriver** into the Driver text field.
4. In the URL text field, specify the URL in the format jdbc:sybase:Tds:<hostname>:<port>/<SID>. In this example, enter **jdbc:sybase:Tds:SybServer:5000/pubs2**.
5. Input the user name **sa** and password **123456** respectively.
6. Click **OK** to set up the connection.

Example 3: Setting up a connection via WebLogic 6.1 Connection Pool

Assume that a catalog has been created with a default data source in JReport Designer. Take the following steps to set up a connection which connects JReport Designer to a database via the WebLogic 6.1 Connection Pool:

1. Set up the WebLogic Connection Pool.
   a. Assume that the WebLogic Server has been started, access the console through a web browser (`http://host:7001/console`). Then, go to the left panel, and expand the **JDBC** node.
   b. Click the **Connection Pools** node. All the defined connection pools will be displayed in the right panel.
   c. Click the **Configure a New JDBC Connection Pool** link.
   d. In the Configuration tab, define the connection pool as follows. Then, click the **Create** button.

      Name: jinfonet
      URL: jdbc:odbc:jinfonet
      Driver Classname: sun.jdbc.odbc.JdbcOdbcDriver
2. Check the Connection Pool.

   In the Monitoring tab, click the link **Monitor all Active Pools** to check if the pool is active. You will see the connection pool that you just created appear in the Monitoring tab.
3. Click the **Targets** tab, and add **examplesServer** to the **Chosen** column (in order to assign the connection pool to the server). Then, click **Apply** to save your changes.
4. Set up the data source.
   a. Click the **Data Sources** node in the JDBC node. In the right panel, click the **Configure a New JDBC Data Source** link.
   b. In the Configuration tab, enter the values in the following fields as follows:

      Name: jinfonetDS
      JNDI Name: jinfonetDS
      Pool Name: jinfonet
   c. Click **Create**. The new data source will then be added in the Data Sources node in the left panel.
5. Click the **Targets** tab and add **examplesServer** in the Available column to the Chosen column. Then, click **Apply** to save the changes.
6. Add weblogic.jar to setenv.bat.

   To connect via the WebLogic Connection Pool in JReport Designer, you will first need to add weblogic.jar in `<weblogic_install_root>\wlserver6.1\lib` to the ADDCLASSPATH variable of setenv.bat. In this example, supposing that you have installed JReport Designer in `C:\JReport\Designer`, and WebLogic in `C:\bea`, you will have to modify setenv.bat in `C:\JReport\Designer\bin` to add `C:\bea\wlserver6.1\lib\weblogic.jar` to the ADDCLASSPATH variable as follows:

   `set ADDCLASSPATH=%JAVAHOME%\lib\tools.jar;`**`C:\bea\wlserver6.1\lib\weblogic.jar;`**
7. Set up the connection.
   a. Start JReport Designer and open the catalog.
   b. In the Data tab of the Catalog Browser, right-click the node of the default data source and select **Add JDBC Connection** from the shortcut menu.
   c. In the Get JDBC Connection Information dialog, check **Use Connection Pool**. The string weblogic.jinfonet.pool.Driver will automatically be displayed in the Driver text field.
   d. Fill in the URL **jdbc:weblogic:jinfonet:@<hostname>:7001:jinfonetDS**, where @<hostname> is the host name or IP address of the WebLogic Server, 7001 is the port on which the WebLogic Server listens, and jinfonetDS is the JNDI Name of the data source. Then, click **OK**.

Note: The data types longbarbinary and BigDecimal are not supported when a WebLogic Connection Pool is used.

Example 4: Setting up a connection via Vertica

Assume that:

- You have already installed the Vertica JDBC driver, and have appended the archive file (vertica-jdk5-6.1.2-0.jar) of the driver to the ADDCLASSPATH variable in the file setenv.bat in `<designer_install_root>\bin`.
- The Vertica database server is with the following information:

  Database Name: vmartdb
  Driver: com.vertica.jdbc.Driver
  URL: jdbc:vertica://192.168.0.1:5433/vmartdb
  User: dbadmin
  Password: test1234

- A catalog has been created with a default data source.

For details about how to install Vertica, refer to https://my.vertica.com/docs/6.1.x/HTML/index.htm#1318.htm.

Take the following steps to set up a connection which connects JReport Designer to a database via Vertica:

1. Start JReport Designer and open the catalog.
2. In the Data tab of the Catalog Browser, right-click the node of the default data source and select **Add JDBC Connection** from the shortcut menu.
3. In the Get JDBC Connection Information dialog, enter the JDBC driver class name **com.vertica.jdbc.Driver** in the Driver text field.
4. In the URL text field, specify the URL in the format jdbc:vertica://<hostname>:<port>/<SID>. In this example, enter **jdbc:vertica://192.168.0.1:5433/vmartdb**.
5. Input the user name **dbadmin** and password **test1234** respectively.
6. Click **OK** to set up the connection.

Example 5: Setting up a connection via Amazon RDS

Assume that:

- You have already installed the JDBC driver, and have appended the archive file (mysql-connector-java-5.0.4-bin.jar) of the driver to the ADDCLASSPATH variable in the file setenv.bat in `<designer_install_root>\bin`.
- The RDS database server is with the following information:

  Driver: com.mysql.jdbc.Driver
  URL: jdbc:mysql://jrdbtest.c4fb8hiicidz.us-west-2.rds.amazonaws.com:3306/sampledb1110 (The URL is dynamically generated when you apply an instance.)
  User: dbadmin
  Password: test1234
- A catalog has been created with a default data source.

For details about how to install Amazon RDS, refer to Getting Started with Amazon RDS.

Take the following steps to set up a connection which connects JReport Designer to a database via Amazon RDS:

1. Start JReport Designer and open the catalog.
2. In the Data tab of the Catalog Browser, right-click the node of the default data source and select **Add JDBC Connection** from the shortcut menu.
3. In the Get JDBC Connection Information dialog, enter the JDBC driver class name **com.mysql.jdbc.Driver** in the Driver text field.
4. In the URL text field, specify the URL as **jdbc:mysql://jrdbtest.c4fb8hiicidz.us-west-2.rds.amazonaws.com:3306/sampledb1110**.
5. Input the user name **dbadmin** and password **test1234** respectively.
6. Click **OK** to set up the connection.

Notes:

- It is not recommended to use RDS for small data query since it costs longer time on the cloud.
- The RDS MySQL database is case sensitive, which may result in that JReport sample reports cannot run.

Example 6: Setting up a connection via RedShift

Assume that:

- You have already installed the PostgreSQL JDBC driver, and have appended the archive file (postgresql-8.4-703.jdbc4.jar) of the driver to the ADDCLASSPATH variable in the file setenv.bat in `<designer_install_root>\bin`.
- The RedShift database server is with the following information:

  Driver: org.postgresql.Driver
  URL: jdbc:postgresql://jinfonet-rsdw-demo.cfcn5ogc14yr.us-east-1.redshift.amazonaws.com:5439/sampledb (The URL is dynamically generated when you apply an instance.)
  User: dbadmin
  Password: test1234

- A catalog has been created with a default data source.

For details about how to install RedShift, refer to http://docs.aws.amazon.com/redshift/latest/dg/c_intro_to_admin.html.

Take the following steps to set up a connection which connects JReport Designer to a database via RedShift:

1. Start JReport Designer and open the catalog.
2. In the Data tab of the Catalog Browser, right-click the node of the default data source and select **Add JDBC Connection** from the shortcut menu.
3. In the Get JDBC Connection Information dialog, enter the JDBC driver class name **org.postgresql.Driver** in the Driver text field.
4. In the URL text field, specify the URL as **jdbc:postgresql://jinfonet-rsdw-demo.cfcn5ogc14yr.us-east-1.redshift.amazonaws.com:5439/sampledb**.
5. Input the user name **dbadmin** and password **test1234** respectively.
6. Click **OK** to set up the connection.

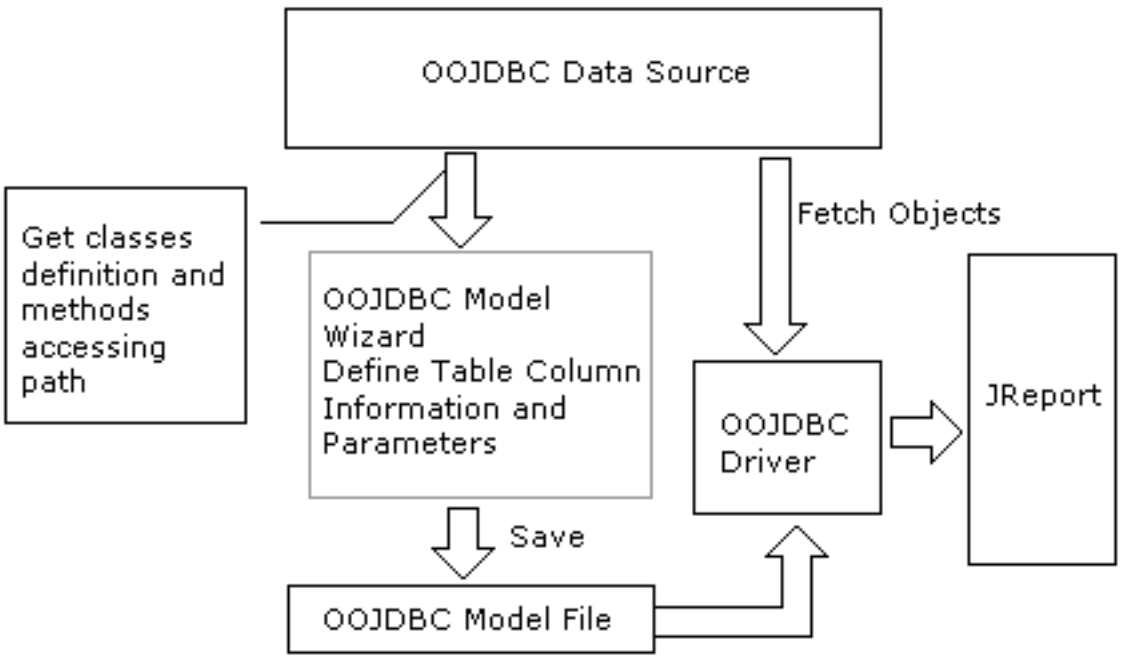
Notes:

- Redshift does not support the "Double" data type and it uses "Decimal" or "Double Precision" instead.
- Redshift does not support "Bytea", "Bit(N)" or "Bit varying (N)" data type, and so far there is no alternative data type for that, therefore JReport binary data fields like photos stored in the demo database cannot be imported to Redshift.
- For detailed features, functions, and data types that are not supported in Redshift, see http://docs.aws.amazon.com/redshift/latest/dg/c_redshift-and-postgres-sql.html.

Setting up OOJDBC connections

OOJDBC (Object Oriented Java Database Connection) is defined to cover an OODS (Object Oriented Data Source) with a JDBC interface. It is similar to a JDBC driver and is based on Java application, EJB (Enterprise Java Bean), LDAP (Light Weight Directory Access Protocol), which can be defined as a database. JReport has also developed a tool named Model Wizard, which you can run to import data via XML data source.

Here is the main frame for OOJDBC:



OOJDBC includes Model Wizard and OOJDBC Driver.

- **Model Wizard**
  Open/Save model file
  Define a Java class as a table
  Define a method as a column
  How to access the instance of the class

- **OOJDBC Driver**
  Read the table/column definitions from model file
  Return meta-data to JReport
  Construct QUERY data
  Return record to JReport

The following sections explain how to import different kinds of data sources via OOJDBC into a JReport catalog:

- Using XML file as a data source via OOJDBC
- Using class file as a data source via OOJDBC
- Using EJB as a data source via OOJDBC
- Using CSV file as a data source via OOJDBC

Using XML file as a data source via OOJDBC

JReport Designer supports XML as a data source which can be accessed via OOJDBC.

The following examples explain how to use XML file as a data source via OOJDBC:

- Example 1: Importing the XML file indirectly
- Example 2: Importing the XML file directly
- Example 3: XML file with dynamic parameters

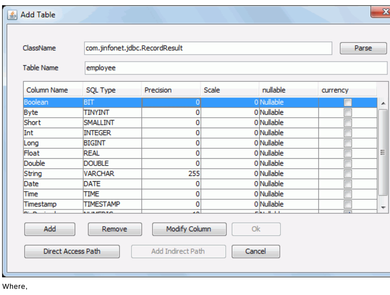
Example 1: Importing the XML file indirectly

In this example, employee.xml in `<install_root>\help\samples\OOJDBC` will be used. Store it to `<install_root>\Demo\Reports\TutorialReports` first. Then follow the steps below to import the file to a catalog.

Importing the XML file using the Model Wizard

To import the XML file employee.xml using the Model Wizard, take the following steps:

1. Run ModelWizard.bat in `<install_root>\bin`.
2. In the Object Source Wizard, click **File** > **Add Table**.
3. In the Add Table dialog, specify the class and table name as required. Here, we fill in the Table Name as **employee** and ClassName as **com.jinfonet.jdbc.RecordResult** (a class of Jinfonet). Then click the **Parse** button.

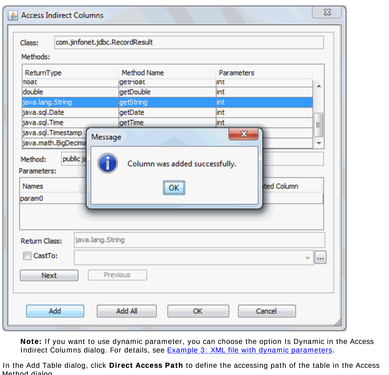


Where,

Options	Description
Column Name	Methods of the class you have defined.
SQL Type	Returned type of your column name. You can double-click the field and choose the SQL type you want.
Precision	Precision of the current column. You can click on the field and change it manually.
Scale	Scale of the current column. You can also click on the field and change it manually.
Nullable	Current status of the column (null or not). You can double-click on the field and choose the current status: Nullable, No Nulls or Nullable Unknown.
Currency	If your DBField contains $, you should check it.
Add	Adds new columns.
Remove	Removes the selected column.
Modify Column	Modifies parameters of the current table.
OK	Adds the table.
Direct Access Path	Specifies the path for the table. OOJDBC driver can save and restore the table you have defined through this path.
Add Indirect Path	
Cancel	Cancels the operation and closes the Add Table dialog.

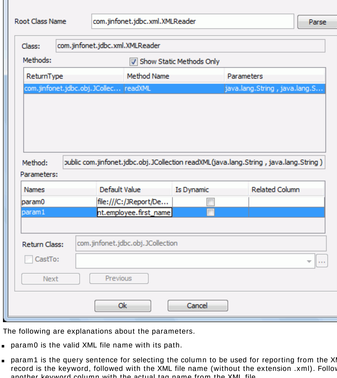
4. Click the **Remove** button to get rid of all the existing columns in the Add Table dialog, and then click the **Add** button to add a new column.
5. In the Access Indirect Columns dialog, specify the data type of the column.

   In this example, we will add a column of String type, so click **java.lang.String** in the ReturnType column of the Methods box, then click the **Add** button. Click **OK** to return to the Add Table dialog.



   Note: If you want to use dynamic parameter, you can choose the option Is Dynamic in the Access Indirect Columns dialog. For details, see Example 3: XML file with dynamic parameters.

6. In the Add Table dialog, click **Direct Access Path** to define the accessing path of the table in the Access Method dialog.
   a. Fill in **com.jinfonet.jdbc.xml.XMLReader** in the Root Class Name text field, click the **Parse** button and check **Show Static Methods Only**.
   b. Click the shown method. In the Parameters box, fill in the default values for param0 and param1 as required. For example, for param0, fill in **file:///C:/JReport/Designer/Demo/Reports/TutorialReports/employee.xml**, then for param1, fill in **record employee root employee column first_name 12 path document.employee.first_name**.



   The following are explanations about the parameters.

   - param0 is the valid XML file name with its path.
   - param1 is the query sentence for selecting the column to be used for reporting from the XML file. record is the keyword, followed with the XML file name (without the extension .xml). Following this is another keyword column with the actual tag name from the XML file.

     Below is the part quoted from employee.xml:

     ```
     <employee>
     <id>0799</id>
     <first_name>Jack</first_name>
     <second_name>Smith</second_name>
     <birthday>12/20/75</birthday>
     <title>manager</title>
     <salary>$7000.00</salary>
     </employee>
     ```

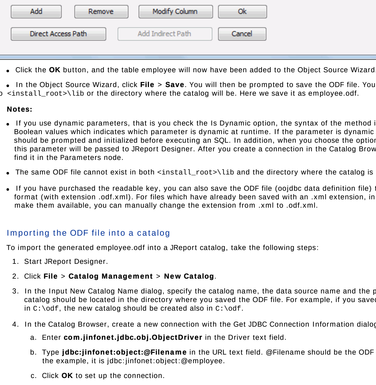
     Since first_name is required, you should then give the actual tag name here. The number afterwards is the SQL type of the column. "12" is for String type. For detailed information about Java SQL types, refer to JDK package java.sql.types.java.

     Notes:

     - In param1, the order of the columns is dependent on the default values given when adding the column.
     - The SQL type BINARY, VARBINARY and LONGVARBINARY are not supported.

7. Click **OK** to return to the Add Table dialog.

- Now you can see that the column has been added in the Add Table dialog. You can rename the default column name by editing the Column Name column. You are recommended to give the name as the actual tag definition in .xml file, for example first_name.



- Click the **OK** button, and the table employee will now have been added to the Object Source Wizard.
- In the Object Source Wizard, click **File** > **Save**. You will then be prompted to save the ODF file. You should save `<install_root>\lib` or the directory where the catalog will be. Here we save it as employee.odf.

Notes:

- If you use dynamic parameters, that is you check the Is Dynamic option, the syntax of the method is an array of Boolean values which indicates which parameter is dynamic at runtime. If the parameter is dynamic the value should be prompted and initialized before executing an SQL. In addition, when you choose the option Is Dynamic, this parameter will be passed to JReport Designer. After you create a connection in the Catalog Browser, you will find it in the Parameters node.
- The same ODF file cannot exist in both `<install_root>\lib` and the directory where the catalog is located.
- If you have purchased the readable key, you can also save the ODF file (oojdbc data definition file) to XML format (with extension .odf.xml). For files which have already been saved with an .xml extension, in order to make them available, you can manually change the extension from .xml to .odf.xml.

Importing the ODF file into a catalog

To import the generated employee.odf into a JReport catalog, take the following steps:

1. Start JReport Designer.
2. Click **File** > **Catalog Management** > **New Catalog**.
3. In the Input New Catalog Name dialog, specify the catalog name, the data source name and the path. The catalog should be located in the directory where you saved the ODF file. For example, if you saved the ODF file in `C:\odf`, the new catalog should be created also in `C:\odf`.
4. In the Catalog Browser, create a new connection with the Get JDBC Connection Information dialog as follows:
   a. Enter **com.jinfonet.jdbc.obj.ObjectDriver** in the Driver text field.
   b. Type **jdbc:jinfonet:object:@Filename** in the URL text field. @Filename should be the ODF file name. In the example, it is jdbc:jinfonet:object:@employee.
   c. Click **OK** to set up the connection.
5. In the Add Table dialog, select the table **employee** and click **Add**.
6. Click **Done** to close the Add Table dialog.

Now, the table employee will have been added into the catalog. You can use it to develop reports as required.

Note: When you publish the catalog and its reports to JReport Server, whether the ODF file is located in `<install_root>\lib` or in the directory where the catalog is located, it will be deployed to the path where the catalog is located in the server side.

Example 2: Importing the XML file directly

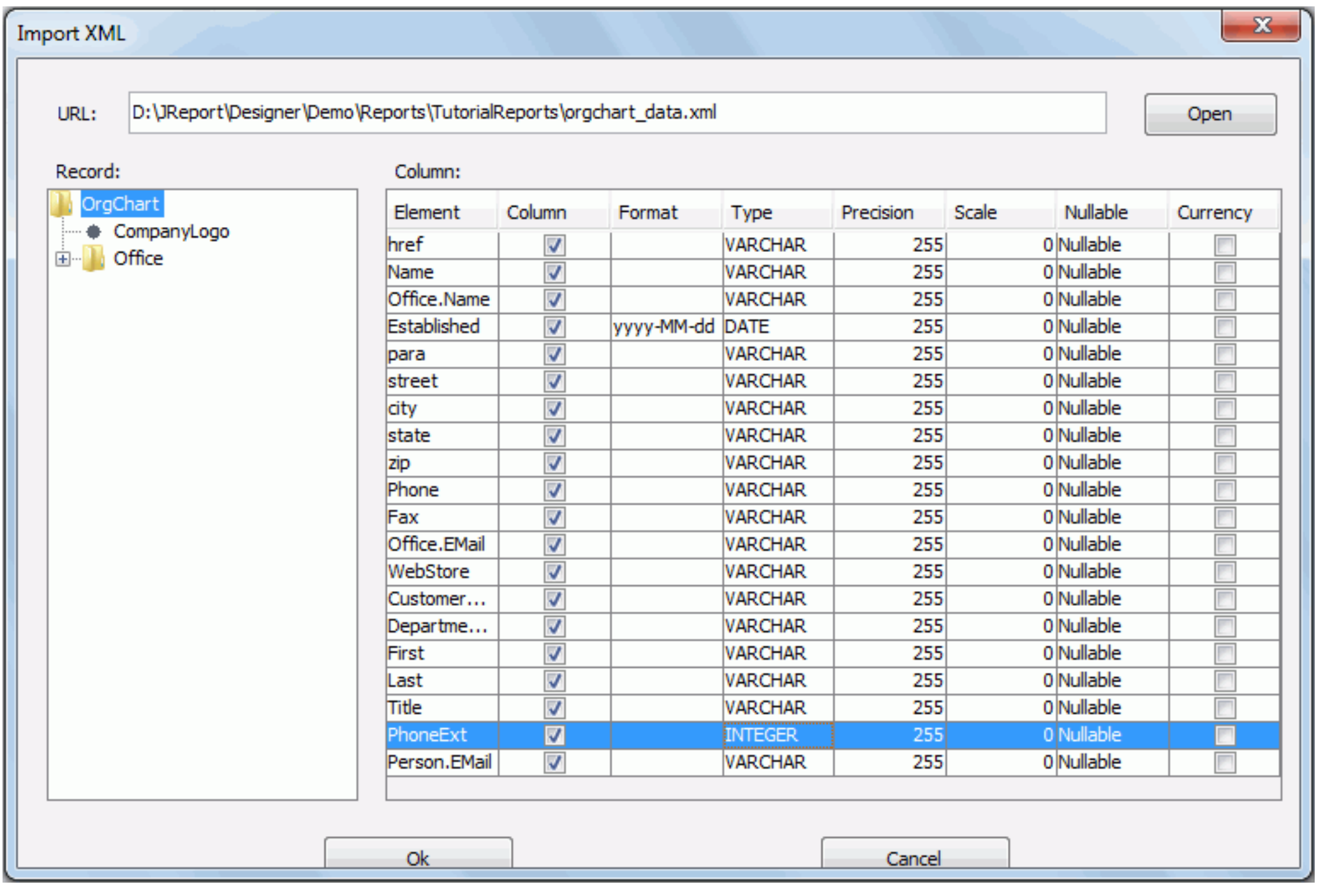
The process of importing an XML file with Model Wizard (as explained in Example 1) can be simplified with JReport's built-in XML file import functionality, and is very easy to use.

In this example, orgchart_data.xml in `<install_root>\help\samples\OOJDBC` will be used. Store it to `<install_root>\Demo\Reports\TutorialReports` first. Then follow the steps below to import the file to a catalog.

Importing the XML file directly using the Model Wizard

To import the XML file directly using the Model Wizard, take the following steps:

1. Run ModelWizard.bat in `<install_root>\bin`. The Object Source Wizard will then appear.
2. Select **File** > **Import XML Files** to bring up the Import XML dialog.
3. Click **Open** to browse to the file orgchart_data.xml.
4. Check **Column** for all elements and change their format and type if necessary (for the details about how to set the format, see Examples of format).



5. Click **OK** and the table OrgChart will now have been added to the Object Source Wizard.
6. In the Object Source Wizard, click **File** > **Save**. You will then be prompted to save the ODF file. You should save it to `<install_root>\lib` or to the directory where the catalog will be located. Here, we save it as orgchart.odf.

   Note: The same ODF file cannot exist in both `<install_root>\lib` and the directory where the catalog will be located.

Examples of format

When using certain SQL types, you should input a standard format in the Format cell in the Import XML dialog. There are some examples:

- If the time format is like Tue Aug 27 16:04:00 BST 2002, you should select:
  Type: TimeStamp
  Format: EEE MMM dd hh:mxm:ss zzz yyyy
- If the time format is like 2002-11-22 11:09:08, you should select:
  Type: TimeStamp
  Format: yyyy-MM-dd hh:mm:ss
- If the time format is like 11/18/2002 08:20:00, you should select
  Type: TimeStamp
  Format: M/d/yyyy hh:mm:ss
- If the time format is like 1999-12-05, you should select:
  Type: date
  Format: yyyy-MM-dd
- If the time format is like 08:20:00, you should select:
  Type: time
  Format: hh:mm:ss
- If the number is like 50,600, you should select:
  Type: integer
  Format: ###,###
- If the currency is like $50,600, you should select:
  Type: integer
  Format: $###,###

Importing the ODF file into a catalog

To import the generated orgchart.odf file into a JReport catalog, take the following steps:

1. Start JReport Designer.
2. Click **File** > **Catalog Management** > **New Catalog**.
3. In the Input New Catalog Name dialog, specify the catalog name, the data source name and the path. The catalog should be located in the directory where you saved the ODF file. For example, if you saved the ODF file in `C:\odf`, the new catalog should be created also in `C:\odf`.
4. In the Catalog Browser, create a new JDBC connection in the Get JDBC Connection Information dialog as follows:
   a. Enter **com.jinfonet.jdbc.obj.ObjectDriver** in the Driver text field.
   b. Type **jdbc:jinfonet:object:@Filename** in the URL text field. @Filename should be the ODF file name. In the example, it is jdbc:jinfonet:object:@orgchart.
   c. Click **OK** to set up the connection.
5. In the Add Table dialog, select the table **OrgChart** and click **Add**.
6. Click **Done** to close the Add Table dialog.

Now, the table OrgChart will be added into the catalog. You can use it to develop reports as required.

Note: When you publish the catalog and its reports to JReport Server, whether the ODF file is located in `<install_root>\lib` or in the directory where the catalog is located, it will be deployed to the path where the catalog is located on the server side.

Example 3: XML file with dynamic parameters

When you design a report from the tables imported through OOJDBC, and the tables have the same input parameters, but you don't want to input the same parameters again, JReport provides a way to allow you to bind parameters in a catalog, so they can share the same parameter value.

This example demonstrates how to import tables with dynamic parameters from the ODF file generated by the Model Wizard.

Importing the XML file using the Model Wizard

To import an XML file with the Model Wizard, take the following steps:

1. Run ModelWizard.bat in `<install_root>\bin`. The Object Source Wizard will then appear.
2. Add two tables as shown in Example 1. The first table named employee1 has one String type column, and the second table named employee2 has one Int type column. They have the same input parameter for param0. The default value is `h:\xml\employee.xml`. Make sure to check the option Is Dynamic.
3. In the Object Source Wizard, click **File** > **Save**. You will then be prompted to save the ODF file. You should save it to `<install_root>\lib` or the directory where the catalog will be. Here we save it as xmlparam.odf.

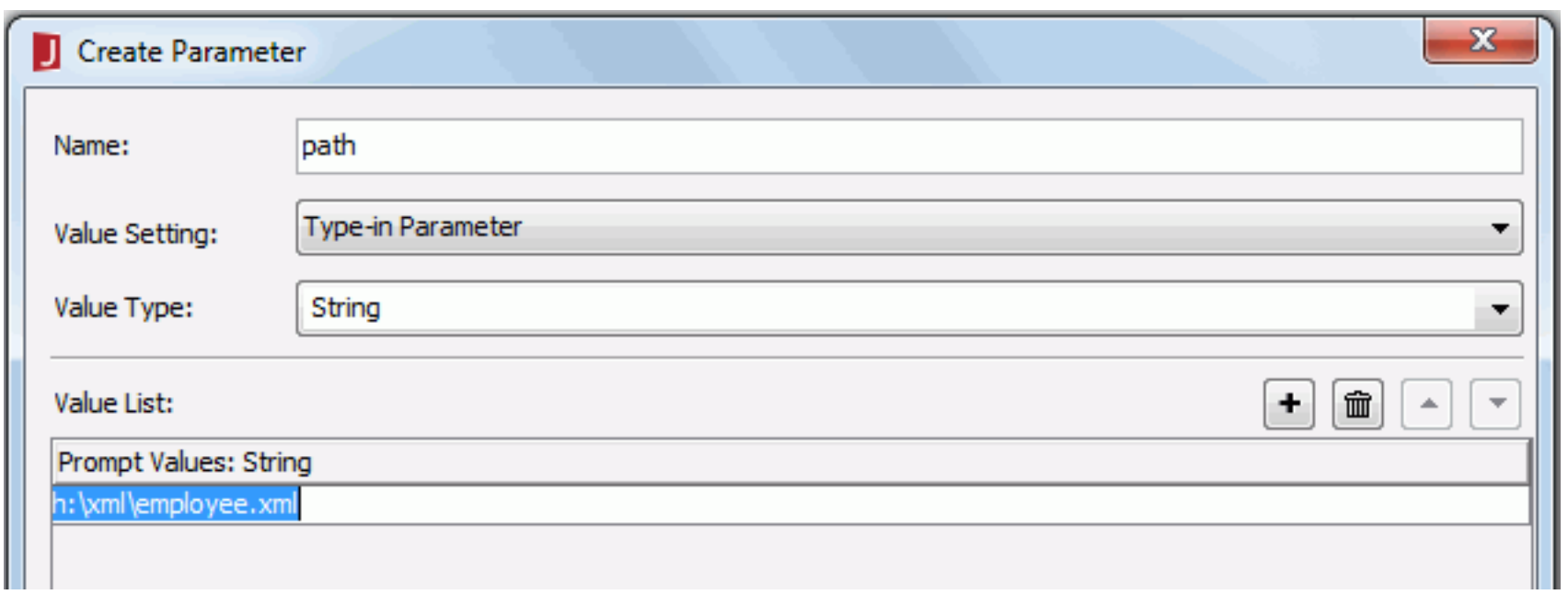
Notes:

- If you choose the option Is Dynamic, this parameter will be passed to JReport Designer. After you create a new connection, you will find this parameter in the node Parameters in the Catalog Browser.
- The same ODF file cannot exist in both `<install_root>\lib` and the directory where the catalog is to be located.

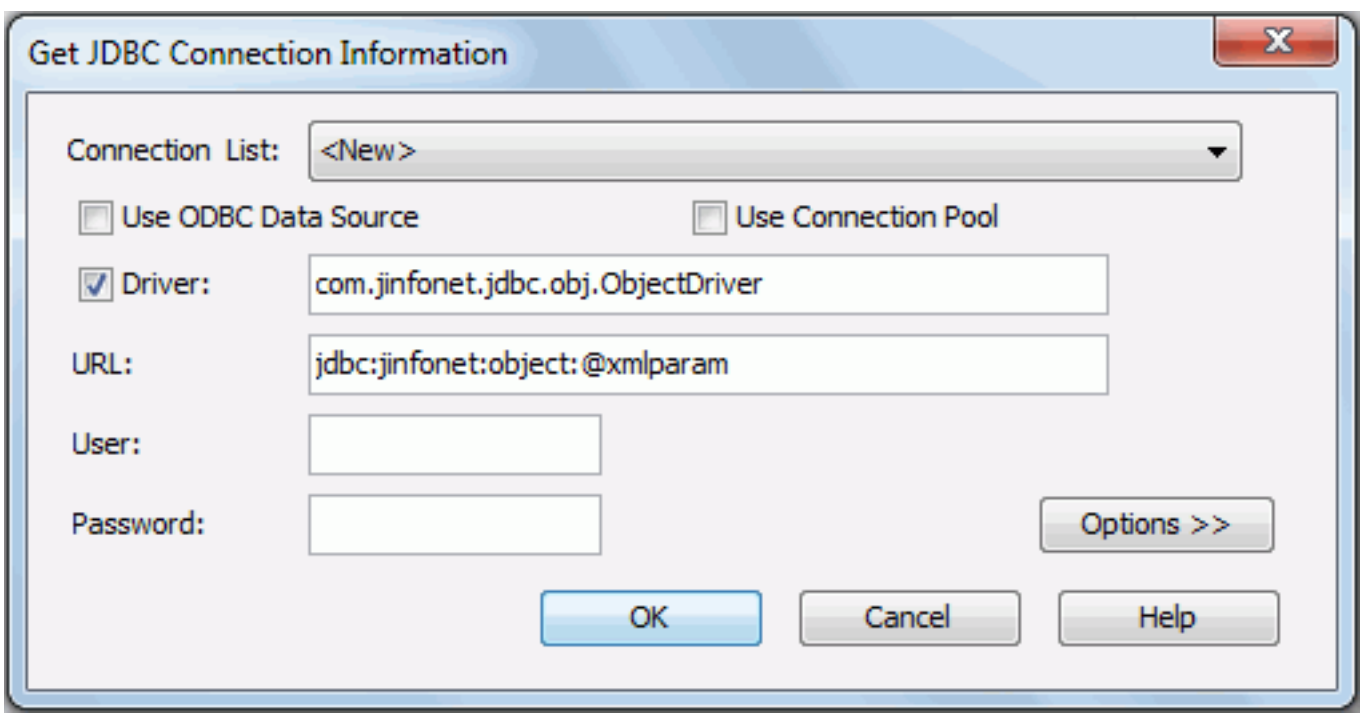
Importing the ODF file into a catalog

To import the generated ODF file into a JReport catalog, take the following steps:

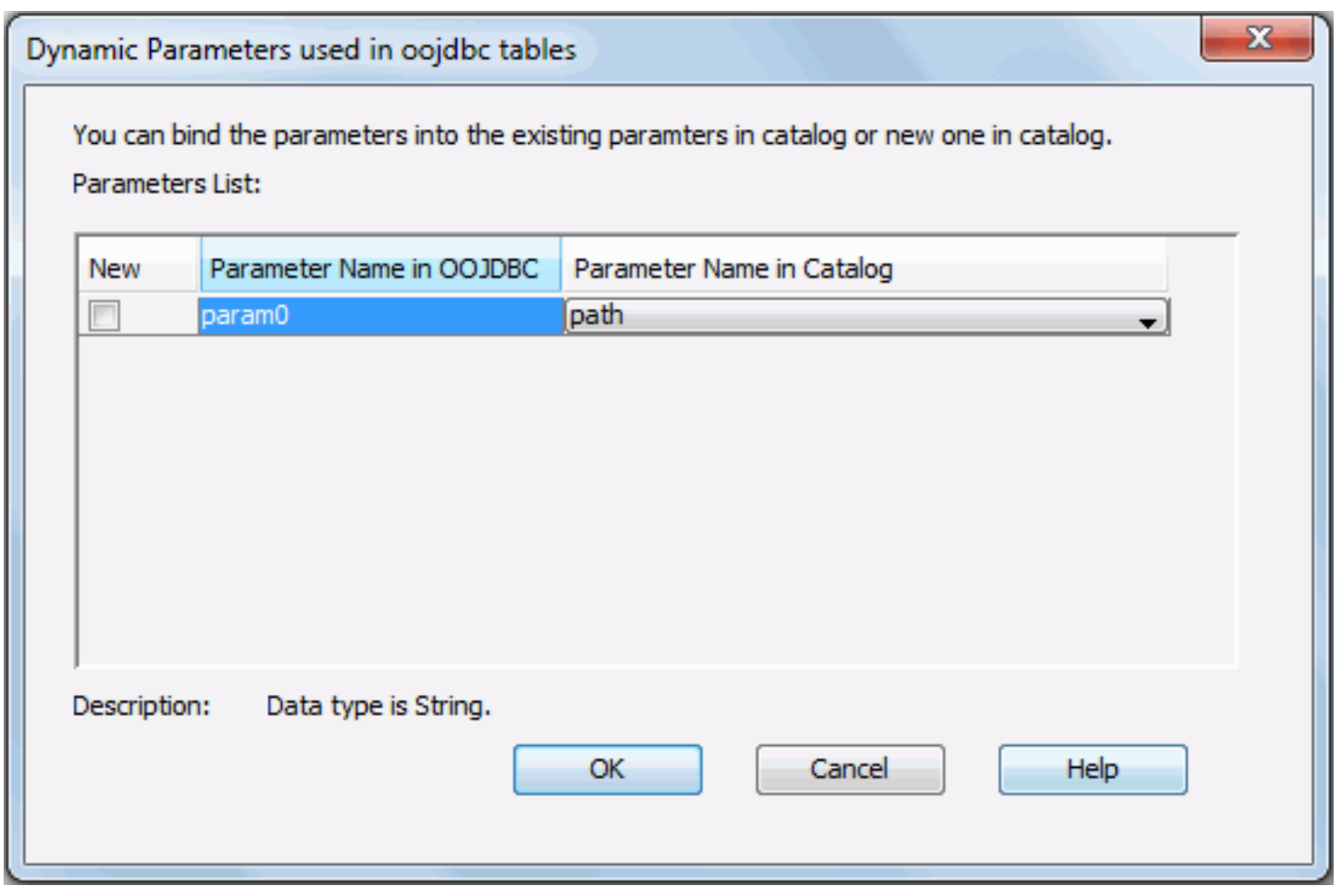
1. Start JReport Designer.
2. Click **File** > **Catalog Management** > **New Catalog**.
3. In the Input New Catalog Name dialog, specify the catalog name, the data source name and the path. The catalog should be located in the directory where you saved the ODF file. For example, if you saved the ODF file in `C:\odf`, the new catalog should be created also in `C:\odf`.
4. In the Catalog Browser, create a parameter that is of String data type and is named path. This parameter will be bound to the parameters in the tables that were imported through OOJDBC.



5. Create a new connection in the Get JDBC Connection Information dialog as follows:



6. The Add Table dialog appears prompting you to add the listed tables. Select the table **employee1** and click the **Add** button. The Dynamic Parameter used in oojdbc tables dialog will then be displayed. Uncheck the checkbox in the New column and select the parameter path from the drop-down list to bind param0 to the parameter path.



7. Repeat the above step to add the employee2 table and bind its parameter param0 to the parameter path. Now the parameters in the two tables will share the same parameter path.
8. Click the **New Page Report** button on the Standard toolbar to develop a report on the newly inserted tables.
9. When viewing the report, the Enter Parameter Values dialog will be displayed, asking you to input the parameter. You will only need to input the path of imported XML file once since the parameter is shared with the parameters in the two tables.

Modifying the dynamic parameters

You can modify the links between the parameters in OOJDBC and parameters in the JReport catalog. To do this:

1. Launch the Catalog Browser in JReport Designer.
2. Select the table with the dynamic parameters that are to be modified. Right-click on the specified table, and then select **Parameters** from the shortcut menu.
3. Highlight the parameter in the catalog, and then select the parameter from the drop-down list to bind it to the parameter in OOJDBC.

Using class file as a data source via OOJDBC

This example shows how to generate the ODF file in a different way using the file demo.zip in `<install_root>\help\samples\OOJDBC`, which contains all the necessary classes and methods needed for making the OOJDBC data source work with Java objects.

Preparations

Before the OOJDBC data source works with Java objects, you should do the preparations, which are detailed in the following two steps:

1. You should define a class that specifies the column name and the method of getting data from this column. In this example, the class Products.java in demo.zip takes this role. This class file contains the method public int getProductID(), which implies that the column name is ProductID and you can get the product ID via this method.
2. You should publish a method to access a group of products. That means OOJDBC needs a result set with multiple rows, and each row is one of the products, so that you should define a class that can retrieve the rows. In this example, the file SQLReader.java in demo.zip takes this role. In this file, there is a static* method JCollection execSQL(String jdbcDriver, String url, String sql), and this method returns a collection that contains the rows of the products. This collection can be one of the following three classes java.util.collection for Java2, java.util.Vector for Java1.1 and com.jinfonet.jdbc.obj.JCollection for both.

Also, the batch files used to start the Model Wizard, JReport Designer and JReport Server should be modified before you can import the product table and use it to generate reports. To do this:

1. Save demo.zip in `<install_root>\help`.
2. Add it to the class path of the batch file setenv.bat in `<install_root>\bin` of both Designer and Server. setenv.bat updates the class path for both ModelWizard and Designer.

Importing the class file using the Model Wizard

To import the class file using the Model Wizard, take the following steps:

1. Start the Model Wizard (ModelWizard.bat).
2. In the Object Source Wizard, click **Add Table** and the Add Table dialog will be displayed.
3. In the ClassName field, type **com.jinfonet.jdbc.demo.Products**, input **demo** in the Table Name field, and then click the **Parse** button.
4. Click **Direct Access Path**. In the Access Method dialog, input **com.jinfonet.jdbc.demo.SQLReader** in the Root Class Name field. Then click the **Parse** button. In the Methods box, select the method **execSQL**, and in the Parameters box, fill in the default values as follows:

   Param0 (the driver of the database): org.hsqldb.jdbcDriver
   Param1 (the connection URL of the database): `jdbc:hsqldb:c:\JReport\Designer\Demo\db\JRDemo`
   Param2 (the user name): sa
   Param3 (the password): (no value)
   Param4 (the SQL statement): select * from products
5. Press the **OK** button and the table demo will now have been added.
6. In the Object Source Wizard, select **File** > **Save**. You will then be prompted to save the ODF file. You should save it to `<install_root>\lib` or the directory where the catalog will be. Here we save it as demo.odf.

Note: The same ODF file cannot exist in both `<install_root>\lib` and the directory where the catalog is located.

Importing the ODF file into a catalog

To import the ODF file to a JReport catalog, take the following steps:

1. Start JReport Designer.
2. Click **File** > **Catalog Management** > **New Catalog**.
3. In the Input New Catalog Name dialog, specify the catalog name, the data source name and the path. The catalog should be located in the directory where you saved the ODF file. For example, if you saved the ODF file in `C:\odf`, the new catalog should be created also in `C:\odf`.
4. In the Catalog Browser, create a new connection with the Get JDBC Connection Information dialog as follows:
   a. Enter **com.jinfonet.jdbc.obj.ObjectDriver** in the Driver text field.
   b. Type **jdbc:jinfonet:object:@Filename** in the URL text field. @Filename should be the .odf name you have saved but do not include the extension. So in the example, it should be jdbc:jinfonet:object:@demo.
   c. Click **OK** to set up the connection.
5. In the Add Table dialog, select the table **demo** and click **Add**.
6. Click **Done** to close the Add Table dialog.

Now, the table demo has been added into the catalog. You can use it to develop reports as required.

Note: When you publish the catalog and its reports to JReport Server, whether the ODF file is located in `<install_root>\lib` or in the directory where the catalog is located, it will be published to the path where the catalog is located on the server side.

Improvement of OOJDBC in this example

JReport provides an interface com.jinfonet.interfaces.query.TableFilter and a simple implementation com.jinfonet.jdbc.obj.TableFilterImpl, which can be used in this example to improve the performance of the select and WHERE clauses in a query. You can extend TableFilterImpl to create your own Reader class. In addition, a demo FilterSQLReader.java in the file demo.zip has been made for your reference.

Note: If you are using FilterSQLReader.java instead of SQLReader.java, you should input com.jinfonet.jdbc.demo.FilterSQLReader in the Root Class Name field of the Access Method dialog.

Using EJB as a data source via OOJDBC

JReport Designer can support any data source through UDS. However, to implement it, users need to be able to code with the Java language. In this case, the usage of UDS is confined to more advanced users. To resolve the problems, JReport has defined OOJDBC to cover an OODS (Object Oriented Data Source) with a JDBC interface which supports EJB as a data source via OOJDBC (currently JReport supports EJB2.x). OOJDBC is similar to a JDBC driver and is based on Java application, EJB (Enterprise Java Bean), LDAP (Light Weight Directory Access Protocol), which can be defined as a database.

The following examples explain how to use EJB as a data source via OOJDBC:

- Example 1: EJB deployed to Java EE
- Example 2: EJB deployed to BEA WebLogic Server 5.1
- Example 3: EJB deployed to BEA WebLogic Server 6.1

Example 1: EJB deployed to Java EE

This example shows how to use an EJB deployed to Java EE as a data source and import it to a JReport catalog via OOJDBC.

Follow the steps below to achieve your goal:

1. Installation
2. Importing the EJB using the Model Wizard
3. Importing the ODF file into a catalog

Installation

You are required to install the following before you can import EJB using the Model Wizard:

1. Install Java EE. You can download it from http://java.sun.com/javaee/downloads/index.jsp. After the installation of Java EE, you will find j2ee.jar in `<java_ee_install_root>\lib`.
2. Install Jaxp1.1. You can download it from http://java.sun.com/xml/download.html. After installation, you will find three jars xalan.jar, crimson.jar and jaxp.jar in `<jaxp1.1_insatll_root>`.
3. Install JReport Designer, then add the four jars into the ADDCLASSPATH variable of the batch file setenv.bat in `<designer_install_root>\bin`. For example:

   ```
   set ADDCLASSPATH=%JAVAHOME%\lib\tools.jar;c:\java\classes;c:\jaxp1.1\xalan.jar;c:\jaxp1.1\crimson.jar;c:\jaxp1.1\jaxp.jar;
   ```

Note: After you have deployed your EJB to the Java EE server, a Jar will be returned in the root directory. In this example, Java(TM) Pet Store Demo of jps1.0 in Java EE has been used. Note that versions later than jps1.0 are not supported. After deploying to the Java EE server, estoreClient.jar is returned to the root directory. Add it to the class path too. For example:

```
set ADDCLASSPATH=%JAVAHOME%\lib\tools.jar;c:\jps1.0\estoreClient.jar;c:\jaxp1.1\xalan.jar;c:\jaxp1.1\crimson.jar;c:\jaxp1.1\jaxp.jar;
```

Importing the EJB using the Model Wizard

Make an EJB initializing class that is coded by Jinfonet, and is for looking up EJBHome by the JNDI name.

This example uses Java(TM) Pet Store Demo. JNDI name is estore/catalog. The class name is com.sun.estore.catalog.model.Category. This table is defined by users. Users can get the table they request through parsing the class.

1. Run the modified batch file ModerWizard.bat in `<designer_install_root>\bin` and the Object Source Wizard will then be displayed. Click **Add Table**.
2. In the Add Table dialog, enter **com.sun.estore.catalog.model.Category** in the ClassName field and define the table name as **Category**. Then click the **Parse** button.
3. Click the **Direct Access Path** button to define the accessing path in the Access Method dialog.
   a. Fill in **com.jinfonet.jdbc.ejb.EJBInitializer** in the Root Class Name field. Then click the **Parse** button.
   b. In the Parameter box, find the DefaultValue column and fill in **estore/catalog** by which the EJB in the Java EE server will be accessed. Check the **CastTo** option, and enter **com.sun.estore.catalog.ejb.CatalogHome**.

      Note: After you enter the DefaultValue, you should press Enter on your keyboard, or an error will occur. When you set cast to, you should also start the Java EE server by running j2ee.bat in `<java_ee_install_root>`. And you should also choose ... first.
4. If you want to use a dynamic parameter, check the option **Is Dynamic**. The syntax of the method is: an array of Boolean values to indicate which parameter is dynamic at runtime. If the parameter is dynamic the value should be prompted and initialized before the SQL is executed.
5. Click **Next**. In the next dialog, choose **create** as the method name.
6. Click **Next**, and now you can see the return type of the method name getCategories defined previously.
7. Click **OK** and the table Category will now have been added.

- In the Object Source Wizard, select **File** > **Save**. You will then be prompted to save the ODF file. u should save it to `<designer_install_root>\lib` or the directory where the catalog will be. Here we e it as ejb.odf.

Notes:

- If specifying to apply dynamic parameters, that is, you check the option Is Dynamic, the parameter will be passed to JReport Designer. After you create a connection in the Catalog Browser, you will find the parameter in the Parameters node.
- The same ODF file cannot exist in both `<designer_install_root>\lib` and the directory where the catalog is located.

Importing the ODF file into a catalog

To import the ODF file that is generated when you imported the EJB using the Model Wizard, take the following steps:

1. Start JReport Designer with the modified batch file.

   Note: Before you start creating reports, you should first start your database. In this example, start cloudscape in `<j2sdkee1.2_install_root>`.
2. Click **File** > **Catalog Management** > **New Catalog**.
3. In the Input New Catalog Name dialog, specify the catalog name, the data source name and the path. The catalog should be located in the directory where you saved the ODF file. For example, if you saved the ODF file in `C:\odf`, the new catalog should be created also in `C:\odf`.
4. In the Catalog Browser, create a new connection with the Get JDBC Connection Information dialog as follows:
   a. Enter **com.jinfonet.jdbc.obj.ObjectDriver** in the Driver text field.
   b. Type **jdbc:jinfonet:object:@Filename** in the URL text field. @Filename should be the .odf name you have saved. So in the example, it should be jdbc:jinfonet:object:@ejb.
   c. Click **OK**.
5. In the Add Tables dialog, select the two tables in the Tables box, and click the **Add** button.

Now, the tables will have been added into the catalog. You can use it to develop reports as required.

Example 2: EJB deployed to BEA WebLogic Server 5.1

In this example, you will be shown how to use an EJB that is deployed to a BEA WebLogic Server 5.1 as a data source and import it to a JReport catalog via OOJDBC.

Follow the steps below to achieve your goal:

1. Installation
2. Deploying the EJB to WebLogic
3. Importing the EJB using the Model Wizard
4. Importing the ODF file into a catalog

Installation

You are required to install the following before you can begin this example:

- Install JReport Designer. In this example, we installed JReport Designer to `C:\JReport\Designer`.
- Install BEA WebLogic Server. To download BEA WebLogic server, go to the BEA WebLogic website: http://www.weblogic.com/licbeta.html, and install and configure your server. Here, we install the BEA WebLogic Server to `D:\Weblogic`.

  After installation, you will find weblogicaux.jar in `<weblogic_install_root>\lib`. Add weblogicaux.jar into the class path of the batch files of both JReport Designer and Model Wizard.

  In this example, we use the property value `Context.INITIAL_CONTEXT_FACTORY = weblogic.jndi.WLInitialContextFactory` to create the connection, and the class weblogic.jndi.WLInitialContextFactory is located in `D:\Weblogic\classes`. So you should also add `D:\Weblogic\classes` into the class path of the batch files that start both JReport Designer and Model Wizard.
- Install Jaxp1.1. You can download it from http://java.sun.com/xml/download.html. After installation, there will be three jars xalan.jar, crimson.jar and jaxp.jar in the installroot of Jaxp1.1. Add these three jars into the class path of the batch files that start both JReport Designer and Model Wizard.
- Add some system property values into the batch files. In this example, we set `-Djava.naming.factory.initial="weblogic.jndi.WLInitialContextFactory" -Djava.naming.provider.url="t3://localhost:7001"` to connect to BEA WebLogic Server.

After performing these steps, the batch file that starts the Model Wizard should look something like this:

```
C:\jdk1.6.0_17\jre\bin\java.exe" "-Dinstall.root=C:\jreport\designer"
-classpath "C:\jreport\designer\lib\report.jar;C:\jreport\designer\lib\resource_en_US.jar;
C:\jreport\designer\lib\JREngine.jar;C:\jreport\designer\lib\mail.jar;
C:\jreport\designer\lib\activation.jar;C:\jreport\designer\lib\unicode.zip;
C:\TEMP";"C:\jreport\designer";D:\Weblogic510\lib\weblogicaux.jar;D:\Weblogic510\classes;
w:\sf\lib\xalan.jar;w:\sf\lib\crimson.jar;w:\sf\lib\jaxp.jar;
D:\weblogic510\myserver\ejb_basic_beanManaged.jar
-Dreporthome="C:\jreport\designer"
-Djava.naming.factory.initial= "weblogic.jndi.WLInitialContextFactory"
-Djava.naming.provider.url="t3://localhost:7001"
com.jinfonet.jdbc.model.wizard. ModelWizard %1 %2 %3 %4 %5 %6 %7 %8 %9
```

Deploying the EJB to WebLogic

To deploy the EJB to a BEA WebLogic Server, take the following steps:

1. Make the EJB include methods to return Collection or JCollection.

   After you have installed BEA WebLogic server, you will find several EJBs. In this example, the entity bean examples.ejb.basic.beanManaged.AccountBean will be used. The JNDI name is beanManaged.AccountHome. The remote interface is examples.ejb.basic.beanManaged.Account.class. The home interface is examples.ejb.basic.beanManaged.AccountHome.class. Before deploying it, you will have to provide a method which returns java.util.Collection or com.jinfonet.jdbc.obj.JCollection. JCollection is an interface coded by Jinfonet.

   In the Model Wizard, you will need to inform the OOJDBC driver of the method with which to use to fetch data. The OOJDBC driver can only accept a method which returns java.util.Collection or com.jinfonet.jdbc.obj.JCollection.

   In this example, we change AccountBean.java to include the following method:

   ```
   /**
   * Finds all EJBeans.
   * @return Enumeration EJBean Primary Keys
   * @exception javax.ejb.EJBException
   * if there is a communications or systems failure
   */
   public Collection ejbFindAllAccounts()
   {
       log("getAllAccounts ()");
       HashSet col = new HashSet();
       Connection con = null;
       PreparedStatement ps = null;
       try
           {
           con = getConnection();
           ps = con.prepareStatement("select id from ejbAccounts");
           ps.executeQuery();
           ResultSet rs = ps.getResultSet();
           String pk;
           while (rs.next())
               {
               pk = rs.getString(1);
               col.add(pk);
               }
           return (Collection)col;
       }
       catch (SQLException sqe)
       {
           log("SQLException: " + sqe);
           throw new EJBException (sqe);
       }
       finally {
       cleanup(con, ps);
   }
   }
   ```

   And we add a method in AccountHome.java:

   ```
   /**
   * Finds all EJBeans.
   * Returns a Collection of found EJBean primary keys.
   *
   * @return Collection of Account
   * @exception javax.ejb.FinderException
   if there is an error while accessing the persistent storage
   * @exception java.rmi.RemoteException if there is a communications
   * or systems failure
   * @see examples.ejb.basic.beanManaged.AccountBean
   */
   public Collection findAllAccounts()
       throws FinderException, RemoteException;
   ```

2. Create the ejb-jar file.

   After calling javac.exe of JDK 6 to compile EJB classes, create a standard EJB jar file, including XML deployment descriptors. Then use the tool WebLogic EJB Deployer to create a deployable Jar file.

3. Deploy the EJB.

   In Step 2, the Jar ejb_basic_beanManaged.jar has been created. Now put the Jar into the class path of the WebLogic Server, edit the weblogic.properties file to include the new EJB. After stopping and restarting the WebLogic Server, you can access the EJB in a client. To do this, first create 20 accounts, and then add the Jar ejb_basic_beanManaged.jar into the class path both in the batch file of JReport Designer and that of Model Wizard.

Importing the EJB using the Model Wizard

Make an EJB initializing class that is coded by Jinfonet and is for looking up the EJBHome by the JNDI name.

The JNDI name of the entity bean examples.ejb.basic.beanManaged.AccountBean is beanManaged.AccountHome. The remote interface is examples.ejb.basic.beanManaged.Account.class. The home interface is examples.ejb.basic.beanManaged.AccountHome.class.

To import the EJB using the Model Wizard, take the following steps:

1. Run the modified batch file ModelWizard.bat in `<designer_install_root>`. The Object Source Wizard will then appear.
2. Click **Add Table** and define a table named **Account**, and then click the **Parse** button. The Add Table dialog will then appear.
3. Click **Direct Access Path** in the Add Table dialog to define the accessing path of the table in the Access Method dialog.
   a. In the Root Class Name text box, fill in **com.jinfonet.jdbc.ejb.EJBInitializer**, and then click the **Parse** button. For DefaultValue, enter JNDI name **beanManaged.AccountHome**, and select **CastTo** as **examples.ejb.basic.beanManaged.AccountHome**.

      Note: After you fill in DefaultValue, you should press Enter in your keyboard, or an error will occur. Also, before you set CastTo, you should start the BEA WebLogic server. Choose ... first, followed with the down arrow. These actions will connect to the server.
4. Check the option **Is Dynamic** if you want to use dynamic parameters. The syntax of the method is: an array of Boolean values to indicate which parameter is dynamic at runtime. If the parameter is dynamic, the value should be prompted and initialized before the SQL is executed.
5. Click **Next**, and then select the method **findAllAccounts** as the method name.
6. Click **OK**, and the table Account will now have been added.

- In the Object Source Wizard, select **File** > **Save**. You will then be prompted to save the ODF file. You ould save it to `<designer_install_root>\lib` or the directory where the catalog will be. Here we save it as ologic.odf.

Notes:

- If you use dynamic parameters, that is, you check the option Is Dynamic , the parameter will be passed to JReport Designer. After you create a connection in the Catalog Browser, you will find the parameter in the Parameters node.
- The same ODF file cannot exist in both `<designer_install_root>\lib` and the directory where the catalog is located.

Importing the ODF file into a catalog

Here you will be shown how to load the table Account into a catalog file, and create reports with it.

Before starting JReport Designer, make sure that the class path includes the relative jar files and path, and has set relative system property values. In this example, the batch file for starting JReport Designer is as follows:

```
C:\jdk1.6.0_17\jre\bin\java.exe;-Dinstall.root=C:\JReport
-classpath :\JReport\Designer\lib;C:\jreport\designer\lib\report.jar;
C:\JReport\Designer\lib\resource_en_US.jar;C:\JReport\Designer\lib\JREngine.jar;
C:\JReport\Designer\lib\mail.jar;C:\JReport\Designer\lib\activation.jar;
C:\JReport\Designer\lib\unicode.zip;C:\TEMP";"C:\JReport\Designer";
D:\Weblogic\lib\weblogicaux.jar;D:\Weblogic\classes;
W:\sf\lib\xalan.jar;W:\sf\lib\crimson.jar;W:\sf\lib\jaxp.jar;
D:\weblogic\myserver\ejb_basic_beanManaged.jar
-Dreporthome=C:\JReport -Djava.naming.factory.initial=weblogic.jndi.WLInitialContextFactory
-Djava.naming.provider.url=t3://localhost:7001
jet.report.ide.JReport %1 %2 %3 %4 %5 %6 %7 %8 %9
```

To import the ODF file into a JReport catalog, take the following steps:

1. Start JReport Designer with the modified batch file.
2. Click **File** > **Catalog Management** > **New Catalog**.
3. In the Input New Catalog Name dialog, specify the catalog name, the data source name and the path. The catalog should be located in the directory where you saved the ODF file. For example, if you saved the ODF file in `C:\odf`, the new catalog should be created also in `C:\odf`.
4. In the Catalog Browser, create a new connection with the Get JDBC Connection Information dialog as follows:
   a. Enter **com.jinfonet.jdbc.obj.ObjectDriver** in the Driver text field.
   b. Enter **jdbc:jinfonet:object:@File name** in the URL text field. @File name should be the .odf name you saved. In the example, it is jdbc:jinfonet:object:@weblogic.

Now, the table will have been added into the catalog. You can use it to develop reports as required.

Example 3: EJB deployed to BEA WebLogic Server 6.1

This example shows how to use an EJB that is deployed to BEA WebLogic Server 6.1 as a data source and import it to a JReport catalog via ODJDBC.

Follow the steps below to achieve your goal:

1. Installation
2. Modifying and rebuilding the EJB
3. Setting up the demo data source
4. Importing the EJB using the Model Wizard
5. Importing the ODF file into a catalog

Installation

You are required to install the following before you can start developing reports using EJB 2.0:

- Install WebLogic Server 6.1 in D:\bea\wlserver6.1_beta.
- Install JReport Designer in D:\JReport\Designer.
- Install Jaxp 1.1 in D:\jaxp1.1.
- Download EJB 2.0 class package from http://java.sun.com/products/ejb/docs.html.

  The downloaded class package is called ejb-2_0-pfd2-class.zip. Save it to your disk and unzip the ejb.jar from this package to your D:\downloads\jdk1.6.0_17 directory.
- Download EJB Upgrade 2.0 from http://commerce.bea.com/downloads/weblogic_server.jsp.

  The package name is ejb20.zip. Unzip ejb20.zip and extract the ejb20.jar to D:\bea\wlserver6.1_beta\lib, and modify the file startWeblogic.cmd in D:\bea\wlserver6.1_beta\config\mydomain to include the ejb20.jar in the class path as the following:

```
set CLASSPATH=.;D:\bea\wlserver6.1_beta\lib\ejb20.jar;....
```

Modifying and rebuilding the EJB

The demo EJB Account.java, AccountBean.java and AccountHome.java will be used in this example. They are located in D:\bea\wlserver6.1_beta\samples\ejb20\basic\beanManaged.

Take the following steps to modify an rebuild the EJB:

1. Modify the source code of the EJB.

   a. Modify AccountBean.java by adding the following two functions:

```
/**This function is used to display the primary key in your report
                 ( a limitation to the java.lang.Object return type). */
public String getPrimaryKeyString()
{
    // return (String) ctx.getPrimaryKey();
    return accountId;
}
/**
* Finds all EJBeans.
* @return Enumeration EJBean Primary Keys
* @exception javax.ejb.EJBException
* if there is a communications or systems failure
*/
public Collection ejbFindAllAccounts()
{
    log("getAllAccounts ()");
    HashSet col = new HashSet();
    Connection con = null;
    PreparedStatement ps = null;
    try {
        {
        con = getConnection();
        ps = con.prepareStatement("select id from ejbAccounts");
        ps.executeQuery();
        ResultSet rs = ps.getResultSet();
        String pk;
        while (rs.next())
            {
            pk = rs.getString(1);
            col.add(pk);
        }
        return (Collection)col;
    } catch (SQLException sqe)
        {
        log("SQLException: " + sqe);
        throw new EJBException(sqe);
    } finally
        {
        cleanup(con, ps);
        }
    }
}
```

Note: Remember to add import java.util.HashSet; into the AccountBean.java file.

   b. Modify AccountHome.java by adding the following function:

```
/**
* Finds all EJBeans.
* Returns an Collection of found EJBean primary keys.
*
* @return Collection of Account
* @exception javax.ejb.FinderException
if there is an error while accessing the persistent storage
* @exception java.rmi.RemoteException if there is a communications
* or systems failure
* @see examples.ejb.basic.beanManaged.AccountBean
*/
public Collection findAllAccounts()
    throws FinderException, RemoteException;
```

   c. Modify Account.java by adding the following function:

```
/*used to display the primary key in reports*/
public String getPrimaryKeyString() throws RemoteException;
```

   d. Modify Client.java which is used to create EJB objects. The comments included below are the statements used to remove the EJB objects:

```
log("Removing beans...");
for (int i=0; i<numBeans; i++)
{
    accounts[i].remove();
}
```

2. Modify the file build.cmd in D:\bea\wlserver6.1_beta\samples\examples\ejb20\basic\beanManaged as follows:

```
*** Begin of build.cmd ***

REM Copyright (c) 2000-2001 BEA Systems, Inc. All Rights Reserved.
REM You must set your environment by running the setExamplesEnv shell script
REM prior to executing this script.
SET CLASSPATH=D:\jdk1.6.0_17\lib\rt.jar;D:\downloads\jdk1.6.0_17\lib\tools.jar;
D:\bea\wlserver6.1_beta\lib\weblogic.jar
SET WL_HOME=D:\bea\wlserver6.1_beta
SET APPLICATIONS=D:\bea\wlserver6.1_beta\config\mydomain\applications
SET EX_WEBAPP_CLASSES=D:\bea\wlserver6.1_beta\config\examples\
applications\examples\webapp\
SET CLIENT_CLASSES=D:\bea\wlserver6.1_beta\config\examples\clientclasses
REM Create the build directory, and copy the deployment descriptors into it
mkdir build build\META-INF
copy *.xml build\META-INF
REM Compile EJB classes into the build directory (jar preparation)
D:\jdk1.6.0_17\bin\javac -classpath %CLASSPATH%
-d build Account.java AccountHome.java ProcessingErrorException.java
AccountBean.java
REM Make a EJB jar file, including XML deployment descriptors
cd build
D:\bea\jdk130\bin\jar cv0f std_ejb20_basic_beanManaged.jar META-INF examples

REM Run EJBC on jar file
D:\jdk1.6.0_17\bin\java -classpath %WL_HOME%\lib\weblogic_sp.jar;
%WL_HOME%\lib\weblogic.jar;%CLASSPATH%
D:\bea\wlserver6.1_beta\lib\weblogic.jar build\std_ejb20_basic_beanManaged.jar
REM Compile servlets and EJB interfaces into
D:\JWC\classes directory of the Examples Web App
REM This ensures that the EJB can be accessed by JSPs
REM servlets of the Examples Web App
D:\jdk1.6.0_17\bin\javac -classpath %CLASSPATH%
%EX_WEBAPP_CLASSES% Servlet.java Account.java AccountHome.java
ProcessingErrorException.java
REM Compile EJB interfaces & client app into the clientclasses directory
D:\jdk1.6.0_17\bin\javac -classpath %CLASSPATH% -d %CLIENT_CLASSES%
Account.java AccountHome.java ProcessingErrorException.java Client.java

** End of file build.cmd ***
```

- Execute build.cmd to build up ejb20_basic_beanManaged.jar.

Setting up the demo data source

Set up the demo data source following the steps below:

1. Copy cloudscape.jar in D:\bea\wlserver6.1_beta\samples\eval\cloudscape\lib\ to D:\bea\wlserver6.1_beta\lib
2. Modify startWeblogic.cmd in D:\bea\wlserver6.1_beta\config\mydomain to include cloudscape.jar in the class path and set the home of cloudscape as D:\bea\wlserver6.1_beta\samples\eval\cloudscape\data (location of the database):

```
set CLASSPATH=.;..;D:\bea\wlserver6.1_beta\lib\cloudscape.jar;

echo on

%JAVA_HOMEN\bin\java" -hotspot -ms64m -mx64m -classpath
%CLASSPATH% -Dweblogic.Domain=mydomain
-Dweblogic.Name=myserver "-Dbea.home=d:\bea"
-Dcloudscape.system.home=D:\bea\wlserver6.1_beta\samples\eval\cloudscape\data
"-Djava.security.policy=D:\bea\wlserver6.1_beta\lib\weblogic.policy"
-Dweblogic.management.password=WebLs_PWk -Dreportfonem0:\jns520 weblogic.Server
```

3. Start the WebLogic Server from the Start menu.
4. Use the web browser to connect to the console through http://localhost:7001/console.
5. Expand **Service** > **JDBC** and select the **Connection Pool** node. From the right panel, click on the link **Create a new JDBC Connection Pool**.
6. Define the Connection Pool as follows:

   Name: demoPool
   URL: jdbc:cloudscape:demo
   Driver Classname: COM.cloudscape.core.JDBCDriver
   Properties: autocommit=true
7. Click on the **Create** button to create the connection pool.
8. Click the **Target** tab, and add myserver to the Chosen list. Then click on the **Apply** button.
9. In the Monitoring tab, click the link **Monitor all Active Pools** to check if the pool is active.
10. Click **Data Source** from the left pane in JDBC node.
11. Click **Configure a new JDBC Data Source** to create a new data source.
12. In the Configuration tab, input the following information:

    Name: demoPool
    JNDI Name: examples-dataSource-demoPool
    Pool Name: demoPool
13. Click the **Create** button to create this data source.
14. Click the **Targets** tab to add myserver to the right text field, and then click the **Apply** button to apply these changes. Now your demo data source for the beans is set up.
15. Run the class Client.class to create EJB objects. The class has been compiled when you run build.cmd, and will be outputted to D:\bea\wlserver6.1_beta\config\examples\clientclasses directory. The command should be as follows:

```
D:\jdk1.6.0_17\bin\java -classpath D:\jdk1.6.0_17\jre\lib\rt.jar;
D:\downloads\jdk\EJB20_BIG\EJB.JAR;D:\bea\wlserver6.1_beta\lib\weblogic.jar;
D:\bea\wlserver6.1_beta\config\examples\clientclasses
examples.ejb20.basic.beanManaged.Client t3://localhost:7001
```

Importing the EJB using the Model Wizard

The JNDI name of the entity bean examples.ejb20.basic.beanManaged.AccountBean is ejb20-beanManaged-AccountHome. The remote interface is examples.ejb20.basic.beanManaged.Account.class. The home interface is examples.ejb20.basic.beanManaged.AccountHome.class.

To import the EJB using the Model Wizard, take the following steps:

1. Modify ModelWizard.bat in D:\jreport\designer\bin as follows:

```
"D:\jdk1.6.0_17\jre\bin\java.exe" "-Dinstall.root=D:\jreport\designer"
-classpath D:\jreport\designer\lib\report.jar;D:\jreport\designer\lib\resource_en_US.jar;
D:\jreport\designer\lib\JREngine.jar;D:\jreport\designer\lib\mail.jar;
D:\jreport\designer\lib\activation.jar;D:\jreport\designer\lib\unicode.zip;
C:\TEMP";"D:\jreport\designer";D:\bea\wlserver6.1_beta\lib\weblogic.jar;
D:\bea\wlserver6.1_beta\config\mydomain\applications\ejb20_basic_beanManaged.jar;
D:\jaxpi.1\jaxp.jar;Dreportfonem"D:\jaxp1.1\crimson.jar;
-Djava.naming.factory.initial="weblogic.jndi.WLInitialContextFactory"
-Djava.naming.provider.url="t3://localhost:7001"
com.jinfonet.jdbc.model.wizard.ModelWizard %1 %2 %3 %4 %5 %6 %7 %8 %9
```

2. Start the Model Wizard using the modified batch file. The Object Source Wizard will then appear.
3. Click the **Add Table** button and define a table named Account, and then click the **Parse** button. The Add Table dialog will then appear. In this method, only the balance value is stored while creating EJB objects in Client.java, so you can remove the columns deposit, withdraw, and isidentical, leaving only the columns balance and PrimaryKeyString.
4. Click **Direct Access Path** in the Add Table dialog to define the accessing path of the table in the Access Method dialog.

   a. In the Root Class Name field, enter **com.jinfonet.jdbc.ejb.EJBInitializer**, and then click the **Parse** button. In DefaultValue, enter JNDI name **ejb20-beanManaged-AccountHome**, check **CastTo** and select **examples.ejb20.basic.beanManaged.AccountHome**.

      Note: After you fill in DefaultValue, you should click Enter on your keyboard, or the error will occur. Also, before you set CastTo, you should start the BEA WebLogic Server. You should also choose Test first. These actions will connect you to the server.
5. Check the option **IsDynamic** if you want to use dynamic parameters. The syntax of the method is: An array of Boolean values to indicate which parameter is dynamic at runtime. If the parameter is dynamic, the value should be prompted and initialized before executing SQL.
6. Click **Next**. Then select the function **findAllAccounts** as the method name.
7. Click **OK** and the table Account will now have been added.
8. In Object Source Wizard, click **File** > **Save**. You will then be prompted to save the ODF file. You should save it to <designer_install_root>\lib or the directory where the catalog will be. Here we save it as weblogic.odf.

Notes:

- If you use dynamic parameters, that is, you check the option Is Dynamic, the parameter will be passed to JReport Designer. After you create a connection in the Catalog Browser, you will find the parameter in the Parameters node.
- The name ODF file cannot exist in both <designer_install_root>\lib and the directory where the catalog is located.

Importing the ODF file into a catalog

Here, you will be shown how to load the table Account into a catalog file, and create reports with it. Before you JReport Designer, make sure that the class path includes the relative jar files and path, and has set the relative system property values.

In this example, the batch file of JReport Designer is as follows:

```
"D:\jdk1.6.0_17\jre\bin\java.exe" "-Dinstall.root=D:\jreport\designer"
-classpath "g:\temp\jdbc\d.zip;D:\jreport\designer\lib\report.jar;
D:\jreport\designer\lib\resource_en_US.jar;D:\jreport\designer\lib\JREngine.jar;
D:\jreport\designer\lib\mail.jar;D:\jreport\designer\lib\activation.jar;
D:\jreport\designer\lib\unicode.zip;C:\TEMP";"D:\jreport\designer";
D:\bea\wlserver6.1_beta\lib\weblogic.jar;
D:\bea\wlserver6.1_beta\config\mydomain\applications\ejb20_basic_beanManaged.jar;
D:\jaxpi.1\jaxp.jar;D:\jaxp1.1\xalan.jar;D:\jaxp1.1\crimson.jar;
D:\bea\wlserver6.1_beta\lib\ejb20.jar;D:\bea\wlserver6.1_beta\lib\wlepool.jar
-Dreportfonem"D:\jreport\designer"
-Djava.naming.factory.initial="weblogic.jndi.WLInitialContextFactory"
-Djava.naming.provider.url="t3://localhost:7001"
jet.report.ide.JReport %1 %2 %3 %4 %5 %6 %7 %8 %9
```

To import the ODF file into a JReport catalog:

1. Start JReport Designer with the modified batch file.
2. Click **File** > **Catalog Management** > **New Catalog**.
3. In the Input New Catalog Name dialog, specify the catalog name, the data source name and the path. The catalog should be located in the directory where you saved the ODF file. For example, if you saved the ODF file in C:\odf, the new catalog should be created also in C:\odf.
4. In the Catalog Browser, create a new connection with the Get JDBC Connection Information dialog as follows:

   a. Enter **com.jinfonet.jdbc.obj.ObjectDriver** in the Driver text field.
   b. Enter **jdbc:jinfonet:object:@File name** in the URL text field. @File name should be the .odf name you saved. In the example, it is jdbc:jinfonet:object:@weblogic61.
   c. Click **OK**. The Add Table dialog appears. Select a table to be added, then click **Done**.

Now, the table will have been added into the catalog. You can use it to develop reports as required.

Using CSV file as a data source via OOJDBC

JReport Designer supports CSV as a data source which can be accessed via OOJDBC.

Here is an example demonstrating how to import a CSV file to an ODF file generated by the Model Wizard. In this example, employee.csv in `<install_root>\help\samples\OOJDBC` will be used. Store it to `<install_root>\Demo\Reports\TutorialReports` first. Then follow the steps below to import the file to a catalog.

Importing the CSV file directly using the Model Wizard

To import the CSV file directly using the Model Wizard, take the following steps:

1. Run ModelWizard.bat in `<install_root>\bin`. The Object Source Wizard will then appear.
2. Select **File** > **Import Text Files** to bring up the Open Text Files dialog.
3. Browse to the file employee.csv and then check the **First Line Is Column Name** checkbox.
4. Click **Open** and the table Employee will now have been added to the Object Source Wizard.
5. In the Object Source Wizard, click **File** > **Save**. You will then be prompted to save the ODF file. You should save it to `<install_root>\lib` or to the directory where the catalog will be located. Here, we save it as employee_csv.odf.

   Note: The same ODF file cannot exist in both `<install_root>\lib` and the directory where the catalog will be located.

Importing the ODF file into a catalog

To import the generated employee_csv.odf file into a JReport catalog, take the following steps:

1. Start JReport Designer.
2. Click **File** > **Catalog Management** > **New Catalog**.
3. In the Input New Catalog Name dialog, specify the catalog name, the data source name and the path. The catalog should be located in the directory where you saved the ODF file. For example, if you saved the ODF file in `C:\odf`, the new catalog should be created also in `C:\odf`.
4. In the Catalog Browser, create a new JDBC connection in the Get JDBC Connection Information dialog as follows:
   a. Enter **com.jinfonet.jdbc.obj.ObjectDriver** in the Driver text field.
   b. Type **jdbc:jinfonet:object:@Filename** in the URL text field. @Filename should be the ODF file name. In the example, it is jdbc:jinfonet:object:@employee_csv.
   c. Click **OK** to set up the connection.
5. In the Add Table dialog, select the table Employee and click **Add**.
6. Click **Done** to close the Add Table dialog.

Now, the table Employee will be added into the catalog. You can use it to develop reports as required.

Note: When you publish the catalog and its reports to JReport Server, whether the ODF file is located in `<install_root>\lib` or in the directory where the catalog is located, it will be published to the path where the catalog is located on the server side.

DBMS objects in a JDBC connection

Tables/Views/Synonyms, stored procedures and SQL files are all from the raw database. If you want to use them in JReport Designer, you first have to add them to a catalog via the JDBC connection you set up which connects JReport Designer to the database driver.

This section details how to work with these objects separately:

- Tables/Views/Synonyms
- Stored procedures
- Predefined SQL files

Tables/Views/Synonyms

Tables, views and synonyms are mapped objects of the tables, views and synonyms in the raw database that a connection refers to. Mapped objects have mapped names that can be different from their names in the raw database. A catalog stores information about the real tables/views/synonyms mapped, such as the name, qualifier, and owner.

Since tables/views/synonyms are only mapped objects, they can be mapped and then re-mapped. That is, the information of the real tables/views/synonyms stored in a catalog is changeable. This is useful when you are off-line from your database. You will still be able to see the table/view/synonym structure and generate reports. In addition, when changing the connection, for example, from an Access database to an Interbase, you will only need to re-map all the tables/views/synonyms in the catalog as opposed to creating a new catalog.

Note: Synonyms are supported for Oracle database only.

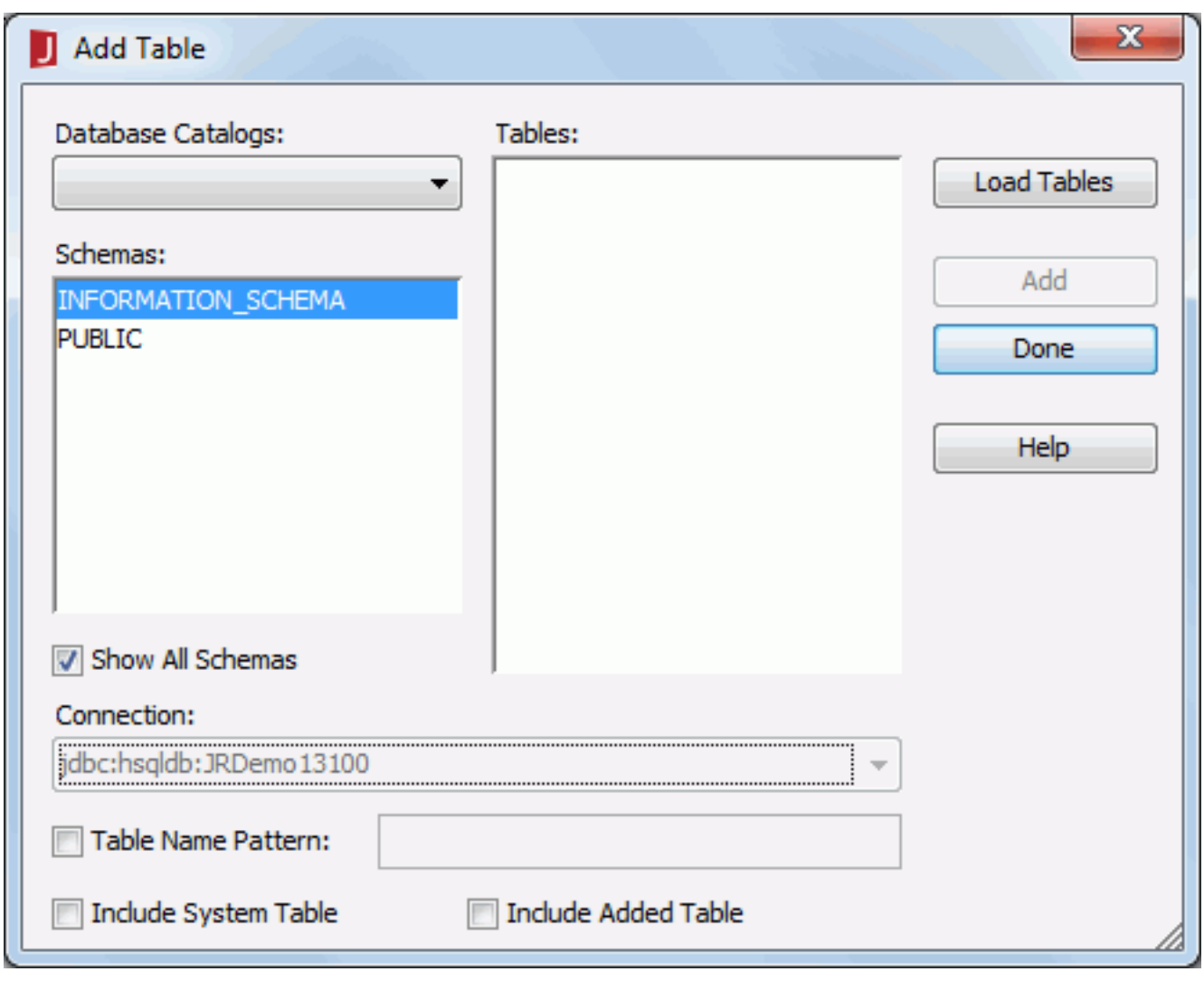
Adding tables/views/synonyms to a catalog

The way to add tables, views or synonyms stored in the database to a catalog via the JDBC connection you set up is the same. Below is the procedure for adding tables, and you can take the same steps to add views or synonyms to a catalog.

To add tables to a catalog:

1. Do either of the following:
   - Right-click the JDBC connection and select **Add Tables** from the shortcut menu.
   - Right-click the **Tables** node of the JDBC connection and select **Add Table** from the shortcut menu.
   - Right-click any folder in the Tables node of the JDBC connection if you have already created some and select **Add Table** from the shortcut menu.
   - Right-click an existing table in the JDBC connection if there is and select **New Table** from the shortcut menu.

   The Add Table dialog appears.



2. From the Database Catalogs drop-down list, select the required catalog, then specify a schema in the selected catalog from the Schemas box.
3. Click the **Load Tables** button. The tables contained in the selected schema will then be displayed in the Tables box.
4. Check the **Include System Tables** option if you want to have the system tables available in the Tables box.
5. Check **Include Added Table** to make the tables that have been added to the connection available in the Tables box.
6. Choose the required tables in the Tables box, and then click **Add**.

   To choose consecutive tables, click the first table, press and hold down Shift, and then click the last table. To choose tables that are not consecutive, press and hold down Ctrl, and then click each table.
7. Repeat steps 2 to 4 to add tables from other catalogs and schemas.
8. To add certain tables by name or by patterns in their name, select the **Table Name Pattern** checkbox. The wildcard % will then be displayed in the text box.

   For example, if you want to add tables beginning with AL, you can type AL% (case sensitive) in the text box, and click the Load Tables button. All the tables beginning with AL in the selected schema will then be displayed in the Tables box. Choose the tables that are required by clicking Add. They will then be added to the connection.
9. After adding the required tables, click **Done** to close the dialog.

Notes:

- When you have added some tables to a catalog, you can then set up pre-joins between them. For detailed information, see Defining the join relationships between tables.
- If you have selected schemas in the Schema tab of the Get JDBC Connection Information dialog when setting up the connection, when you add tables, by default only the selected schemas will be listed in the Schemas box of the Add Table dialog, however, you can check the Show All Schemas option to show all the schemas. It is the same case when adding views.
- When adding tables, if the Include Added Table option is checked in the Add Table dialog, when you click the Add button to add a table that has already been added to the connection once, the Input Table Name dialog will be displayed, prompting you to give the new table a different name. It is the same case when adding views or synonyms.

Refreshing tables/views/synonyms

The tables/views/synonyms in your catalog are a temporary deposit to improve the performance when you design and test your report. Your database keeps changing over the time. However, these will not be reflected automatically in your catalog. To synchronize your tables/views/synonyms in the connection and database, you can choose to refresh the table/view/synonym information using the Refresh command on the shortcut menu of the table/view/synonym. Then when the refreshing job is done, a reporting dialog will be shown, summarizing the changes and operations that have been taken.

Organizing tables/views/synonyms

You can organize the tables/views/synonyms, which have been added to a catalog via a JDBC connection, by arranging them in different folders in the Catalog Browser.

To add a folder:

1. Right-click the **Tables/Views/Synonyms** node, or any existing folder and select **Add Folder** from the shortcut menu.
2. A new folder is added to the Tables/Views node, or in the selected folder. Type a name for the folder in the editing area, and then click outside to confirm the change.

To move tables/views/synonyms to a folder:

1. Right-click the table/view/synonym you want to move in the Tables/Views node and select **Move To** from the shortcut menu.
2. In the Move Table/View/Synonym dialog, select the folder to which the table/view/synonym will be moved, then click **OK**.

Removing and adding columns to tables/views/synonyms

By default, when you add a table/view/synonym, you add all the columns reside in the table/view/synonym. However, JReport gives you the flexibility to remove columns in a table/view/synonym, leaving only the columns useful to your reports. Additionally, your database keeps changing over the time. You can add the columns to the table after you delete them or the database has been updated.

To remove a column from a table/view/synonym:

1. Select the column you want to remove.
2. Right-click the column and select **Delete** from the shortcut menu.

To add columns to a table/view/synonym:

1. Right-click the table/view/synonym to which you want to insert columns, then select **Insert Column** from the shortcut menu.
2. In the Add Column dialog, select the required columns from the Columns box.

   There will be no available column in the Columns box if you do not delete any in the table/view/synonym before, or no update has been made to the database. That is to say, you cannot add columns that do not belong to the table/view/synonym.
3. Click the **Done** button to add the columns to the table/view/synonym.

Tip: If you want to add all the columns in a table/view/synonym at a time, you can also choose to refresh the table/view/synonym, which synchronizes the table/view/synonym in catalog with the one in your database.

Previewing the column data of tables/views/synonyms

In the Catalog Browser, you can preview the data for a database column, which enables you to have a general idea about the field contents, field type, and other useful information related to this field.

To preview the data of a column:

1. Right-click the column with the data you want to preview, then on the shortcut menu, click **Browse Data**.
2. The data of this column is shown in a list in a dialog, which also includes other information including the field type, precision, scale and number of records.

Stored procedures

A stored procedure is a compiled program, consisting of one or more statements and is saved into a database. The statement in a stored procedure can be an SQL statement or a control flow statement, such as an If-else statement or a Loop statement. A stored procedure is stored in the DBMS, so that it can be called locally or remotely. In addition, a stored procedure can return a value, single result set or multiple result sets. Currently, JReport supports stored procedures that return a single result set. If a stored procedure returns more than one result set, the first one will be used by JReport.

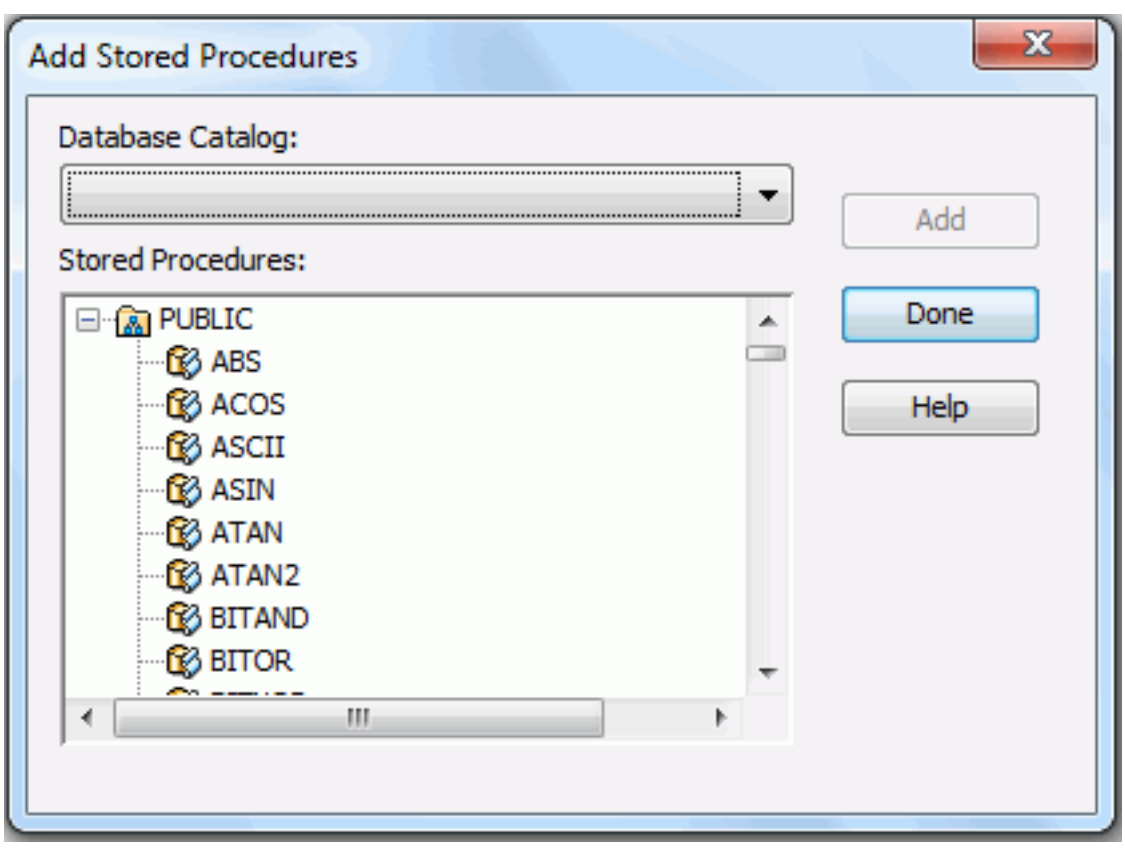
As a program, a stored procedure can take parameters. There are three types of parameters: IN, OUT and INOUT. The IN parameter lets you pass values to the procedure. The OUT parameter returns values to the caller. The INOUT parameter enables you to pass initial values to the procedure, and then returns the updated value to the caller.

In addition, JReport Designer provides you with the Data Manager which allows you to control the data retrieval of your stored procedures that function as queries, including the number of rows to be displayed and the duration required for the retrieval. It can also keep access information from previous runs of a stored procedure (for details, see Data Manager).

Adding stored procedures to a catalog

To add procedures stored in the database to a catalog via a JDBC connection, take the following steps:

1. In the Data tab of the Catalog Browser, right-click the JDBC connection node, and then click **Add Stored Procedure** on the shortcut menu. The Add Stored Procedures dialog appears.



2. The dialog lists all the stored procedures in the Stored Procedures box in a three-level tree. The top level is SQL-catalog, the second is SQL-schema and the last level is stored procedures. Select the required stored procedures, and then click **Add**.
3. If any of the selected procedure contains parameters, the Stored Procedure Parameters dialog will appear for you to input the required values. These values will be saved inside the stored procedure object, and will be used as the default value when executing this procedure. You can edit the stored procedure parameters at any time. For details, see Editing parameter values.
4. When finished, click the **Done** button to close the dialog.

When a stored procedure has been added into a catalog, JReport will in turn do the following:

- Retrieve the stored procedure's information from DBMS via JDBC.
- Prompt you to provide the values of IN and INOUT parameters if the stored procedure has them.
- Execute the stored procedure once in order to get the result set.
- Create the field objects of the procedure object according to the result set returned.

You will then be able to use the field objects to design your reports.

Notes:

- Not all database management systems (DBMS) support database stored procedures. In such a case, you may see nothing in the Procedures box.
- When adding a stored procedure, you may get the following error:

  [DBS-B]Could not find stored procedure 'test1;1'.

  This is because test;1 becomes quoted when JReport Designer finds the semicolon in it. To remove the quotation marks, you will have to modify your connection information. To do this:

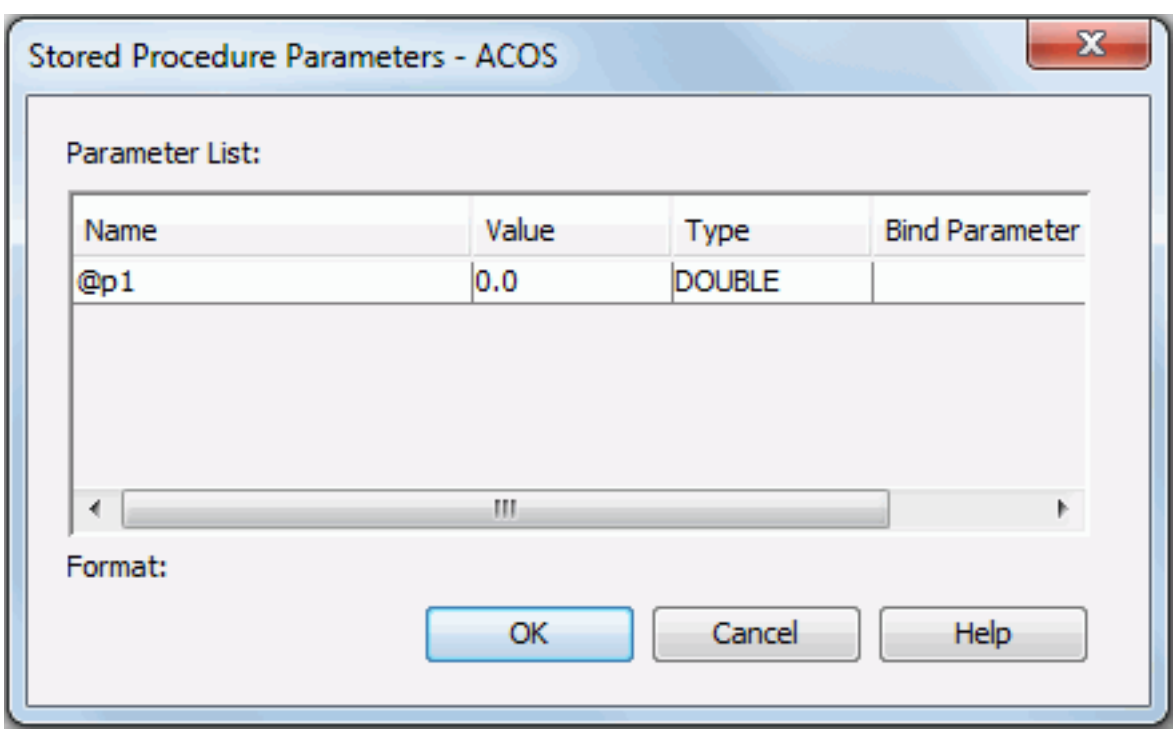
  1. Right-click your connection and select **Edit Connection** from the shortcut menu to bring out the Get JDBC Connection Information dialog.
  2. Select the **Qualified** tab, then in the Quote Qualifier box, check the **User Defined** radio button and remove all the characters from the Extra Characters and Quote Character text boxes.
  3. Click **OK** to confirm the settings.
- Due to the unique nature of Oracle stored procedures and EnterpriseDB stored procedures, you are unable to add them directly into a JReport Designer connection. Thus, the usage of JReport Designer is limited when you use an Oracle or EnterpriseDB database. As an alternative, JReport has developed the user data source API which can use a stored procedure in Oracle or EnterpriseDB. For details, see Oracle stored procedure UDS and EnterpriseDB stored procedure UDS.

Editing parameter values

When you add a stored procedure that contains parameters, you will be asked to input values for its parameters. These values will then be saved inside the stored procedure object in the catalog, and will be used as the default values when executing the stored procedure. You can edit stored procedure parameters at any time. Also, the IN type parameter is available for use with a stored procedure in a report the same as any other JReport parameter.

To edit the parameter values of a stored procedure:

1. Right-click the stored procedure and click **Parameters** on the shortcut menu. The Stored Procedure Parameters dialog appears.



2. This dialog lists all the IN and INOUT parameters in the stored procedures. Double-click the **Value** cell to edit the value of each parameter.
3. Double-click the **Bind Parameter Name** cell to bind the IN and INOUT parameters in the stored procedure to an existing parameter or constant level formula of the same type predefined in the catalog data source in which the JDBC connection is created, or to the special field User Name. You can input the parameter/formula name as @parametername/@formulaname, or the special field User Name as username. By default, the bound parameter will be the one with the same name of the IN and INOUT parameter which is automatically created when adding the stored procedure.
4. Click **OK** to apply the changes to the parameters.

Updating stored procedures

If you make any changes to stored procedures in the database, you will need to update them in the connection so that the reports built on them will work properly. To do this:

1. Select any stored procedure, right-click it, and click **Update** on the shortcut menu.
2. In the Add Stored Procedures dialog, select the stored procedures you want to update, and then click the **Update** button.

Developing reports from stored procedures

When you have added some stored procedures to a catalog, you can then use them to develop reports as required.

1. Make sure the catalog where the stored procedure is located is open, then click the **New Page Report** button on the Standard toolbar.
2. In the New Page Report dialog, specify the title and layout of the first report tab in the report as required, then click **OK**.
3. In the Data screen of the report wizard, choose the stored procedure with which you want to build the report tab, then follow the report wizard to create the report tab.

Note: When running a report which is built on a stored procedure, you will need to specify the values for the IN and INOUT parameters defined in the stored procedure. The values stored inside the stored procedure object in the catalog will be the default values. For details about specifying the values, see Editing parameter values in this document.

Creating cached query results for stored procedures

By default, when you run a report, JReport Engine fetches data from the database using the JDBC driver. For reports that are built on stored procedures which function as queries, JReport enables you to create cached result files and save them somewhere in your machine. Then, when you view these reports, you can choose to use the data from the cached query result file as opposed to the database.

For details about cached query results, see Cached query results.

Predefined SQL files

For users who wish to write their own SQL statement, JReport enables them to put the SQL statement into a file and then load them from this file. Note that at present each SQL file can only contain one SQL statement.

Usually, you can save the file containing the SQL statement as a plain text file (.txt). The SQL statement here supports the SQL 92 standard. Although for different databases, it may vary. The basic statement is:

```
SELECT...FROM...WHERE
```

Nested query is also supported.

The following is an example of an SQL file:

```
SELECT Catalog."Product Type Name", Catalog."Product Type ID",
Catalog.Description, Products.Category,
Products."Product ID", Products.Price, Products."Product Name",
Customers.Region, Customers."Contact Position", Customers.Country,
Customers."Customer ID", Customers.Address2, Customers.Address1,
Customers."Contact Title", Customers.Phone, Customers."Contact Last Name",
Customers.City, Customers.Fax, Customers."Contact First Name",
Customers."Annual Sales", Customers."Customer Name", Customers."Postal Code",
Orders."Order ID", Orders."Required Date", Orders."Customer ID",
Orders."Shipping Cost", Orders."Ship Date", Orders."Order Date",
Orders."Employee ID", Orders.Shipped, Orders."Payment Received",Orders."Ship Via"
FROM Catalog, Products, Customers, "Orders Detail", Orders
WHERE (Products."Product Type ID"=Catalog."Product Type ID")
AND ("Orders Detail"."Product ID"=Products."Product ID")
AND (Orders."Customer ID"=Customers."Customer ID")
AND (Orders."Order ID"="Orders Detail"."Order ID")
AND (( Customers.Country='USA' ))
```

The SQL files can work like queries in JReport, but there are two basic differences:

- A query is based on the tables/views added from a raw database, which means, if you have only set up a connection, but have not added any tables or views into the catalog, you will not be able to build a query. However, after you have set up a connection, even if you have not added any tables or views, you can still import an SQL file into a catalog, and then build reports on it.
- Queries built using the interactive query designer will work with other tables/views when you create or modify a working set. That is, if you design a report using interactively created queries, you can add additional objects from tables/views into the working set afterwards. However, when you design a report using an SQL file, you will only be able to use the contents already included in the SQL file. Due to this, you should make sure that you have included all the required fields in your SQL file.

In addition, JReport Designer provides you with the Data Manager which allows you to control the data retrieval of your SQL files that function as queries, including the number of rows to be displayed and the duration required for the retrieval. It can also keep access information from previous runs of an SQL file.

Adding SQL files to a catalog

After you have set up the JDBC connection and created some SQL files, you can then add them to a catalog. To do this:

1. In the Data tab of the Catalog Browser, right-click the JDBC connection, then click **Add SQL** on the shortcut menu to display the Select an SQL File dialog.
2. Browse to the SQL file and click **Open**.
3. In the Input SQL Name dialog, enter a name for the SQL file in the SQL Name text box, then click **OK** to add the SQL file to the connection.

Notes:

- The format of an SQL file, such as comments, is remained when it is added into a JReport catalog.
- If you hope to get different result set from the imported SQL at runtime, you can use parameters predefined in the catalog data source in which the JDBC connection is created in the WHERE clause of an SQL file to filter the dataset, and then reference them in the imported SQL directly by @ParamterName and :ParameterName. For example, in the above mentioned SQL file, if you want to use a parameter to return a result set in which the customer ID is greater than the parameter, then the WHERE clause would be like this:

  ```
  WHERE (Products."Product Type ID"=Catalog."Product Type ID") AND ("Orders
  Detail"."Product ID"=Products."Product ID") AND (Orders."Customer
  ID"=Customers."Customer ID") AND (Orders."Order ID"="Orders Detail"."Order ID") AND
  ( ( Customers.Country='USA' )) AND (Customers."Customer ID" > @id)
  ```

  Where, id is a parameter created in the catalog.
- You can use the special field User Name as @username in the WHERE clause of an SQL file to filter the dataset.
- You can also use constant level formulas predefined in the catalog data source in which the JDBC connection is created in the SQL file to calculate your data.
- In an Oracle database, if you want to use a Date or DateTime type parameter or a specific date or time in the WHERE clause of an SQL file to filter the dataset, you need to add the to_date() or to_timestamp() function in the clause, for example:

  ```
  WHERE (@COL_DATE = to_date(@p_DATE,'yyyy-mm-dd')
  WHERE (@COL_DATE = to_date('2003-12-06','yyyy-mm-dd')
  ```

Updating an SQL file

If you make any changes to SQL files in the database, you will need to update them in the connection so that reports built on them can work properly. To do this:

1. Select any SQL file, right-click it and select **Update** from the shortcut menu.
2. In the Select SQL File dialog, select the SQL files you want to update, and then click **Open**.
3. In the Select Catalog dialog, select a catalog from the From DB Catalog drop-down list.
4. Click the **OK** button to close the dialog.

Developing reports from SQL files

When you have added some SQL files to a catalog, you can then use them to develop reports as required.

1. Make sure the catalog where the SQL file locates is opened, then click the **New Page Report** button on the Standard toolbar.
2. In the New Page Report dialog, specify the title and layout of the first report tab in the report as required, then click **OK**.
3. In the Data screen of the report wizard, choose the SQL file with which you want to build the report tab, then follow the report wizard to create the report tab.

Creating cached query results for SQL files

By default, when you run a report, JReport Engine fetches data from the database using the JDBC driver. For reports that are built on imported SQL files which function as queries, JReport enables you to create cached result files and save them somewhere in your machine. Then, when you view these reports, you can choose to use the data from the cached query result file as opposed to the database.

For details about cached query results, see Cached query results.

XML connections

An XML connection contains relational data which is transformed from an XML data source.

JReport Designer supports parsing of the schema from XML instances, or importing them from XSD and then transforming them to relational tables. The tables can then be added to the current connection from the transformed relational tables and accessed in the same way as JDBC-supplied tables.

This section describes the following tasks:

- Importing an XML schema
- Transforming an XML schema to a relational schema
- Setting up an XML connection
- Managing tables in an XML connection

Importing an XML schema

Before an XML schema is transformed to relational tables, it must be imported. JReport Designer provides two ways for you to import an XML schema: importing from an XSD file or parsing from an XML instance. After importing, in order to provide a precision schema to be transformed, you can supplement or modify properties of the imported schema according to your requirements.

This section describes the following tasks related to importing XML schemas:

- Importing from an XSD file
- Importing by parsing an XML instance
- Supplementing information for the XML schema

Importing from an XSD file

In order to transform the imported XML schema to the relational schema, you can choose to import an XML schema by importing the structure of an XML data source and the information of its elements that are recorded in an XSD file.

You should be aware of the following about XSD in order to generate a correct imported XML schema:

Data type conversion table

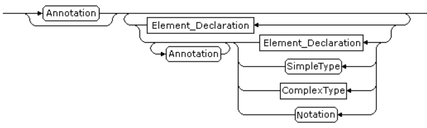
Before the data type defined in the XSD file can function with JReport Designer, it should first be converted into a corresponding data type when the XML schema is imported, following the rules in the conversion table below.

XSD Data Type	SQL Data Type
BOOLEAN	BIT
BYTE	TINYINT
UNSIGNEDBYTE	SMALLINT
SHORT	SMALLINT
UNSIGNEDSHORT	INTEGER
INT	INTEGER
UNSIGNEDINT	BIGINT
LONG	BIGINT
UNSIGNEDLONG	DECIMAL
NEGATIVEINTEGER	DECIMAL
NONPOSITIVEINEGGER	DECIMAL
POSITIVEINTEGER	DECIMAL
NONNEGATIVEINTEGER	DECIMAL
INTEGER	DECIMAL
FLOAT	FLOAT
DOUBLE	DOUBLE
DECIMAL	DECIMAL
STRING	VARCHAR
NORMALIZEDSTRING	VARCHAR
TOKEN	VARCHAR
LANGUAGE	VARCHAR
NAME	VARCHAR
NMTOKEN	VARCHAR
NMTOKENS	VARCHAR
NCNAME	VARCHAR
ID	VARCHAR
IDREF	VARCHAR
ENTITY	VARCHAR
IDREFS	VARCHAR
ENTITIES	VARCHAR
ANYSIMPLETYPE	VARCHAR
GYEAR	VARCHAR
GYEARMONTH	VARCHAR
GMONTH	VARCHAR
GMONTHDAY	VARCHAR
GDAY	VARCHAR
DATE	DATE
TIME	TIME
DATETIME	TIMESTAMP
HEXBINARY	BLOB
BASE64BINARY	BLOB

XSD syntax supported by JReport Designer

In JReport Designer, not all XSD syntax can be supported. The following diagram shows the XSD syntax that is supported by JReport Designer and syntax is in accordance with the W3C standard:

XML Schema::=



Text description of the XML Schema:

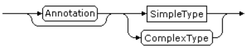
```
annotation*, ((element declaration | SimpleType| ComplexType | notation), annotation*)*;
```

Notes:

- When a ComplexType is being defined, the element of which cannot be declared as an anonymous ComplexType.
- Neither loop nor network structures can be defined in the ComplexType, so direct or indirect recursion cannot occur when defining a ComplexType. That is, the type of the element declared in the ComplexType cannot be that defined in the current direct or indirect context.
- Only the anonymous attribute definition is supported by JReport Designer.

The following are diagrams show the detail syntax of elements in the above diagram:

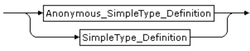
- **Element Declaration::=**



Text description of the Element Declaration:

```
element declaration:= annotation?, SimpleType | ComplexType;
```

- **SimpleType::=**



Text description of the SimpleType:

```
anonymous simpleType definition | simpleType definition;
```

Note: The SimpleType here should be of the schema built-in type, such as xs:string.

- **Anonymous SimpleType Definition::=**

Text description of the Anonymous SimpleType Definition:

```
annotation?, (restriction | list);
```

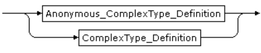
Note: List type here cannot support some functions, such as minLength, and maxLength.

- **SimpleType Definition::=**

Text description of the SimpleType Definition:

```
name, annotation?, (restriction | list)
```

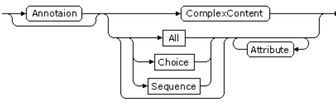
- **ComplexType::=**



Text description of the ComplexType:

```
anonymous ComplexType definition | ComplexType definition;
```

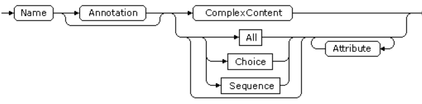
- **Anonymous ComplexType Definition::=**



Text description of the Anonymous ComplexType Definition:

```
annotation?, (ComplexContent | ((all | choice | sequence)?, attribute*));
```

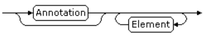
- **ComplexType Definition::=**



Text description of the ComplexType Definition:

```
name, annotation?, (ComplexContent | ((all | choice | sequence)?, attribute*));
```

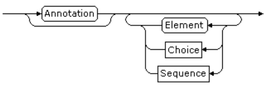
- **All::=**



Text description of the All:

```
annotation?, element*;
```

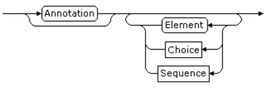
- **Choice::=**



Text description of the Choice:

```
annotation?, (element | choice | sequence)*;
```

- **Sequence::=**



Text description of the Sequence:

```
annotation?, (element | choice | sequence)*;
```

Namespace limitation in the XSD file

Default namespace (3w) and target namespace are supported by JReport Designer. You can define a prefix for the default namespace, such as xs, or xsd, but the value of elementFormDefault must be qualified. The value of attributeFormDefault should be unqualified, but JReport Designer will not allow you to add prefixes before the attributes of the elements.

Importing by parsing an XML instance

In order to transform the imported XML schema to the relational schema, you can also choose to parse an XML data source to get the XML structure. You should be aware of the following points about XSD in order to generate a correct imported XML schema.

- The way to import XML schema from XSD is recommended, because when you import an XML schema by parsing an XML instance, data types in the schema may not very accurate and all elements will be parsed in one tree.
- The XML schema structure will be parsed to a lot of complex elements which begin with XML root node. You can specify data types for the imported XML schema if you know data types accurately, otherwise it will be specified as default of JReport Designer. (For the details about data types and the XML structures supported by JReport Designer, see the section Importing from an XSD file).

Supplementing information for the XML schema

When importing an XML schema, you can modify the properties or supplement some necessary information of the XML schema according to your own requirements in order to make it more useful to you.

The following are some special properties and its description for your reference:

Name

Displays the name of an element (E), attribute (A) or a reference of element (E) in the XML instance.

Data Type

Specifies the data type of an element or attribute. The data type is imported when importing the XML schema from XSD file or given by JReport when parsing from XML source. This property on a reference of element cannot be modified.

Format Pattern Type

This property is activated when you modify properties for data of Date, Time or DateTime type in the XML schema. It can be Default or Customized:

- If Default is selected, the lexical formats inspired by ISO 8601 will be used for Date, Time or DateTime data following the specification XML Schema Part 2: Datatypes Second Edition.
- If Customized is selected, the Format Pattern dialog will appear for you to specify patterns to format the Date, Time or DateTime data in the XML schema (for details, see Appendix 1: Details of format patterns).

Format Pattern

This property is available when you specify to parse data from the XML instance at runtime.

Is Multiple

Indicates whether or not an element occurs more than once in the XML instance. By default, the value of the Is Multiple property is affected by the value of its maxOccurs property in the XSD: if the value of maxOccurs property is greater than 1 or unbounded, it is true; otherwise, it is false. You can modify this property according to your own requirements.

List

Specifies whether or not to set the data type of an element or attribute as List. This property on a reference of element cannot be modified.

- **List Delimiter**
  Specifies the delimiter to separate items in the list. Now only the following delimiters are supported: , . ; ! / \ | and space. Space will be specified by default.

The following two properties are alternative:

Default Value

Specifies the default value for an element or attribute. If there is no specific value of an element or attribute in the XML schema, the default value you specify here will be applied; otherwise, the value in the XML schema for this element or attribute will be applied.

Fixed Value

Specifies the fixed value for an element or attribute, which will be applied no matter there is a value for the element or attribute.

Transforming an XML schema to a relational schema

After you finish importing an XML schema, you can transform it to relational tables. In the transformation process, elements in the XML schema will be transformed to either tables or columns in tables according to the idiographic transformation rules, and the hierarchical logic in the XML schema will be maintained in the transformed relational schema.

Pick a topic from the following to get details about the transformation:

- Transformation rules
- XML hierarchical logic in relational schema

Transformation rules

When an XML schema is transformed to a relational schema automatically, the following transformation rules will be applied:

- A simple element in the XML schema will be transformed according to the following two rules:
  - If the Is Multiple property is set to true, elements will be transformed to a relational table. Contents of the simple element will be transformed to records of a column in the table and the column is named the same with the element.
  - If the Is Multiple property is set to false, contents of the simple element will be transformed to columns in the table transformed from the parent element of the simple element, and columns are named the same with the simple element.

    Note: The value of the Is Multiple property of a simple element is defined by the value of its maxOccurs property in the XSD. If you modify it and make it not match with the maxOccurs property in the transformation process, the element will be transformed according to the Is Multiple property regardless of the maxOccurs property. The following are the two conditions:

    - If the value of maxOccurs property of a simple element in XSD is greater than 1 or unbounded, but its Is Multiple property is modified to False in the transformation process, then only the last content of this element in XML instance can be transformed to a column.
    - If the value of maxOccurs property of a simple element in XSD is less than or equals to 1, but its Is Multiple property is modified to True in the transformation process, then the element will be transformed to a table with only one column and one record.
- Attributes of an element in the XML schema will be transformed to columns in the table, which is transformed from the element.
- A complex element in the XML schema will be transformed to a relational table. Contents of complex elements may contain three kinds of sub-nodes: text, simple element and sub-complex element.
  - Text in the XML will be ignored in the transformation process.
  - Each simple element will be transformed to a column in the table according to the rules above.
  - Each sub-complex element will be transformed to a table according to the rules above.

Below presents four illustrative examples for you to understand the transformation rules more easily.

Example 1: Transforming a simple element occurred more than once to a table

In the example, Employee is a simple element and its Is Multiple property is true according to the value of maxOccurs in the XSD which is 10, so it will be transformed to a table, named Employee. And contents of the simple element will be transformed to records of a column with the same name as the Employee table.

The following is a segment of an XSD file:

```
...
<xs:element name="Employee" maxOccurs="10" type="xs:string"/>
...
```

The following is a segment of the corresponding XML instance:

```
...
<Employee>John</Employee>
<Employee>Sally</Employee>
...
```

The following table, named Employee, is the result which is transformed from the simple element occurred more than once in the XML instance:

	Field Name	Allow NULL	Data Type
	NodePrimaryKey	☐	int
	NodeForeignKey	☑	int
	Employee	☑	varchar

In the result, these two columns - NodePrimaryKey and NodeForeignKey are generated by system automatically in the transformation process, and the column Employee is transformed from contents of the simple element.

Note: If the value of the maxOccurs property in the XSD is greater than 1 or unbounded, the simple element will occur more than once in the XML, and the Is Multiple property is true by default, but if you set Is Multiple to false in the transformation process, only the last element of the simple element in the XML will be transformed to a column of a table, which is transformed from the parent element of the simple element.

Example 2: Transforming a simple element occurred once or not occurred to a column

In the example, Employee is a simple element and its property Is Multiple is false according to the value of maxOccurs in the XSD which is less than or equals to 1, so the simple element Employee will be transformed to a column, named Employee in the Employees table which is transformed from the parent element.

The following is a segment of an XSD file:

```
...
<xs:element name="Employee" type="xs:string" maxOccurs="1"/>
...
```

The following is a segment of the corresponding XML instance:

```
...
<Employees>
  <Employee>John</Employee>
</Employees>
...
```

The following table, named Employees, is the transformed result which is transformed from the parent element of the simple element:

	Field Name	Allow NULL	Data Type
	NodePrimaryKey	☐	int
	NodeForeignKey	☑	int
	Employee	☑	varchar

In the result table, these two columns - NodePrimaryKey and NodeForeignKey are generated by system automatically in the transformation process, and the column Employee is transformed from the simple element.

Example 3: Transforming an attribute of an element to a column

In the example, the complex element - Employees will be transformed to a table, named Employees, the sub-elements Employee will be transformed to another table, named Employee, and attributes of these sub-elements will be transformed to columns in the Employee table.

The following is a segment of an XSD file:

```
...
<xsd:attribute name="id" type="xsd:long"/>
<xsd:attribute name="name" type="xsd tring"/>
<xsd:attribute name="age" type="xsd:int"/>
...
```

The following is a segment of an XML file:

```
...
<Employees groupId="1";>
<Employee id="1", name="John", age="23"/>
<Employee id="9" name="Sally" age="22"/>
...
</Employees>
...
```

The following table named Employees is transformed from the parent element - Employees:

	Field Name	Allow NULL	Data Type
	NodePrimaryKey	☐	int
	NodeForeignKey	☑	int

In the result table, these two columns NodePrimaryKey and NodeForeignKey are generated by system automatically in the transformation process.

The following table named Employee is transformed from sub-elements - Employee:

	Field Name	Allow NULL	Data Type
	NodePrimaryKey	☐	int
	NodeForeignKey	☑	int
	id	☑	bigint
	name	☑	varchar
	age	☑	int

In the result table, these two columns NodePrimaryKey and NodeForeignKey are generated by system automatically in the transformation process and the foreign key in the Employee is mapped to the primary key in the Employees table.

Example 4: Transforming a complex element to a table

In the example, the element StockMarket is of complex type and its property Is Multiple is true according to the value of maxOccurs in the XSD file which is 10, so StockMarket will be transformed to a table, named StockMarket and its sub-elements such as: Date, Open, High, Low, Close, Volume and ID will all be transformed to columns of the table automatically.

The following is a segment of an XSD file:

```
<xs:element name="StockMarket">
  <xs:complexType>
    <xs:choice maxOccurs="10">
      <xs:element name="Date" type="xs:date" />
      <xs:element name="Open" type="xs:double" />
      <xs:element name="High" type="xs:double" />
      <xs:element name="Low" type="xs:double" />
      <xs:element name="Close" type="xs:double" />
      <xs:element name="Volume" type="xs:double" />
      <xs:element name="ID" type="xs:long" />
    </xs:choice>
  </xs:complexType>
</xs:element>
```

The following is a segment of a corresponding XML file:

```
<StockMarket>
  <Date>1999-02-11</Date>
  <Open>11.5</Open>
  <High>12.4375</High>
  <Low>11.5</Low>
  <Close>12.4375</Close>
  <Volume>26600</Volume>
  <id>284</id>
</StockMarket>
```

The following table named StockMarket is the transformed result:

	Field Name	Allow NULL	Data Type
	NodePrimaryKey	☐	int
	NodeForeignKey	☐	int
	id	☐	int
	Date	☑	date
	Open	☑	float
	Low	☑	float
	Close	☑	float
	Volume	☑	bigint

In the result table, these two columns NodePrimaryKey and NodeForeignKey are generated by system automatically in the transformation process and the columns- id, Date, Open, Low, Close, Volume are transformed from sub-elements of the complex element StockMarket.

XML hierarchical logic in relational schema

The XML hierarchical logic will be maintained in the transformed relational schema. The parent-child relationship in the XML schema can be maintained by the following two ways:

- If the parent and children nodes are transformed to different tables, the parent-child relationship will be maintained by primary key and foreign key in tables, and this relationship can be reproduced by applying the join of the primary key and foreign key.
- If the children nodes can be transformed to columns, the parent-child relationship in the XML hierarchical logic will be maintained in the including relationship of table-column.

Setting up an XML connection

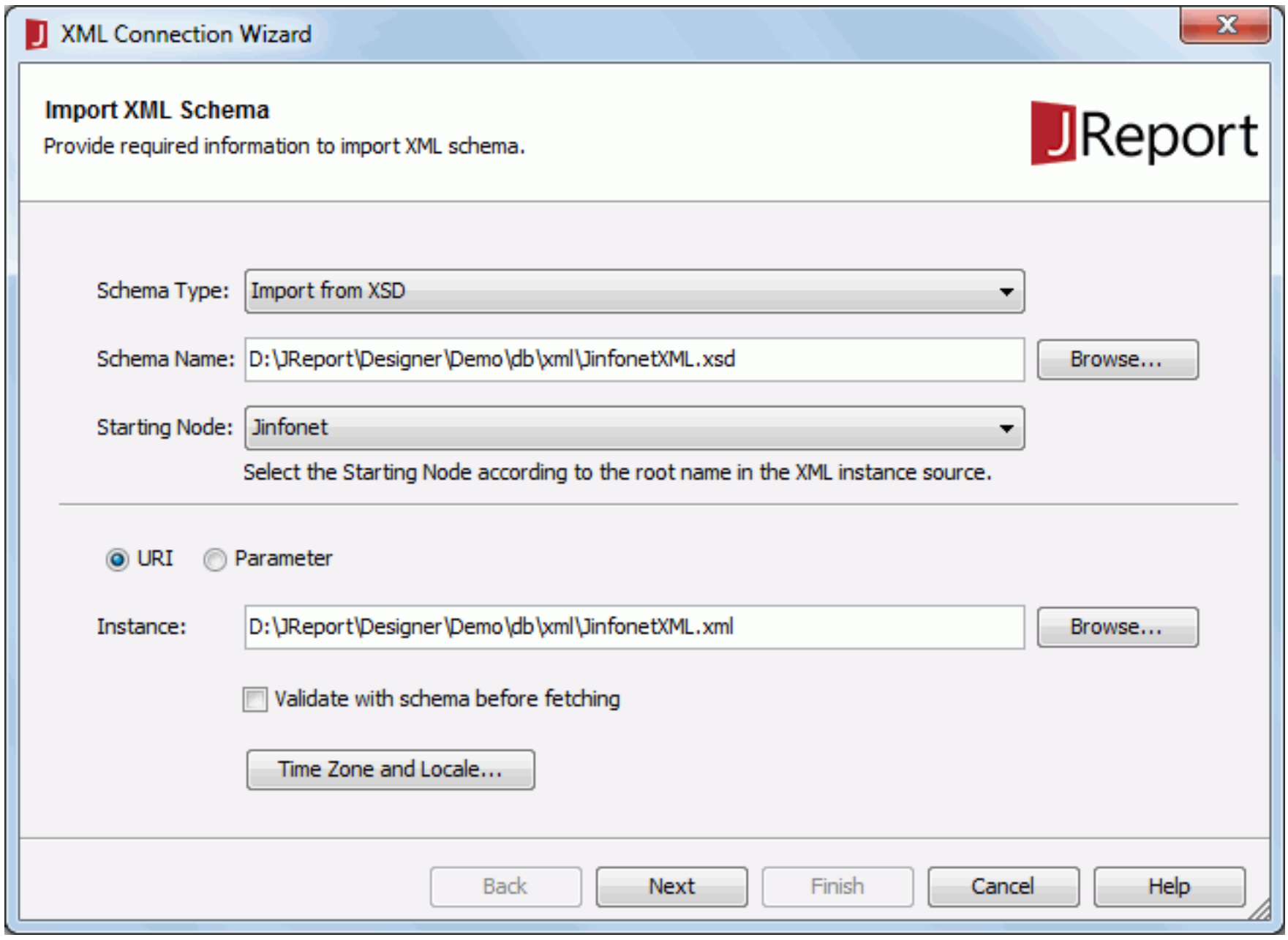
To set up an XML connection to connect a JReport catalog to an XML data source, follow the steps below:

1. Create a catalog or open a catalog, then in the Data tab of the Catalog Browser, right-click the node of a data source and choose **Add XML Connection** from the shortcut menu.

   If you want to set up the connection by adding a data source to an existing catalog and then defining the connection on the data source, open the catalog, select any data source contained in it, click the **New Data Source** button on the Catalog Browser toolbar, then in the New Data source dialog, specify the name of the data source, select the **XML** connection type and click **OK**.

   The XML Connection Wizard appears.

2. In the Import XML Schema screen, provide the required information for importing an XML schema.



   - If Import from XSD is selected from the Schema Type drop-down list,
     a. Click the **Browse** button next to the Schema Name text field to specify the schema file.
     b. From the Starting Node drop-down list, specify the starting node accordingly.
     c. Specify the XML instance file. You can specify the file directly or use a parameter to do this at runtime.

        To specify the file directly, check the **URI** radio button, and click **Browse** to specify the instance file. To use a parameter to specify the file at runtime, check the **Parameter** radio button and then select the parameter from the Instance drop-down list or click to create one.
     d. Check the **Validate with schema before fetching** to validate whether or not values in the XML instance is valid according to the W3C standard and the selected XSD schema.
     e. Click the **Time Zone and Locale** button to specify the time zone and the locale as required.
   - If Parse from XML Instance is selected from the Schema Type drop-down list, specify the instance file and time zone and the locale as required.

   When done, click **Next** to go to the next screen.

3. In the Modify Schema Properties screen, supplement and modify the element properties as required and then click **Next**.
4. In the Transform XML Schema screen, select an XPath to be the transforming start point, and then click **Next**.
5. In the Transformed Relational Schema screen, check the transformed result, and then click **Next**.
6. In the Add Table screen, add the required tables to the connection, and check the **Generate the default pre-join path** checkbox for generating default pre-join paths for tables of the XML data source.
7. Click **Finish** to confirm the transformed result and finalize the transformation process.

Managing tables in an XML connection

A table contains fields mapped to attributes, simple elements, contents of complex elements, and other nodes in XML files. Queries can be built on these tables and a report is developed from a query or something else which is functionally similar.

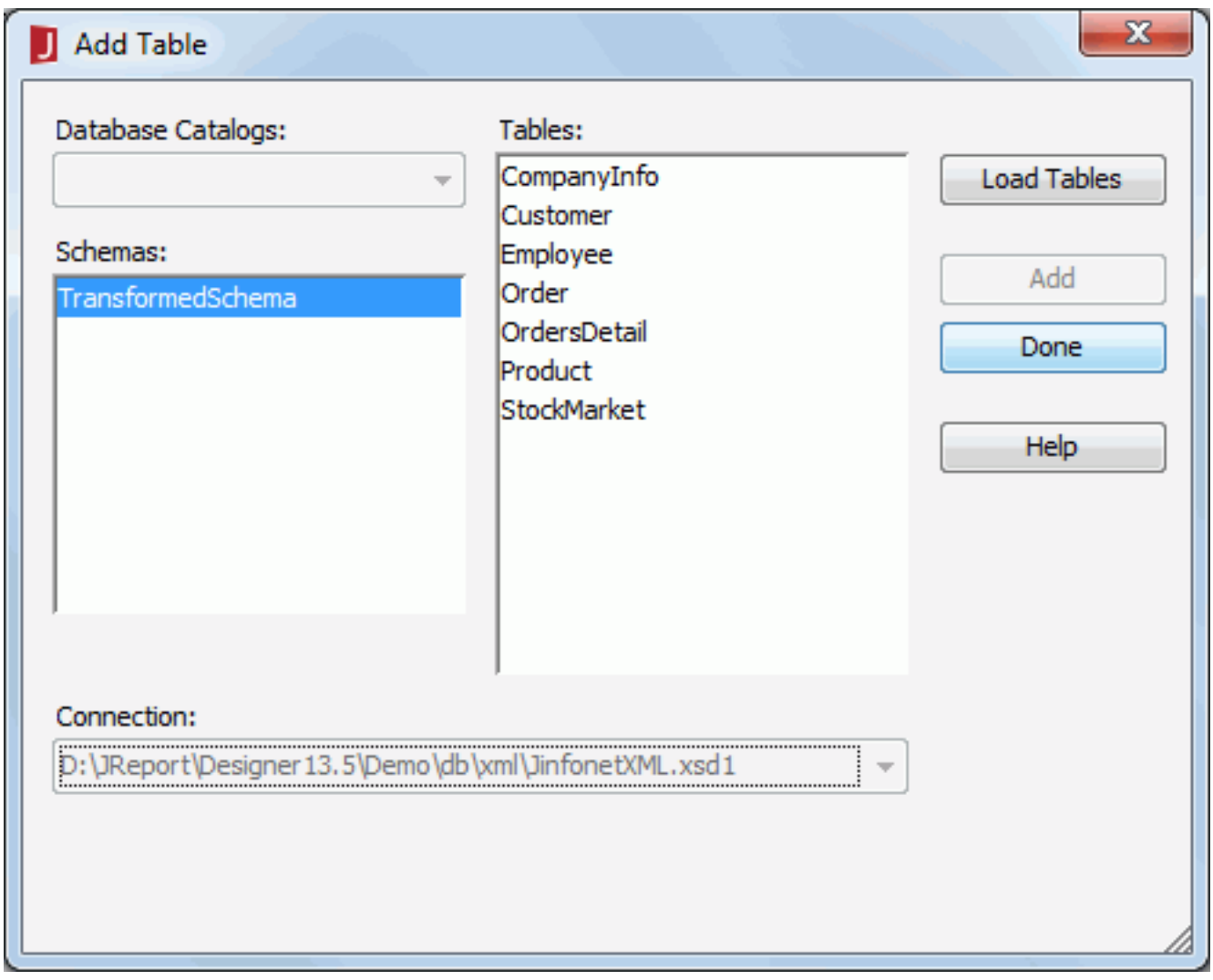
This document shows how to add more tables to an XML connection, organize the tables, and so on.

Adding tables

To add tables which are transformed from an XML file to an XML connection, follow the steps below:

1. Do either of the following:
   - Right-click the XML connection and select **Add Tables** from the shortcut menu.
   - Right-click the **Tables** node of the XML connection and select **Add Table** from the shortcut menu.
   - Right-click any folder in the Tables node of the XML connection if you have already created some and select **Add Table** from the shortcut menu.
   - Right-click an existing table in the XML connection if there is and select **New Table** from the shortcut menu.

   The Add Table dialog appears.



2. Click the **Load Tables** button. The tables contained in the schema that is transformed from the XML file will then be displayed in the Tables box.
3. Choose the required tables in the Tables box, and then click **Add**.

   To choose consecutive tables, click the first table, press and hold down the **SHIFT** key, and then click the last table. To choose tables that are not consecutive, press and hold down **CTRL**, and then click each table.
4. After adding the required tables, click **Done** to close the dialog.

Refreshing tables

The tables in your catalog are a temporary deposit to improve the performance when you design and test your report. Your data source keeps changing over the time. However, these will not be reflected automatically in your catalog. To synchronize your tables in the connection and data source, you can choose to refresh the table information using the Refresh command on the shortcut menu of the table. Then when the refreshing job is done, a reporting dialog will be shown, summarizing the changes and operations that have been taken.

Organizing tables

You can organize the tables in an XML connection by arranging them in different folders in the Catalog Browser.

To add a folder:

1. Right-click the **Tables** node, or any existing folder and select **Add Folder** from the shortcut menu.
2. A new folder is added to the Tables node, or in the selected folder. Type a name for the folder in the editing area, and then click outside to confirm the change.

To move tables to a folder:

1. Right-click the table you want to move in the Tables node and select **Move To** from the shortcut menu.
2. In the Move Table dialog, select the folder to which the table will be moved, then click **OK**.

Removing and adding columns to tables

By default, when you add a table, you add all the columns reside in the table. However, JReport gives you the flexibility to remove columns in a table, leaving only the columns useful to your reports. To remove a column from a table, right-click the column and select **Delete** from the shortcut menu.

Also, you can add the columns back to a table after you have removed them. To do this:

1. Right-click the table to which you want to insert columns, then select **Insert Column** from the shortcut menu.
2. In the Add Column dialog, select the required columns from the Columns box and click **Add** to add the columns to the table/view.

   There will be no available column in the Columns box if you do not delete any in the table before. That is to say, you cannot add columns that do not belong to the table.
3. Click the **Done** button to close the Add Column dialog.

Tip: If you want to add all the columns in a table at a time, you can also choose to refresh the table, which synchronizes the table in catalog with the one in your database.

User defined data sources

JReport Designer can access data from an external data source, such as a text file, or Lotus Notes, which is not stored in a database, or when there is no JDBC driver available. This feature uses the user data source API provided by JReport.

However, due to the unique nature of Oracle stored procedures, you will not be able to add them into a JReport catalog directly. The usage of JReport is therefore limited when you use an Oracle database. As a substitute, JReport has developed a user data source class that can use stored procedures in Oracle.

In addition, JReport Designer provides you with the Data Manager which allows you to control the data retrieval of your UDS that function as queries, including the number of rows to be displayed and the duration required for the retrieval. It can also keep access information from previous runs of a UDS (for details, see Data Manager).

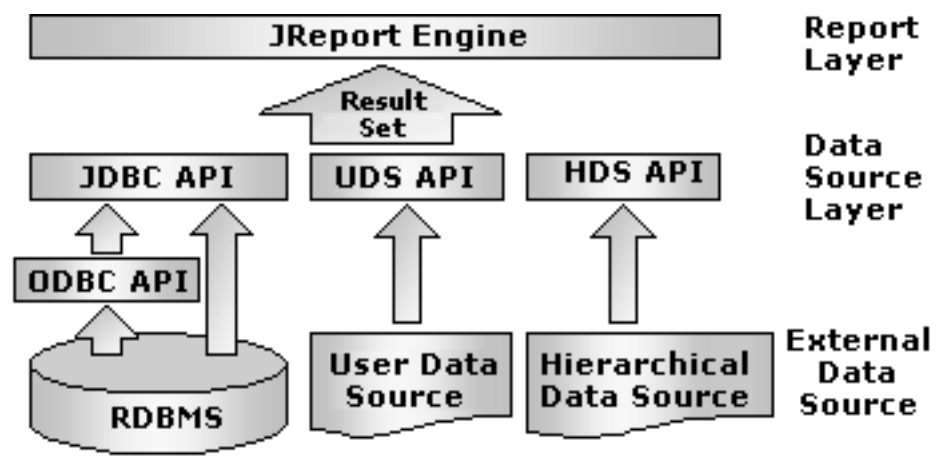
Pick a task from the following:

- User data source API
- Adding a UDS to a catalog
- Developing reports from UDS
- Oracle stored procedure UDS
- EnterpriseDB stored procedure UDS

User data source API

As a database application, JReport Designer can access data stored in a database through a JDBC connection. However, through the user data source API, JReport Designer can also access data from an external data source, such as a text file, or Lotus Notes, which is not stored in a database, or when there is no JDBC driver available.

The UDS API is a part of the JReport Data Access Model as shown in this diagram:



JReport Data Access Model

The user data source API:

- Is a Java interface that provides a dataset to JReport.
- Has a PARAMETER string, which can be changed when the UDS object is added into JReport Designer. The result set may differ according to the PARAMETER string. It is very flexible and convenient.

Notes:

- The PARAMETER here is the parameter string given to the UDS interface. That is, the UDS API provides a function getResultSet (String strParam), and PARAMETER is string strParam in this function. The following topics about UDS API will help make this information more clear. As a reminder, PARAMETER here is different from the parameter object in the JReport catalog. To distinguish between them, we have used an upper case PARAMETER to refer to the parameter string in the UDS, and a lower case parameter for the parameter object in a catalog.
- To design a report, JReport requires a meta data (see java.sql.ResultSetMetaData) which corresponds to the ResultSet that the data source has returned. To view or run a report, JReport requires a ResultSet object. The user data source must provide both of these two parts for JReport.

UDS API interface

To use the UDS API, the interface jet.datasource.JRUserDataSource must be implemented, and the class definition may be as follows:

```
import jet.datasource.*;
public class UserUDS
    implements JRUserDataSource
{
    // Class body.
}
```

Reference: JReport Javadoc jet.datasource.JRUserDataSource class in `<install_root>\help\api`.

The following are methods of the interface:

Public java.sql.ResultSet getResultSet(String strParam) throws JRUserDataSourceException

This method gets the result set for user data source.

- **PARAMETER**: strParam - A String value used to request and get different result sets. The format of the PARAMETER string is defined and parsed by yourself. You can enter the PARAMETER string when you add the UDS to a catalog. If you want to use parameters predefined in the Catalog Browser, you can add @paraName into the PARAMETER string.

  Keep in mind the parsing rules for the PARAMETER string when you enter the PARAMETER string.

  - The characters used for parameter name can be any character defined on the key board, except for quotation marks.
  - Parameter name is introduced with @, and sometimes colon : is also recognized as an introducer.
  - When your parameter string contains characters such as: At sign(@), ':', double quotation mark ("'"), or other strings that do not need to be parsed by JReport, you can use a pair of double quotation marks to quote them. For example, your parameter string may be:

    `jdbc:oracle:thin:@204.177.148.30:1521:orcl`

    Here, @ is a character used by the URL. If you add this parameter string into a catalog, JReport will regard 204 as the name of a parameter in the catalog. The correct form is:

    `"jdbc:oracle:thin:@204.177.148.30:1521:orcl"`

- **Returns**: java.sql.ResultSet - A row and column dataset that can be used in JReport.
- **Throws**: JRUserDataSourceException - If a database access error occurs.

  When you provide a java.sql.ResultSet instance, to reduce your work, you do not need to implement all methods of java.sql.ResultSet. You can just extend abstract class jet.datasource.JRResultSet. If you specify the column properties, for example, SQL data type, and precision, you will not need to provide a ResultSetMetaData instance, and JReport will not invoke getMetaData() function of ResultSet. However, if you do not specify the properties, you will have to implement the getMetaData () function and provide a ResultSetMetaData instance.

Public void releaseResultSet() throws JRUserDataSourceException

This method frees the ResultSet object.

- **Throws**: JRUserDataSourceException - If there are some errors when freeing the resource.

JDBC API used by UDS

In order to get data from user defined data source, JReport need to invoke the following JDBC API (for details about the API, refer to http://java.sun.com/javase/6/docs/api/).

java.sql.ResultSet

- boolean next() throws SQLException
- void close() throws SQLException
- boolean wasNull() throws SQLException
- String getString(int columnIndex) throws SQLException
- boolean getBoolean(int columnIndex) throws SQLException
- byte getByte(int columnIndex) throws SQLException
- short getShort(int columnIndex) throws SQLException
- int getInt(int columnIndex) throws SQLException
- long getLong(int columnIndex) throws SQLException
- float getFloat(int columnIndex) throws SQLException
- double getDouble(int columnIndex) throws SQLException
- byte[] getBytes(int columnIndex) throws SQLException
- java.sql.Date getDate(int columnIndex) throws SQLException
- java.sql.Time getTime(int columnIndex) throws SQLException
- java.sql.Timestamp getTimestamp(int columnIndex) throws SQLException
- java.io.InputStream getAsciiStream(int columnIndex) throws SQLException
- java.io.InputStream getBinaryStream(int columnIndex) throws SQLException
- ResultSetMetaData getMetaData() throws SQLException
- Object getObject(int columnIndex) throws SQLException
- int findColumn(String columnLabel) throws SQLException
- BigDecimal getBigDecimal(int columnIndex) throws SQLException
- Blob getBlob(int columnIndex) throws SQLException
- Clob getClob(int columnIndex) throws SQLException

java.sql.ResultSetMetaData

- int getColumnCount() throws SQLException
- boolean isCurrency(int column) throws SQLException
- int isNullable(int column) throws SQLException
- int getColumnDisplaySize(int column) throws SQLException
- String getColumnLabel(int column) throws SQLException
- String getColumnName(int column) throws SQLException
- String getSchemaName(int column) throws SQLException
- int getPrecision(int column) throws SQLException
- int getScale(int column) throws SQLException
- String getTableName(int column) throws SQLException
- int getColumnType(int column) throws SQLException

Note: The method ResultSetMetaData getMetaData() throws SQLException is only invoked when the Specify Columns option is not checked in the Add User Defined Data Source dialog.

Examples of using the UDS API

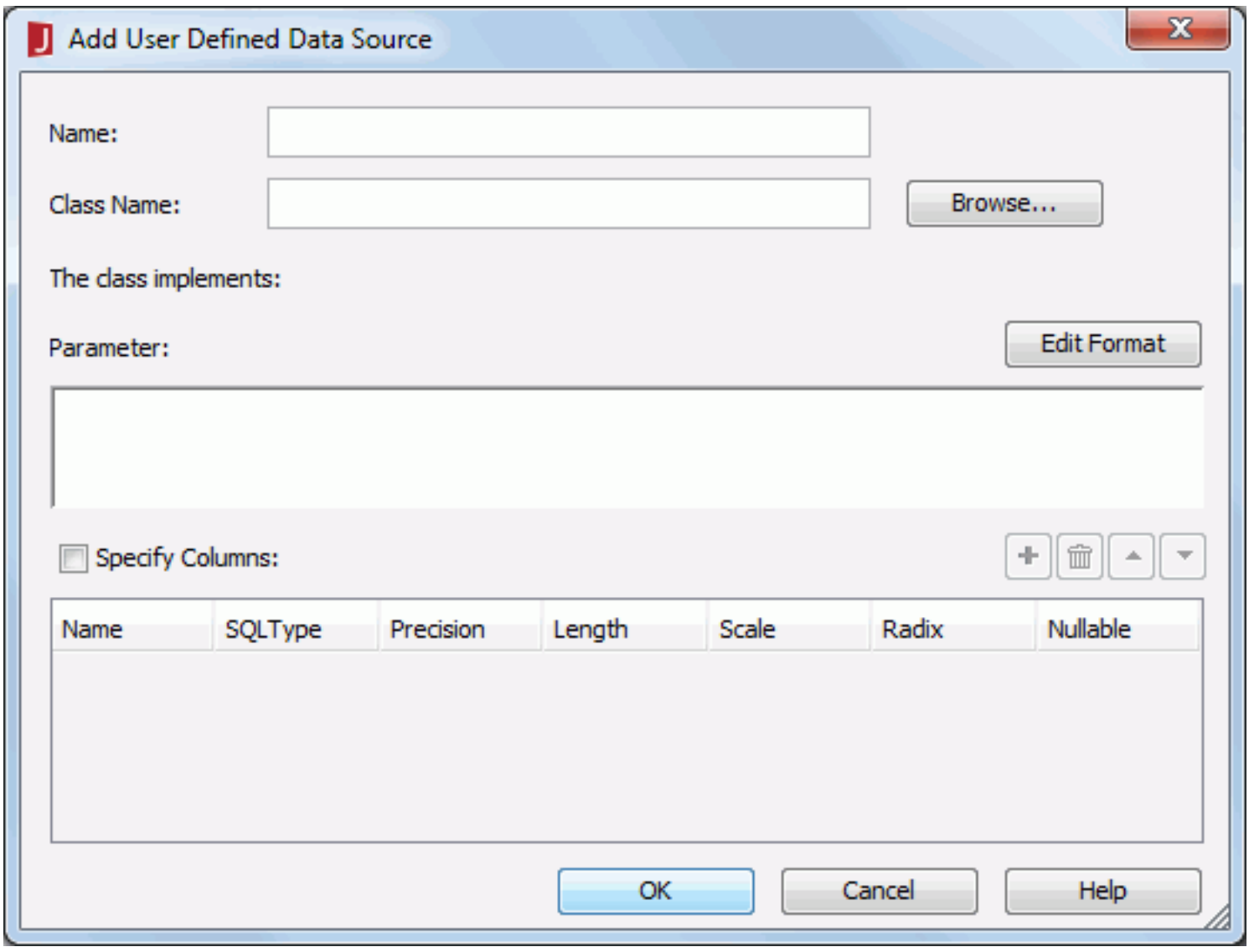
The user data source API is flexible and convenient to use. Before you implement it, you should first make an overall consideration of the architecture. JReport provides you with several scenarios which use the JReport UDS API. You can refer to them for assistance.

- When there is no JDBC driver available for an external data source (for example, a comma-delimited text file), you can create a ResultSet object to return the data to the requesting report. See Example 1: Developing reports from flat file UDS.
- You can return a ResultSet through the JDBC API. See Example 2: Developing reports from SQL data source UDS.
- Oracle stored procedure UDS. See Oracle stored procedure UDS.

Adding a UDS to a catalog

To add user defined data sources to a catalog, follow the steps below:

1. Open the catalog to which you want to add the UDS.
2. In the Data tab of the Catalog Browser, right-click the catalog data source to which the UDS is to be added, then select **Add User Defined Data Source** from the shortcut menu. The Add User Defined Data Source dialog appears.



3. In the Name field, specify a name for the UDS as required.
4. Provide the class name with package name in the Class Name field. You can also click **Browse** to find the class file. The class you entered should actually exist and can be found by JReport Designer, which means the class should be in the class path of the system environment. After filling in this field, the class name of the interface that the class implements will be displayed automatically behind The class implements:.
5. Specify the parameter for the UDS in the Parameter box. The parameter string must match the format defined in the UDS class. You can use parameters or constant level formulas predefined in the current catalog data source or the special field User Name in the parameter string, and the format of the predefined parameters used in the string is "@" + parameter name. For example, if the parameter string of a UDS is SQL=select * from employee, and you want to use the parameter sql in a catalog to replace the part after "=" in the string, then the parameter string of the UDS will be SQL=@sql.
6. Check the **Specify Columns** option to enable the column definitions list and add column definitions. If you don't specify the column definitions, JReport will get them from the result set automatically.
   - **Name**
     The name of a column. It should have the same validation with a common table column. The default names for column definitions are ‘column1', ‘column2', and so on.
   - **SQLType**
     The data type of the column.
   - **Precision**, **Length**, **Scale**, **Radix**
     The default value for each SQL type. Click the cell and modify the value if necessary.
   - **Nullable**
     Indicates whether the value of the column can be null. X stands for NoNulls, •stands for Nullable and ? stands for Nullable Unknown.
7. Click **OK** to add the UDS.

See also the examples in Developing reports from UDS for details about how to add some specific UDS to a catalog.

Note: JReport Designer supports the Dynamic UDS feature. This feature improves performance significantly by retrieving the selected fields. At runtime, an option is provided for you to pick up the columns you want to see in the report. Also, JReport Designer will generate dynamic report according to user's selection. For details, see Example 4: Developing reports with dynamic UDS.

Developing reports from UDS

When you have added the required UDS to a catalog, you can then use it to develop reports as you like.

To develop a report from a UDS:

1. Make sure the catalog where the UDS is located is open, then click the **New Page Report** button on the Standard toolbar.
2. In the New Page Report dialog, specify the title and layout of the first report tab in the report as required, then click **OK**.
3. In the Data screen of the report wizard, choose the UDS with which you want to build the report tab.
4. Follow the report wizard to create the report tab.

The following examples show how to develop report from some specific UDS:

- Example 1: Developing reports from flat file UDS
- Example 2: Developing reports from SQL data source UDS
- Example 3: Developing reports from Java object data source UDS
- Example 4: Developing reports with dynamic UDS

Example 1: Developing reports from flat file UDS

This example is a plain text file AddressesList.txt (available in `<install_root>\help\samples\APIUDS\txtUDS`), which contains data information in a dummy table:

```
Laurena    Croft    34826 Atwood St.    New York City    NY    10004    USA...
Jonathan    Hopkins    5062 Brandon Green Ave.    Minneapolis    MN    55402    USA...
Jeremy    Miner    9283 Cherry Leaf Lane    Palo Alto    CA    94303    USA...
....
```

You will notice that each line sequentially records the name, address, state, region, zip code, country, e-mail address and other information. In this example, you will be shown how to add an UDS using the data stored in a formatted text file. You can modify this example to meet your requirements.

Compiling and running the Address UDS

There are three classes used in this example and their source code are AddressListUDS.java, AddressListResultSet.java and AddressListResultSetMetaData.java. The data file is AddressesList.txt. All these files are available in `<install_root>\help\samples\APIUDS\txtUDS`.

If you use your own UDS classes, you must be sure that the class files can be found when running. So, make sure that the directory is classpath/package name. For this demo's UDS classes, the classes belong to a package named help. Copy the above necessary files to `<install_root>\help`, and add the additional entry to the ADDCLASSPATH variable of the batch file setenv.bat in `<install_root>\bin`.

For example, on Windows, if you installed JReport Designer into the default directory `C:\JReport\Designer`,

1. Copy the above necessary files to `C:\JReport\Designer\help`.
2. Compile the Java files.

   ```
   javac -classpath "C:\JReport\Designer\lib\JREngine.jar" Address*.java
   ```

3. Add the additional entry into the ADDCLASSPATH variable of setenv.bat in `C:\JReport\Designer\bin`.

   ```
   set ADDCLASSPATH=%JAVAHOME%\lib\tools.jar;C:\JReport\Designer\help;
   ```

Adding the Address UDS to a catalog

After compilation, you can now add the UDS to a JReport catalog.

1. Start JReport Designer with the batch file you just modified.
2. Open an existing catalog.
3. In the Data tab of the Catalog Browser, right-click the data souce to which the UDS is to be added, then select **Add User Defined Data Source** from the shortcut menu.
4. In the Add User Defined Data Source dialog, enter the following information:
   - **Name**: The name you want to use for identifying the UDS in JReport. Here we use Address.
   - **Class Name**: The class name of the UDS. You can type one in or click Browse to select a class file. Here we use help.AddressListUDS.
   - **Parameter**: AddressesList.txt
5. Click **OK** to add the UDS.

Developing a report with the Address UDS

Now that the Address UDS has been successfully added to the catalog, you can use it to develop reports as required. For details about how to create reports in JReport Designer, see Creating a page report.

Example 2: Developing reports from SQL data source UDS

This is a fairly complex example, which executes an SQL statement and returns a result set through the JDBC API.

Below is the Java code for class definition:

```
import jet.datasource.*;

public class SQLDataSource implements JRUserDataSource {
        // Define data.
        public ResultSet getResultSet(String param)
                        throws JRUserDataSourceException {
                // Method body.
        }

        /**
         * Free the resources such as: Connection, Statement, ResultSet.
         */
        public void releaseResultSet() throws JRUserDataSourceException {
                // Method body.
        }
}
```

The following explains the above code.

- The format of the String variable parameter is DRIVER=driver&URL=url&USER=user&PSWD=password&SQL=sql.

  For example, we used HSQL DB as data source, and `C:\JReport\Designer\Demo\db\JRDemo` as the hsql db path; "sa" as the user's name, "" as the password, and `select * from authors` as the SQL string. The parameter string may be similar to:

  ```
  DRIVER="org.hsqldb.jdbcDriver"&URL="jdbc:hsqldb:C:\\JReport\\Designer\\Demo\\db\
  \JRDemo"&USER=sa&PSWD=&SQL=select * from employee
  ```

  If you want to use predefined parameters in the JReport catalog, you can input the parameter string as:

  ```
  DRIVER="org.hsqldb.jdbcDriver"&URL="jdbc:hsqldb:C:\\JReport\\Designer\\Demo\\db\
  \JRDemo"&USER=@un&PSWD=&SQL=@sql
  ```

  Where, @sql is a parameter predefined in the catalog, and its default value is "select * from employee". The default value of @un is "sa".

  > **Note:** Every time when running, java.sql.Resultset must return the same meta data (UDS fields), including the number and order of fields, and the field properties such as column name, SQL type, precision, scale, nullable, currency, and array.

- The getResultSet() method will parse the parameter string, build a connection to the URL, and then execute the SQL statement.
- The releaseResultSet() method will release the resource such as Connection, Statement, and ResultSet.

Compiling and running SQLDataSource

There is one class used in this example and its source code is SQLDataSource.java, which is available in `<install_root>\help\samples\APIUDS\sqlUDS`. In this example, SQLDataSource will return the result set from demo hsql db. You should set up the HSQL DB path in the URL string and point it to the demo database at `<install_root>\Demo\db\JRDemo`.

Copy SQLDataSource.java to `<install_root>\help`, and compile it to generate SQLDataSource.class. Append the path `<install_root>\help` to the ADDCLASSPATH variable of the batch file setenv.bat in `<install_root>\bin`, so that SQLDataSource can be found at runtime.

Adding the SQLDataSource UDS to a catalog

After compilation, you can now add the UDS to a JReport catalog.

1. Start JReport Designer with the batch file you just modified.
2. Open an existing catalog.
3. In the Data tab of the Catalog Browser, right-click the data source to which the UDS is to be added, then select **Add User Defined Data Source** from the shortcut menu.
4. In the Add User Defined Data Source dialog, enter the required information.

   If you check the Specify Columns option, the column definitions list will be enabled and you can add column definitions. If you don't specify column definitions, JReport will obtain them from the result set automatically.

   Here, we will not specify column definitions. Instead, we will use the default ones from the ResultSet and the ResultSetMetaData. Enter **employees** in the Name field and **SQLDataSource** in the Class Name field. In the Parameter box, enter:

   DRIVER="org.hsqldb.jdbcDriver"&**URL**="jdbc:hsqldb:C:\\JReport\\Designer\\Demo\\db\\JRDemo"&**USER**=sa&**PSWD**=&**SQL**=select * from employee

   Make sure that the five bold parts in the above line must be capitalized.
5. Click **OK** to add the UDS.

Developing a report with the SQLDataSource UDS

Now that the SQLDataSource UDS has been successfully added to the catalog, you can use it to develop reports as required. For details about how to create reports in JReport Designer, see Creating a page report.

Example 3: Developing reports from Java object data source UDS

The JReport Designer UDS can use any Java object as a data source for you to create reports. To implement this function, JReport Designer provides a class UDSForJavaBean. Inside the UDSForJavaBean, which the getResultSet() method creates instances for the interface JavaBeanDataProvider, and then uses the method init() to initialize the data of the created instances.

Below is the Java code for class definition:

```
public class UDSForJavaBean implements JRuserDataSource
{
        // Define data.
        public ResultSet getResultSet(String param)
                        throws JRUserDataSourceException
        {
                // Method body.
        }

        /**
        * Free the resources such as : Connection, Statement, ResultSet.
        */
        public void releaseResultSet() throws JRUserDataSourceException
        {
                // Method body.
        }
}
```

The interface JavaBeanDataProvider is defined as follows:

```
public interface JavaBeanDataProvider
{
  public void init(String dataID, Properties initprops)
                  throws JavaBeanDataProviderException;
  public Class getMetaDataJavaBeanClass();
  public void setMetaDataJavaBeanClass(Class pClass);
  public object next() throws JavaBeanDataProviderException;
  public boolean requireDetails(String collectionPropName);
  public int getMaxShareTimes(String collectionPropName);
  public int getTimeoutforSubCollection(String collectionPropName);
  public void exit()throws JavaBeanDataProviderException;
}
```

For examples of how to create reports using the Java object data interface, JReport provides you with two demos which are available in <install_root>\help\samples\API\UDS\JavaUDS.

- Demo 1: Creating reports from simple Java object UDS
- Demo 2: Creating reports from Java object UDS with multi-levels of collections

Before running the demo, you need to do the following:

1. Copy all the content in <install_root>\help\samples\API\UDS\JavaUDS to <install_root>\help\UDSForJavaBean. You need to create the UDSForJavaBean directory.
2. Compile all the java files.

```
javac -classpath <install_root>\help\UDSForJavaBean;<install_root>\lib\JREngine.jar;<install_root>\lib\log4j-1.2.8.
jar; <install_root>\help\UDSForJavaBean\report\uds\javabean*.java
```

3. Copy data.txt in <install_root>\help\UDSForJavaBean to <install_root>\bin.
4. Add the path <install_root>\help\UDSForJavaBean in the ADDCLASSPATH variable in setenv.bat in the <install_root>\bin directory.

Demo 1: Creating reports from simple Java object UDS

In this demo, a Java object named Person will be used as the data source, you can find the person.java file in <install_root>\help\UDSForJavaBean\report\uds\javabean\demos. It is defined as follows:

```
public class Person implements Serializable
  {
    private String gender;
    private int mySalary;
    private String emailAddress;
    private Date birthDate;
    private long mediumWeight;
    private Vector isDataVerified;
    private Name name;
    private Address currentMailingAddress;
    private Phone currentWorkPhone;
/**
* @return the birthDate.
*/
public Date getBirthDate()
{
    return birthDate;
}
  }
```

Similar to the code above, other attributes of the Person object are defined by other Java objects, such as currentMailingAddress which is defined by the Address class.

Creating a report from the Person Java object

Take the following steps:

1. In JReport Designer, open an existing catalog, then in the Data tab of the Catalog Browser, expand the data source to which the UDS is to be added.
2. Create a parameter as follows, in order to get the real collection of the data objects, create two parameters before importing the Person Java object as follows:

   Name : pUseFakeData (Specify whether to use generated data when running the report.)
   Value Type : Boolean
   Prompt Value : True

   Name : pNumOfFakeData (Specify the number of data records to generate that will be shown when running the report.)
   Value Type : Integer
   Prompt Value : Any integer number

   For details about how to create parameters, see Creating a parameter.
3. Right-click the data source node, and select **Add User Defined Data Source** from the shortcut menu.
4. In the Add User Defined Data Source dialog, enter **Person** in the Name text field.
5. For the Class Name field, click the **Browse** button, go to <install_root>\help\UDSForJavaBean\report\uds\javabean, and then choose **UDSForJavaBean.class**, which will import the class jreport.uds.javabean.beans.Person with full class name.

Creating a report from the Person Java object

Take the following steps:

1. In JReport Designer, open an existing catalog, then in the Data tab of the Catalog Browser, expand the data source to which the UDS is to be added.
2. Create a parameter as follows, which is used to specify the number of data records to generate that will be shown when running the report.

   Name : pNumOfFakeData
   Value Type : Integer
   Prompt Value : Any integer number. Note that the parameter must have at least one value that is larger than 0, otherwise you will get exceptions when viewing reports.

   For details about how to create parameters, see Creating a parameter.
3. Right-click the data source node and click **Add User Defined Data Source** on the shortcut menu.
4. In the Add User Defined Data Source dialog, type **SimpleTestBean** in the Name text field.
5. For the Class Name filed, click the **Browse** button, go to <install_root>\help\UDSForJavaBean\report\uds\javabean, and then choose **SimpleTestBean.class**. The UDS class SimpleTestBean.class will import the class jreport.uds.javabean.beans.SimpleTestBean with full class name.
6. In the Parameter box, type in the following string:

   JavaBeanDS_DataProvider:jreport.uds.javabean.GenericBeanDataProvider
   GBeanDS_RuntimeDataID:GBeanProvider_BeanClsName=jreport.uds.javabean.beans.SimpleBeanTest
   GBeanProvider_FakeDataSubCollectionDefaults:jreport.uds.javabean.beans.Person,1,30000
   GBeanProvider_RptDataInitializer:jreport.uds.javabean.SubRptCollectionDataInitializer, which is our built-in data provider set up especially for the subreport to return the collection of data by referencing the sub-collection in the primary report. The referencing information is passed via the parameter value @pRunTimeDataInfo which will be used when we set up the link between the primary report and the subreport.

   - The class name for GBeanProvider_BeanClsName is Person because the subreport will use the Person object.
7. Click **OK**, and the UDS PersonAsSubRpt is added.
8. Create a page report named SubRpt.cls, and create a table report in it as the subreport based on the UDS PersonasAsSubRpt.

Linking the primary report and the subreport

1. In the Catalog Browser, create a formula named NotUseFakeData to return false all the time, for example, `return false`. This formula will be passed into the subreport as the value of the parameter pUseFakeData in the subreport, so that when the subreport runs with the primary report, it will always use the data from the primary report instead of constructing the fake data itself.
2. Open MainRpt.cls, add a new detail panel into the report.
3. Selected the newly added panel and click **Insert > Subreport**. When a box attached to your mouse pointer, click in the panel and the Subreport dialog is displayed.
4. Click the **Browse** button, choose **SubRpt.cls** as the subreport, then in the Parameters tab, specify values for the parameters as follows:

Parameter	Value
pCopyOfDataSrc	pNumOfFakeData
pUseFakeData	NotUseFakeData (gives a false value to the parameter)
pRunTimeDataInfo	persons (specifies to use the sub-collection persons of the primary report as the data source of the subreport at runtime)

5. Click **OK** to insert the subreport.
6. Run the primary report, and you will find that the corresponding persons information is displayed in the subreport.
7. If you want to insert another subreport which shares the same sub-collection with SubRpt.cls, you should modify the value of the key word GBeanProvider_ListOfDetailProps in the primary report UDS parameter string to GBeanProvider_ListOfDetailProps=persons,2,30000. That changes the share amount of the persons property from 1 to 2.

For more information about subreport in JReport, see the section Subreports.

Example 4: Developing reports with dynamic UDS

JReport Designer supports the Dynamic UDS feature. This feature improves performance significantly by retrieving the selected fields. At runtime, an option is provided for you to pick up the columns you want to see in the report. In this way, JReport Designer generates a dynamic report according to your selection.

In this example, SQLDataSource is used to illustrate the usage and effect of the Dynamic UDS feature. Assume that you have generated SQLDataSource.class, start JReport Designer with the modified batch file (for details, see Compiling and running SQLDataSource).

1. Open an existing catalog.
2. In the Data tab of the Catalog Browser, expand the data source to which the UDS is to be added.
3. Right-click the **Parameters** node and select **Add Parameter** from the shortcut menu.
4. In the Create Parameter dialog, enter the following:
   - **Name**: sql
   - **Value Type**: String
   - **Prompt Values**:

     select * from employee
     select salary from employee
     select employeeid, employeeposition, hiredate, notes, salary, photo from employee

     For details about how to create a parameter, see Creating a parameter.
5. Right-click the data source node, and then select **Add User Defined Data Source** from the shortcut menu.
6. In the Add User Defined Data Source dialog, enter the following information:
   - **Name**: employees
   - **Class Name**: SQLDataSource
   - **Parameter**: `DRIVER="org.hsqldb.jdbcDriver"&URL="jdbc:hsqldb:C:\\JReport\\Designer\\Demo\\db\\JRDemo"&USER=sa&PSWD=&SQL=@sql`

   The UDS employees will now have been added to the catalog.
7. Click on the Standard toolbar to open an existing report, and then select **File** > **New** > **Page Report Tab**.
8. In the New Page Report Tab dialog, specify the name and layout of the report tab as required and then click **Create**. Here, we choose to create a table report.
9. In the Data screen of the Table Wizard, check the **New** radio button and choose the UDS **employees** from the User Defined node. Then, click **Next** to go to the next tab.
10. In the Display screen, add the fields EMPLOYEEID, employees_NOTES, employees_SALARY and HIREDATE to be displayed in the table, edit their display names to **ID**, **Notes**, **Salary**, **Hire Date**, and then click **Next**.
11. In the Group screen, specify to group on the field **EMPLOYEEPOSITION**.
12. Skip the Summary screen and in the Layout and Style screen, specify to display the report in the Warm style.
13. Click **Finish** to create the report.
14. Click the **View** tab to view the report. You will then be prompted to enter a parameter.
15. Select **select * from employee** as the parameter, and you will see that data for all fields has been retrieved.

ID	Notes	Salary	Hire Date
Marketing			
3	j.miner@marketing.gourmetjava.co	50,000.00	05/19/1979 12:00:00 AM
5	g.schultz@marketing.gourmetjava.	49,000.00	10/20/1983 12:00:00 AM
10	m.everett@marketing.gourmetjava	41,000.00	09/30/1987 12:00:00 AM
13	d.edmund@marketing.gourmetjav	49,000.00	08/14/1980 12:00:00 AM
President			
6	k.chen@gourmetjava.com	60,000.00	10/05/1978 12:00:00 AM
Sales Representative			
1	l.croft@sales.gourmetjava.com	40,000.00	09/12/1984 12:00:00 AM
2	j.hopkins@sales.gourmetjava.com	42,000.00	04/03/1987 12:00:00 AM
7	j.rayner@sales.gourmetjava.com	45,000.00	07/23/1990 12:00:00 AM
9	l.fung@gourmetjava.com	50,000.00	02/07/1981 12:00:00 AM
11	e.packard@sales.gourmetjava.cor	42,000.00	05/04/1985 12:00:00 AM
12	j.westray@sales.gourmetjava.com	42,000.00	07/15/1980 12:00:00 AM
15	m.olympus@sales.gourmetjava.cc	43,000.00	08/07/1986 12:00:00 AM
Vice President			
4	j.lee@gourmetjava.com	55,000.00	06/09/1981 12:00:00 AM
8	w.penne@gourmetjava.com	50,000.00	11/16/1979 12:00:00 AM
14	j.compton@gourmetjava.com	55,000.00	05/25/1980 12:00:00 AM

16. Go back and run this report again. This time, choose **select salary from employee**. You will notice that no groups are divided and the group name changes to NULL. This is because only the field employees_SALARY has been selected this time. JReport Designer will make no reference to the employees_Position column, on which the group is based.

Salary
40,000.00
42,000.00
50,000.00
55,000.00
49,000.00
60,000.00
45,000.00
50,000.00
50,000.00
41,000.00
42,000.00
42,000.00
49,000.00
55,000.00
43,000.00

Note: To make a dynamic UDS work fine, if the SQL statement at runtime doesn't include all the UDS columns, you need to make sure that none of the UDS columns' properties is edited, that is the Specify Columns option in the Add User Defined Data Source dialog should be unchecked, otherwise exceptions will be produced when the SQL statement is used to generate dynamic report from the UDS.

Oracle stored procedure UDS

Due to the unique nature of Oracle stored procedures, you cannot add them directly into a JReport catalog. In this sense, the usage of JReport Designer is limited when you use an Oracle database. As a substitute, JReport has developed the user data source API which can use stored procedures in Oracle. The class is named OracleProcedureUDS, in the package jet.datasource.oracle.

To add a stored procedure into a JReport catalog:

1. Start JReport Designer and open the catalog to which you want to add it.
2. In the Catalog Browser, right-click the catalog data source the stored procedure is to be added, then select **Add User Defined Data Source**.
3. In the Name field of the Add User Defined Data Source dialog, specify a name for the UDS.
4. In the Class Name field, input the UDS class **jet.datasource.oracle.OracleProcedureUDS**.
5. In the Parameter box, enter the parameter as required.
6. Click **OK**, and the UDS class will be added into the catalog.

Formats of the parameter string

In the Add User Defined Data Source dialog, the format of the parameter string is in either of the following. You can choose one according to your requirement. When the stored procedures used in Oracle have different names, either of the following two formats can be used. However, when the stored procedures used in Oracle have the same names, you can only use the second parameter string. In addition, when you use the second parameter string, the parameter value only contains value. You need to add the data type and index of the parameter value in the user defined data source class.

- `DRIVER=drivername&URL=url&USER=user&PSWD=password&OWNER=owner&PROCNAME=procname&SQL=sql&REFCURSORINDEX=index&PARAMVALUE=value1, value2,value3&OUTPARAMINDEX=index1,index2`
- `DRIVER=drivername&URL=url&USER=user&PSWD=password&OWNER=owner&PROCNAME=procname&SQL=sql&REFCURSORINDEX=index&INPARAMVALUE=value1, datatype1,index1;value2,datatye2,index2;…Valuen,datatypen,indexn&OUTPARAM=datatype1,index1;… datatypen,indexn`

Substitute the words in lower case according to your requirements.

Where:

- **drivername** - The name of JDBC driver, for instance: oracle.jdbc.driver.OracleDriver.
- **url** - A database URL of the form jdbc:subprotocol:subname, for instance: jdbc:oracle:thin:@dbserver:1521:orcl.
- **user** - The database user on whose behalf the Connection is being made, for instance: scott.
- **password** - The user's password, for instance: tiger.
- **owner** - The owner of the procedure, for instance: scott. It can also be left empty.
- **procname** - The name of the Stored Procedure, for instance: getAuthor.
- **sql** - The SQL statement used to call the procedure or function. For instance, with regards to procedure: {call getAuthor(?, ?)} and for function: {? = getAuthor(?, ?)}.
- **index** - The index of the output parameter with data type equal to Ref Cursor. The start index is 1.
- **valueX** - The value of the parameters separated by ',' and the data types must correspond to the ones defined in the DBMS. If valueX should contain a set of constant string values, for example, 10, 20, and 30 together serve as valueX, you will need to use '\\' to escape the comma.

  For example, if the PARAMVALUE part of the parameter string is as follows:
  …&PARAMVALUE=10\\,20\\,30,30\\,10\\,23,,a,

  Then the parameter values will be parsed as follows:
  ParaValue1 =10,20,30
  ParaValue2 =30,10,23
  ParaValue3 =null
  ParaValue4 =a
  ParaValue5 =null

  Or, instead of typing the parameter values one by one in the above parameter string, you can also use the parameters and constant level formulas in the current catalog data source to substitute the parameter values, then '\\' will automatically be added to escape the comma. And it is recommended to use the parameters and constant level formulas to control one input parameter value of the oracle stored procedure. For example,
  …&PARAMVALUE=@PARAM1,30\\,10\\,23,,a,

  In this case, in the Create/Edit Parameter dialog, the prompt value of parameter PARAM1 must be typed as follows:
  10,20,30
- **datatypeX** - The name of the data type mapping to JDBC. The name is the same as the static variable name in java.sql.types.
- **IndexX** - The index of the input parameter. It is natural number and integer.

Below is an example of the parameter string:

```
DRIVER=oracle.jdbc.driver.OracleDriver&URL="jdbc:oracle:thin:@dbserver:1521:orcl"
&USER=scott&PSWD=tiger&OWNER=SCOTT&PROCNAME=getAuthor&SQL={call getAuthor(?, ?, ?}&REFCURSORINDEX=2&PARAMVALUE=0.5,1999-7-10
```

If you want to change the connection dynamically, you can define the parameters in this way:

```
oracle.jdbc.driver.OracleDriver=@driver, jdbc:oracle:thin:@dbserver:1521:orcl=@url, scott=@user,
```

The parameter string will then be either of the following:

- `DRIVER=@driver&URL=@url&USER=@user&PSWD=tiger&OWNER=SCOTT&PROCNAME=getAuthor&SQL={call getAuthor(?, ?,?} &REFCURSORINDEX=1&PARAMVALUE=0.5,1999-7-10`
- `DRIVER=@driver&URL=@url&USER=@user&PSWD=tiger&OWNER=SCOTT&PROCNAME=getAuthor&SQL={call getAuthor(?, ?,?} &REFCURSORINDEX=1&INPARAMVALUE=0.5,number,2;1999-7-10,date,3`

You can change the parameter dynamically to suit your requirements.

Troubleshooting

If you get an "ArrayOutOfBound" error or a "Not all variables are bound" error while importing a stored procedure, you can check the below points:

- Make sure the OWNER is correct.
- If your stored procedure is defined in Package/Package Body and the PROCNAME argument does not contain package name, but the call argument contains.
- Make sure that your REFCURSORINDEX has been set correctly.

  For example, if you have defined that the Out cursor is the third argument in your stored procedure, then REFCURSORINDEX should be 3.

When you get a NegativeArraySizeException error while running a report designed with a stored procedure, you can check whether some UDS columns return a negative scale (right-click the UDS column, select Properties, and then check Scale). If yes, it means that your original table column called by the stored procedure is likely to be of a Float data type. This returns the wrong Scale for the UDS column. Follow the steps below to modify the UDS column's scale:

1. Open the Catalog Browser and click the **Options** button on the toolbar.
2. In the Category box of the Options dialog, select **Catalog**, uncheck the option **Forbid editing data object properties**, and then click the **OK** button.
3. Highlight the UDS column name that you are having the problem with and click the **Expand** button on the toolbar to show the Properties sheet. Now you can change the property values, such as SQL type, precision, and scale to the correct values. For instance, you can change the negative scale of the problematic column to the correct value.

Notes:

- Your oracle.jdbc.driver should use at least ojdbc14.jar.
- The UDS class is in a package named jet.datasource.oracle. When it is being added into the Catalog Browser, it requires classes in package oracle.jdbc.driver. So before adding it, make sure the class path contains the package oracle.jdbc.driver.
- In the parameter string, if you do not specify the connection information, the connection defined on the data source, to which the stored procedure UDS is added will be used as the default connection. In this scenario, use the third parameter string listed above, and the connection can be changed at runtime via datasource.xml, Engine API or Server API in JReport Server.
- When your parameter string contains characters such as: At sign(@), ':', double quotation mark('"'), or other strings that do not need to be parsed by JReport, you can use a pair of double quotation marks to quote them. For example, your parameter string may be:

  `jdbc:oracle:thin:@204.177.148.30:1521:orcl`

  Here, @ is a character used by the URL. If you add this parameter string into a catalog, JReport will regard 204 as the name of a parameter in the catalog. The correct form is:

  `"jdbc:oracle:thin:@204.177.148.30:1521:orcl"`
- The procedure must contain one and only one OUT parameter of the REF CURSOR type. If the procedure doesn't have such a parameter, it cannot be added into the JReport catalog since JReport cannot create a report without a ResultSet. If it contains additional OUT parameters they will be ignored.
- The default format of Date and Time data type is:

  Date: yyyy-MM-dd
  Time: hh:MM:ss
  Timestamp: yyyy-MM-dd hh:MM:ss.SSS

  You can modify the format in the Date Format tab of the Get JDBC Connection Information dialog.

The following are some specific examples of adding Oracle stored procedures UDS to a catalog:

- Example 1: Stored procedures defined outside of a package
- Example 2: Stored procedures declared and defined in a package and package body
- Example 3: Functions defined outside of a package
- Example 4: Functions declared and defined in a package and package body

Example 1: Stored procedures defined outside of a package

Suppose that an Oracle stored procedure is defined as:

```
CREATE OR REPLACE PACKAGE SHDEMO as
type curtype is ref cursor;
END SHDEMO;
CREATE OR REPLACE PROCEDURE empquery (p_job in varchar2, cur OUT SHDEMO.curtype) as
lcur SHDEMO.curtype;
begin
  open lcur for
  select * from EMP WHERE JOB = p_job;
  cur:=lcur;
END empquery;
```

To add the stored procedure into a JReport catalog, follow the steps below:

1. Start JReport Designer and open an existing catalog.
2. In the Data tab of the Catalog Browser, right-click the data source to which the stored procedure is to be added, then select **Add User Defined Data Source**.
3. In the Name field of the Add User Defined Data Source dialog, specify a name for the UDS.
4. In the Class Name field, input the UDS class **jet.datasource.oracle.OracleProcedureUDS**.
5. In the Parameters box, enter one of the following:
   - `DRIVER=oracle.jdbc.driver.OracleDriver&URL="jdbc:oracle:thin:@<hostname>:1521:oracle8"&USER=scott&PSWD=tiger&OWNER=SCOTT&PROCNAME=empquery&SQL={call empquery(?,?)}&REFCURSORINDEX=2&PARAMVALUE="SALESMAN"`
   - `DRIVER=oracle.jdbc.driver.OracleDriver&URL="jdbc:oracle:thin:@<hostname>:1521:oracle8"&USER=scott&PSWD=tiger&OWNER=SCOTT&PROCNAME=empquery&SQL={call empquery(?,?)}&REFCURSORINDEX=2&INPARAMVALUE="SALESMAN",varchar,1`
   - `OWNER=SCOTT&PROCNAME=empquery&SQL={call empquery(?,?)}&REFCURSORINDEX=2&PARAMVALUE="SALESMAN"`
6. Click the **OK** button, and the UDS class will then be added into the catalog.

Example 2: Stored procedures declared and defined in a package and package body

Suppose that an Oracle stored procedure is defined as:

```
CREATE OR REPLACE PACKAGE DEMO_PKG AS
type curtype is ref cursor;
PROCEDURE empquery_proc(p_job in varchar2, cur OUT DEMO_PKG.curtype);
end DEMO_PKG ;
CREATE OR REPLACE PACKAGE BODY DEMO_PKG AS
PROCEDURE empquery_proc(p_job in varchar2, cur OUT DEMO_PKG.curtype) AS
lcur DEMO_PKG.curtype;
begin
  open lcur for
  select * from EMP WHERE JOB = p_job;
  cur:=lcur;
end empquery_proc;
end DEMO_PKG;
```

To add the stored procedure into a JReport catalog, follow the steps below:

1. Start JReport Designer and open an existing catalog.
2. In the Data tab of the Catalog Browser, right-click the data source to which the stored procedure is to be added, then select **Add User Defined Data Source**.
3. In the Name field of the Add User Defined Data Source dialog, specify a name for the UDS.
4. In the Class Name field, input the UDS class **jet.datasource.oracle.OracleProcedureUDS**.
5. In the Parameter box, enter one of the following:
   - `DRIVER=oracle.jdbc.driver.OracleDriver&URL="jdbc:oracle:thin:@<hostname>:1521:oracle8"&USER=scott&PSWD=tiger &OWNER=SCOTT&PROCNAME=empquery_proc&SQL={call demo_pkg.empquery_proc(?,?)} &REFCURSORINDEX=2&PARAMVALUE="SALESMAN"`
   - `DRIVER=oracle.jdbc.driver.OracleDriver&URL="jdbc:oracle:thin:@<hostname>:1521:oracle8"&USER=scott&PSWD=tiger& OWNER=SCOTT&PROCNAME=empquery_proc&SQL={call demo_pkg.empquery_proc(?,?)} &REFCURSORINDEX=2&INPARAMVALUE="SALESMAN",varchar,1`
   - `OWNER=SCOTT&PROCNAME=empquery_proc&SQL={call demo_pkg.empquery_proc(?,?)} &REFCURSORINDEX=2&PARAMVALUE="SALESMAN"`
6. Click the **OK** button. The UDS class will then be added into the catalog.

Example 3: Functions defined outside of a package

Suppose that an Oracle stored procedure is defined as:

```
CREATE OR REPLACE PACKAGE SHDEMO AS
type curtype is ref cursor;
end SHDEMO;
CREATE OR REPLACE FUNCTION "SCOTT"."DEPTQUERY"(p_dname in varchar2)
return SHDEMO.curtype
is
lcur SHDEMO.curtype;
begin
  open lcur for
  select * from dept where dname = p_dname;
  return lcur;
end DEPTQUERY;
```

To add the stored procedure into a JReport catalog, follow the steps below:

1. Start JReport Designer and open an existing catalog.
2. In the Data tab of the Catalog Browser, right-click the data source to which the stored procedure is to be added, then select **Add User Defined Data Source**.
3. In the Name field of the Add User Defined Data Source dialog, specify a name for the UDS.
4. In the Class Name field, input the UDS class **jet.datasource.oracle.OracleProcedureUDS**.
5. In the Parameter box, enter one of the following:
   - `DRIVER=oracle.jdbc.driver.OracleDriver&URL="jdbc:oracle:thin:@<hostname>:1521:oracle8"&USER=scott&PSWD=tiger&OWNER=SCOTT&PROCNAME=deptquery&SQL={? = call deptquery(?)}&REFCURSORINDEX=1&PARAMVALUE=@p_dname`
   - `DRIVER=oracle.jdbc.driver.OracleDriver&URL="jdbc:oracle:thin:@<hostname>:1521:oracle8"&USER=scott&PSWD=tiger&OWNER=SCOTT&PROCNAME=deptquery&SQL={? = call deptquery(?)}&REFCURSORINDEX=1&INPARAMVALUE=@p_dname,varchar,1`
   - `OWNER=SCOTT&PROCNAME=deptquery&SQL={? = call deptquery(?)}&REFCURSORINDEX=1&PARAMVALUE=@p_dname`
6. Click the **OK** button. The UDS class will then be added into the catalog.

Example 4: Functions declared and defined in a package and package body

Suppose that an Oracle stored procedure is defined as:

```
CREATE OR REPLACE PACKAGE "SCOTT"."SHDEMO_PKG" AS
type curtype is ref cursor;
Procedure empquery_proc (p_job in varchar2, cur OUT SHDEMO_PKG.curtype);
Function deptquery_func (p_dname in varchar2) return SHDEMO_PKG.curtype;
end SHDEMO_PKG;
CREATE OR REPLACE PACKAGE BODY "SCOTT"."SHDEMO_PKG" AS
Procedure empquery_proc (p_job in varchar2, cur OUT SHDEMO_PKG.curtype)
as
lcur SHDEMO_PKG.curtype;
begin
  open lcur for
  select * from EMP WHERE JOB = p_job;
  cur:=lcur;
end empquery_proc;
Function deptquery_func (p_dname in varchar2) return SHDEMO_PKG.curtype
as
lcur SHDEMO_PKG.curtype;
begin
  open lcur for
  select * from dept where dname = p_dname;
  return lcur;
end deptquery_func;
end SHDEMO_PKG;
```

To add the stored procedure into a JReport catalog, follow the steps below:

1. Start JReport Designer and open an existing catalog.
2. In the Data tab of the Catalog Browser, right-click the data source to which the stored procedure is to be added, then select **Add User Defined Data Source**.
3. In the Name field of the Add User Defined Data Source dialog, specify a name for the UDS.
4. In the Class Name field, input the UDS class **jet.datasource.oracle.OracleProcedureUDS**.
5. In the Parameter box, enter one of the following:
   - `DRIVER=oracle.jdbc.driver.OracleDriver&URL="jdbc:oracle:thin:@<hostname>:1521:oracle8"&USER=scott&PSWD=tiger&OWNER=SCOTT&PROCNAME=deptquery_func&SQL={? = call shdemo_pkg.deptquery_func(?)}&REFCURSORINDEX=1&PARAMVALUE="SALES"`
   - `DRIVER=oracle.jdbc.driver.OracleDriver&URL="jdbc:oracle:thin:@<hostname>:1521:oracle8"&USER=scott&PSWD=tiger&OWNER=SCOTT&PROCNAME=deptquery_func&SQL={? = call shdemo_pkg.deptquery_func(?)}&REFCURSORINDEX=1&INPARAMVALUE="SALES",varchar,2`
   - `OWNER=SCOTT&PROCNAME=deptquery_func&SQL={? = call shdemo_pkg.deptquery_func(?)}&REFCURSORINDEX=1&PARAMVALUE="SALES"`
6. Click the **OK** button. The UDS class will then be added into the catalog.

EnterpriseDB stored procedure UDS

Due to the unique nature of EnterpriseDB stored procedures, you cannot add them directly into a JReport catalog. In this sense, the usage of JReport Designer is limited when you use an EnterpriseDB database. As a substitute, JReport has developed the user data source API which can use stored procedures in EnterpriseDB. The usage of EnterpriseDB stored procedures is the same as that of Oracle stored procedures, except that the class name for EnterpriseDB stored procedures is EnterpriseDbProcedureUDS, in the package jet.datasource.enterprisedb.

Example of adding an EnterpriseDB stored procedure to a catalog

Suppose that an EnterpriseDB stored procedure is defined as follows:

```
create or replace package shdemo
as
type curtype
is
ref
cursor;
end shdemo;

create or replace procedure empquery (param1 in varchar2,
cur out shdemo.curtype) as lcur shdemo.curtype;
begin
open lcur for
select
*
from
emp
where
job = param1;
cur:=lcur;
end empquery;
```

To add the stored procedure into a JReport catalog, follow the steps below:

1. Start JReport Designer and open the catalog to which you want to add the stored procedure.
2. In the Catalog Browser, right-click the data source the stored procedure is to be added, then select **Add User Defined Data Source**.
3. In the Name field of the Add User Defined Data Source dialog, specify a name for the UDS.
4. In the Class Name field, input the UDS class **jet.datasource.enterprisedb. EnterpriseDbProcedureUDS**.
5. In the Parameter box, enter any of the following three:
   - `DRIVER=com.edb.Driver&URL="jdbc:edb://222.222.222.45:5444/edb"&USER=enterprisedb&PSWD=db1234&OWNER=enterprisedb&PROCNAME=empquery&SQL={call empquery(?,?)}&REFCURSORINDEX=2&PARAMVALUE=@job1`
   - `DRIVER=com.edb.Driver&URL="jdbc:edb://222.222.222.45:5444/edb"&USER=enterprisedb&PSWD=db1234&OWNER=enterprisedb&PROCNAME=empquery&SQL={call empquery(?,?)}&REFCURSORINDEX=2&INPARAMVALUE=@job1,varchar,1`
   - `OWNER=enterprisedb&PROCNAME=empquery&SQL={call empquery(?,?)} &REFCURSORINDEX=2&PARAMVALUE=@job1`
6. Click **OK**, and the UDS class will be added into the catalog.

Hierarchical data sources

JReport Designer supports hierarchical data sources (HDS). This feature uses the hierarchical data source API provided by JReport.

In addition, as the XML technology has become a popular standard for data interchange on the web, JReport Designer directly supports XML format data source by wrapping the provided HDS API. JReport Designer's built-in classes can implement the XML format hierarchical data source interface. You can directly import an XML data source to a catalog using the Catalog Browser.

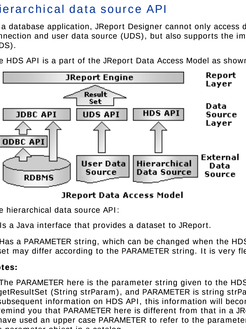
The following are some specific topics about working with HDSs in a catalog:

- hierarchical data source API
- Adding an HDS to a catalog
- HDS with a parallel structure
- Developing reports from HDS

Hierarchical data source API

As a database application, JReport Designer cannot only access data stored in a database through the JDBC connection and user data source (UDS), but also supports the importing of data from hierarchical data sources (HDS).

The HDS API is a part of the JReport Data Access Model as shown in the following diagram:



JReport Data Access Model

The hierarchical data source API:

- Is a Java interface that provides a dataset to JReport.
- Has a PARAMETER string, which can be changed when the HDS object is added into JReport Designer. The result set may differ according to the PARAMETER string. It is very flexible and convenient.

Notes:

- The PARAMETER here is the parameter string given to the HDS interface. That is, the HDS API provides a method getResultSet (String strParam), and PARAMETER is string strParam in the above method. After reading the subsequent information on HDS API, this information will become clearer to you. As a note, we would like to remind you that PARAMETER here is different from that in a JReport catalog. To distinguish between them, we have used an upper case PARAMETER to refer to the parameter string in the HDS, and a lower case parameter for a parameter object in a catalog.
- To design a report, JReport requires meta data (See java.sql.ResultSetMetaData) which corresponds with the ResultSet that the data source has returned. To view or run a report, JReport requires a ResultSet object. The hierarchical data source must provide both of these two parts to JReport.

HDS API interfaces

The HDS API contains three interfaces as listed below that you must implement.

- jet.datasource.JRHierarchicalDataSource
- jet.datasource.JRHierarchicalDataset
- jet.datasource.JRHierarchicalDatasetMetaData

Reference: JReport Javadoc jet.datasource.JRHierarchicalDataSource, jet.datasource.JRHierarchicalDataset and jet.datasource.JRHierarchicalDatasetMetaData classes in `<install_root>\help\api`.

jet.datasource.JRHierarchicalDataSource

Provides data to JReport for generating reports. The JRHierarchicalDataSource class is developed by users of JReport, and can provide data from a flat file, non-relational database, or application data. The data returned by this class is in JRHierarchicalDataset object, so that users are required to create a JRHierarchicalDataset instance for JReport to use this instance to fetch data. Users can also create their own JRHierarchicalDataset class.

The following are methods provided by the interface.

public JRHierarchicalDataset getHierarchicalDataset (String param) throws JRUserDataSourceException

This method gets the data in JRHierarchicalDataset according to the parameters set.

- **PARAMETER**: param - A String value used to request and get different result sets. The format of the PARAMETER string is defined and also parsed by yourself.
- **Returns**: JRHierarchicalDataset object, which implements the interface JRHierarchicalDataset.
- **Throws**: JRUserDataSourceException - If a data access error occurs.

public void releaseHierarchicalDataset() throws JRUserDataSourceException

This method releases the data and related resources.

- **Throws**: JRUserDataSourceException - If a data access error occurs.

The class definition may be as follows:

```
 public class HierarchicalDataSource implements JRHierarchicalDataSource {
        // Define data.
        public JRHierarchicalDataset getHierarchicalDataset(String param)
                        throws JRUserDataSourceException {
                // Method body.
        }

        public void releaseHierarchicalDataset() throws JRUserDataSourceException {
                // Method body.
        }
}
```

jet.datasource.JRHierarchicalDataset

Provides data to JReport for generating reports. The JRHierarchicalDataset class is developed by users of JReport, and can provide data from a flat file, non-relational database, or application data.

The following are methods provided by the interface.

```
 public interface JRHierarchicalDataset
 {
     /**
     * Retrieves the number, types and properties of this object's leaves
     */
    public JRHierarchicalDatasetMetaData getMetaData();
     /**
     * Moves the cursor down one row from its current position in the current node.
     * return true if the new current row is valid; false if there are no more rows
     * in the current node.
     */
    public boolean next(String branchName);
     /**
     * Gets the value of the designated leaf node in the current branch node row as a boolean
     * in the Java programming language.
     */
    public boolean getBoolean(int leafIndex);
     /**
     * Gets the value of the designated leaf in the current branch node row as a byte
     * in the Java programming language.
     */
    public byte getByte(int leafIndex);
     /**
     * Returns the value of the designated leaf in the current branch node row as a short
     * in the Java programming language.
     */
    public short getShort(int leafIndex);
     /**
     * Returns the value of the designated leaf in the current branch node row as an int
     * in the Java programming language.
     */
    public int getInt(int leafIndex);
     /**
     * Returns the value of the designated leaf in the current branch node row as a long
     * in the Java programming language.
     */
    public long getLong(int leafIndex);
     /**
     * Returns the value of the designated leaf in the current branch node row as a float
     * in the Java programming language.
     */
    public float getFloat(int leafIndex);
     /**
     * Returns the value of the designated leaf in the current branch node row as a double
     * in the Java programming language.
     */
    public double getDouble(int leafIndex);
     /**
     * Gets the value of the designated leaf in the current branch node row as a
     * java.math.BigDecimal with full precision.
     */
    public java.math.BigDecimal getBigDecimal(int leafIndex);
     /**
     * Returns the value of the designated leaf in the current branch node row as a
     * java.sql.Date object
     * in the Java programming language.
     */
    public java.sql.Date getDate(int leafIndex);
     /**
     * Returns the value of the designated leaf in the current branch node row as a
     * java.sql.Time object
     * in the Java programming language.
     */
    public java.sql.Time getTime(int leafIndex);
     /**
     * Returns the value of the designated leaf in the current branch node row as
     * a java.sql.Timestamp object
     * in the Java programming language.
     */
    public java.sql.Timestamp getTimestamp(int leafIndex);
     /**
     * Returns the value of the designated leaf in the current branch node row as a String
     * in the Java programming language.
     */
    public String getString(int leafIndex);
     /**
     * Returns the value of the designated leaf in the current branch node row as an Array object
     * in the Java programming language.
     */
    public java.sql.Array getArray(int leafIndex);
     /**
     * Reports whether the last leaf read had a value of SQL NULL
     */
    public boolean wasNull(int leafIndex);
     /**
     * Releases this object's resource.
     */
    public void close();
 }
```

jet.datasource.JRHierarchicalDatasetMetaData

Provides meta data for JRHierarchicalDataset. The JRHierarchicalDatasetMetaData class is developed by users of JReport, and it works together with a JRHierarchicalDataset.

The following are methods provided by the interface.

```
 public interface JRHierarchicalDatasetMetaData
 {
     /**
     * Return root node names
     */
    public String[] getRoot() ;
     /**
     * Return the name of the parent.
     */
    public String getParentName(String name)
     /**
     * Return all leaf node names.
     */
    public String[] getLeafNames(String parentName);
     /**
     * Return the number of leaves of the specified branch node.
     */
    public int getLeafCount(String parentName);
     /**
     * Return all branch names for the specified parent node.
     */
    public String[] getBranchNames(String parentName);
     /**
     * Return the data type of the leaf.
     */
    public int getLeafType(String parentName, String leafName);// throws SQLException;
     /**
     * Return the name of data type.
     */
    public String getLeafTypeName(String parentName, String leafName);
     /**
     * Return the precision of the leaf.
     */
    public int getPrecision(String parentName, String leafName);
     /**
     * Return the scale of the leaf.
     */
    public int getScale(String parentName, String leafName);
     /**
     * Return the nullable state of the leaf.
     */
    public int isNullable(String parentName, String leafName);
     /**
     * Return the currency state of the leaf.
     */
    public boolean isCurrency(String parentName, String leafName);
 }
```

Examples of using the HDS API

The HDS API is flexible and convenient to use. Before you implement it, you should make an overall consideration of the architecture. JReport provides you with several scenarios which use the JReport HDS API. You can refer to them for assistance.

- You can import an XML format hierarchical data source directly from an external data source to JReport Designer using the JReport built-in classes. See Example 2: Importing an HDS from an XML file to a catalog.
- You can also develop your own classes to implement the HDS APIs, and then import your customized hierarchical data source. See Example 1: Adding a general HDS to a catalog.

Adding an HDS to a catalog

You have two ways to add an HDS to a catalog:

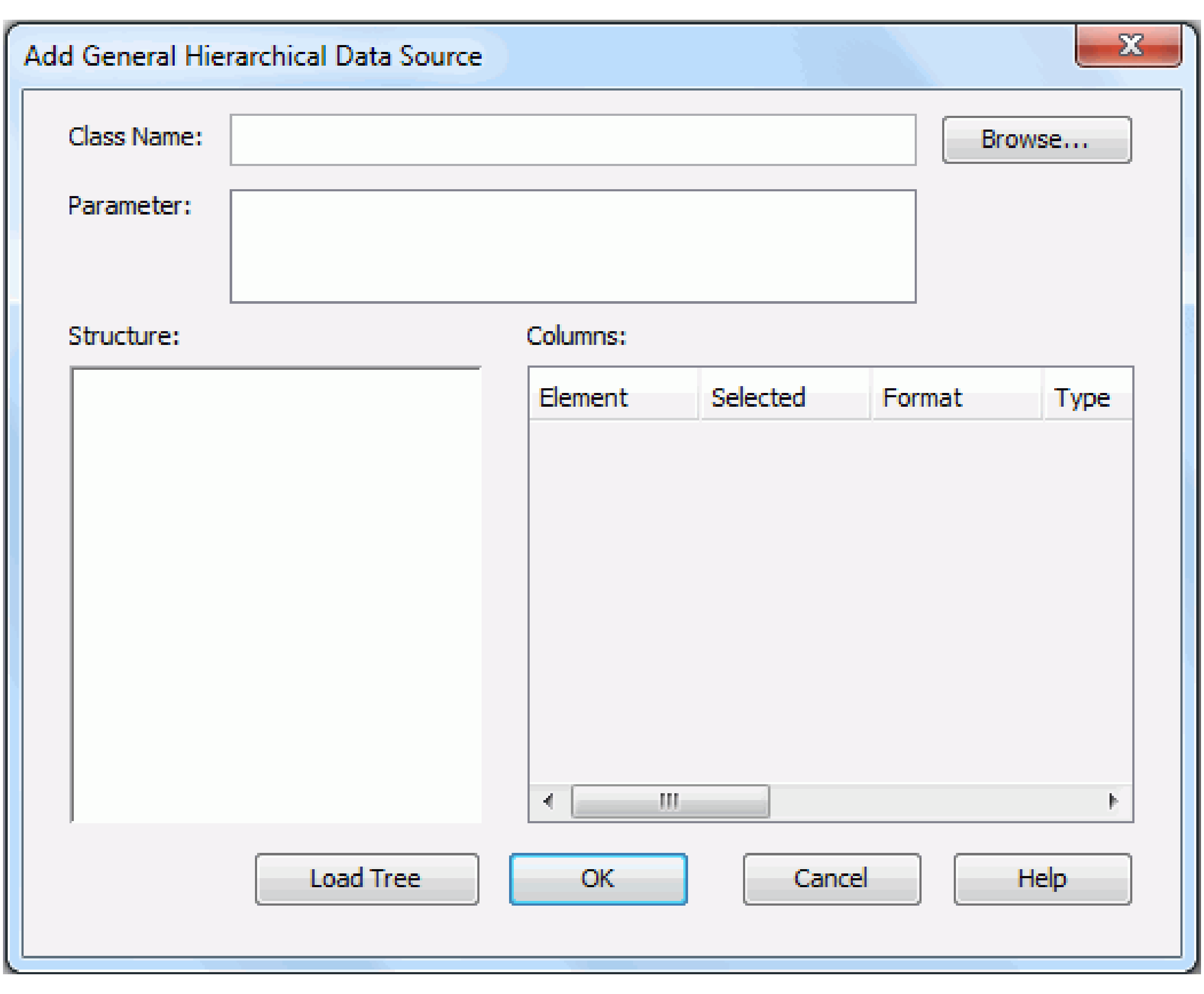
- **Adding a general HDS**
  1. Open the catalog to which you want to add the HDS.
  2. In the Data tab of the Catalog Browser, right-click the data source to which the HDS is to be added, then select **Add General Hierarchical Data Source** from the shortcut menu.
  3. In the Add General Hierarchical Data Source dialog, click the **Browse** button to specify the HDS class name, and type a number in the Parameter box.
  4. Click **Load Tree**. The data source tree will then be parsed. Modify the column properties in the Columns box as required.
  5. Click **OK** to add the HDS.
- **Importing an HDS from an XML file**
  1. Select **Add XML Hierarchical Data Source** from the shortcut menu after right-clicking the data source node.
  2. In the Import XML Hierarchical Data Source dialog, click the **Browse** buttons to specify the XML URI and XSD URI as required.
  3. Click the **Load Tree** button to load the structure of the XML file. The root of the file will then be listed in the Root Name text box. Modify the column properties in the Columns box as required.
  4. Click **OK** to import the HDS into the catalog.

The following examples explain the above two methods in detail.

Example 1: Adding a general HDS

There are three classes used in this example. Their source code files are HierarchicalDataSource.java, HierarchicalDatasetMetaData.java, and HierarchicalDataset.java, which are available in `<install_root>\help\samples\APIUDS\hierarchicalUDS`. In this example, the HierarchicalDataSource.java will return the result set from the demo HSQL DB.

1. Copy the above three files to `<install_root>\help`. Modify the demo HSQL DB path in both HierarchicalDatasetMetaData.java and HierarchicalDataset.java.
2. Compile the Java files to generate HierarchicalDataSource.class, HierarchicalDatasetMetaData.class, and HierarchicalDataset.class.
3. Append the path `<install_root>\help` into the ADDCLASSPATH variable of the batch file setenv.bat in `<install_root>\bin`, so that at runtime HierarchicalDataSource can be found.
4. Start JReport Designer with the batch file you just modified and open an existing catalog.
5. In the Data tab of the Catalog Browser, right-click the data source to which to add the HDS, then select **Add General Hierarchical Data Source** from the shortcut menu. The Add General Hierarchical Data Source dialog appears.

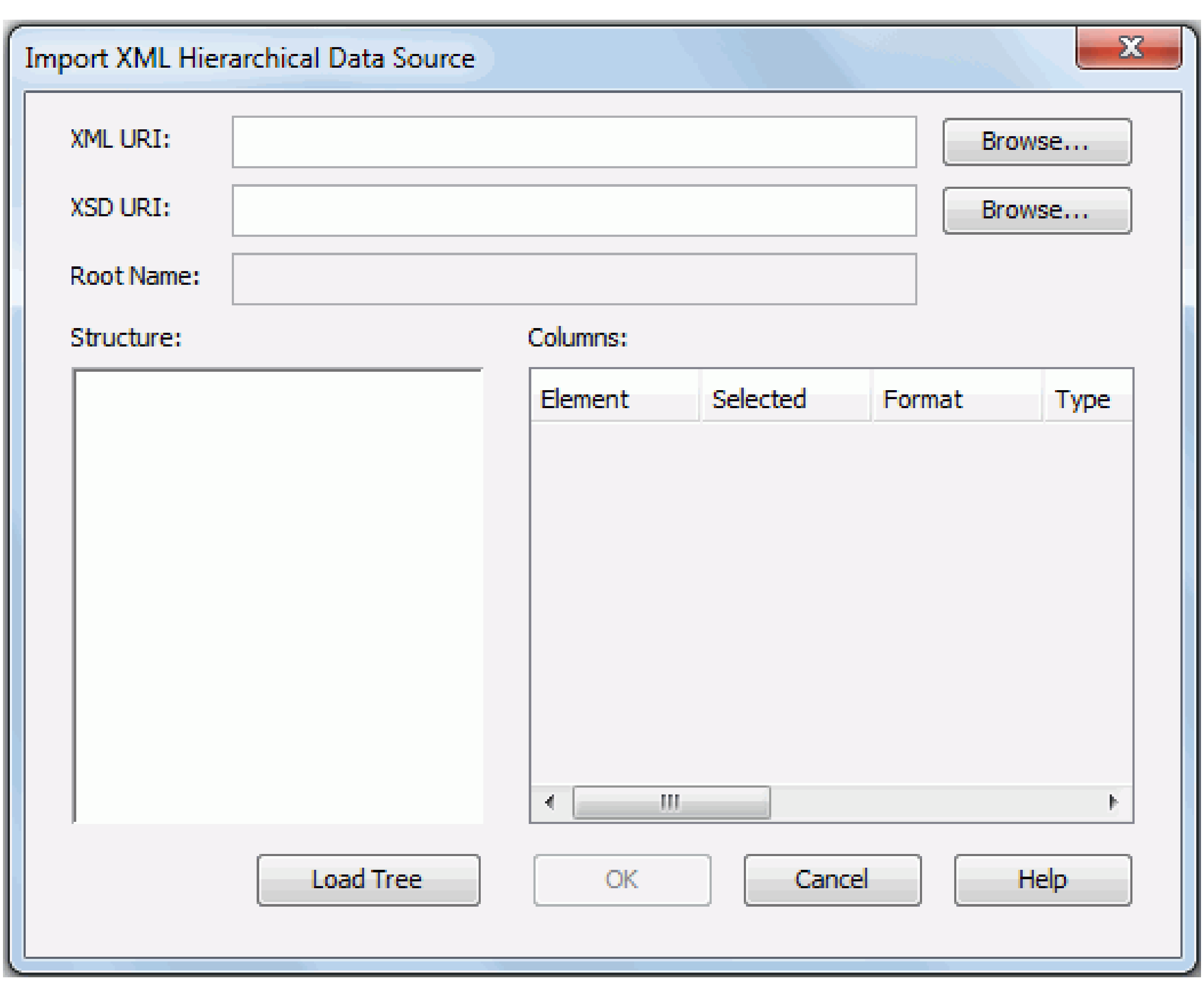


6. Click the **Browse** button to find the class HierarchicalDataSource.class.
7. Type a number in the Parameter box. Then, only the records whose employee ID is less than this number will be fetched.
8. Click **Load Tree**. The data source tree will then be parsed.
9. Modify the column properties in the Columns box as required.
10. Click **OK** to add the HDS.

Example 2: Importing an HDS from an XML file to a catalog

To import an HDS from an XML file, follow the steps below:

1. Start JReport Designer and open an existing catalog.
2. In the Data tab of the Catalog Browser, right-click the data source to which to add the HDS, and then select **Add XML Hierarchical Data Source** from the shortcut menu. The Import XML Hierarchical Data Source dialog appears.



3. Provide the information respectively according to your data source information.

   JReport Designer supports all kinds of URI as the XML data source.

   - If you want to specify a schema file for the XML data source file, you must first make sure that the schema file path specified in the XML data source file is consistent with the path you specified in the XSD URI field, and that this file actually exists. Click the **Load Tree** button to load the structure of the XML file. The root of the file will then be listed in the Root Name field. Modify the column properties in the Columns box as required.
   - JReport Designer supports static string inline with multiple parameters, for example:

     `"http://localhost:8888/jrserver%2fSampleReports%2fSampleReports.cat/StockMarket.cls?jrs.cmd=jrs.try_vw&jrs.result_type=7&jrs.param$p_Year=2007&jrs.param$p_Month=2"`

     It is the URL used to run the sample report StockMarket.cls to the XML format in JReport Server. Type the URL into the XML URI text field, and click the **Load Tree** button, the structure of the returned XML stream will be loaded in the Structure box (before doing so, make sure that JReport Server is started). Modify the column properties in the Columns box as required.

     The colon ':' and '@' symbols are used to identify JReport parameter names. If these symbols are used in your XML URI and you do not want JReport to parse them as parameters, you must add quotation marks to them. For example, when you browse to `d:\test\employee.xml`, you can quote it either as `"d:\test\employee.xml"` or `d":"\test\employee.xml`.

     Note that you should check the No Security Checking option in the Advanced panel in the JReport Server Administration > Configuration page before parsing this URL. In addition, JReport parameters can be dynamically referenced in the URI for setting different values at runtime.

   When importing the XML file, you have to define the types of some data in the Format column. For example, if it is Date type, such as 1978-03-12, in the corresponding Format column, type in yyyy-MM-dd. If the data is $12,345.32, in the corresponding Format column, type in $##,###.##. By default, the value of the Scale column is 0, therefore, for decimal type data, you will have to specify the scale value in its corresponding Scale column, that is, modify this value to the number of digits that you want to appear to the right of the decimal point, for example, if the data is 123.23, then in the Scale column, modify this value to 2. For the Currency and Array columns, check them if required.
4. Click **OK** to import the HDS.

The XSD file

When you import an XML format HDS with an XSD file, the XML file only provides the data to the JReport Designer reports, while the structure, data type, and so on of the data from the XML file is defined in the XSD file. That is, the structure of the HDS is determined by the XSD file. You should be aware of the following points about the XSD file in order to generate a correct report based on an XML format HDS with an XSD file.

Data type conversion table

Before the data type defined in the XSD file can function with JReport Designer, it should first be converted into a corresponding data type when the XML format hierarchical data source is imported, following the rules in the conversion table below.

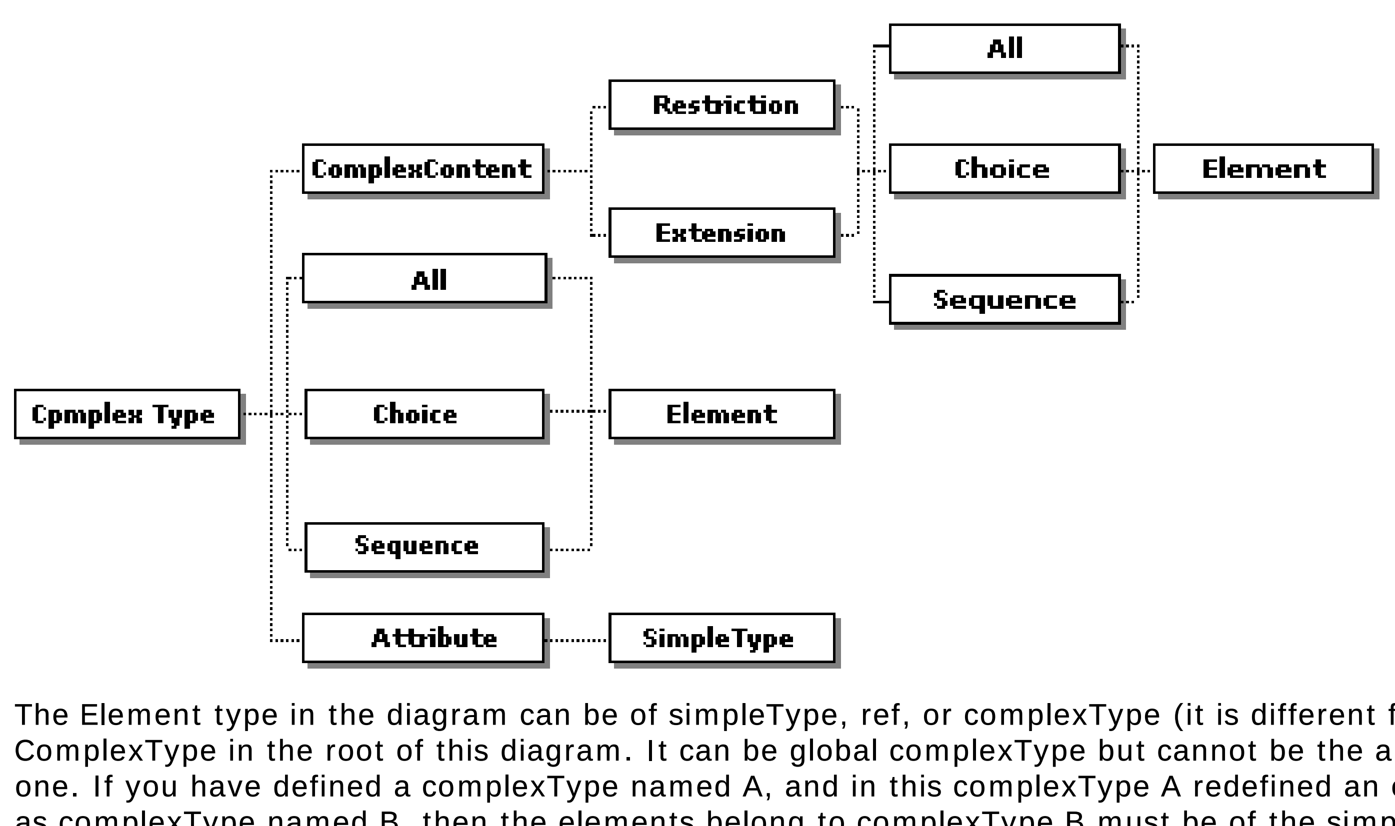
XML Data Type	JReport Data Type
SchemaSymbols.ATTVAL_BOOLEAN	java.sql.Types.BIT
SchemaSymbols.ATTVAL_INT	java.sql.Types.INTEGER
SchemaSymbols.ATTVAL_SHORT	java.sql.Types.SMALLINT
SchemaSymbols.ATTVAL_BYTE	java.sql.Types.TINYINT
SchemaSymbols.ATTVAL_INTEGER	java.sql.Types.INTEGER
SchemaSymbols.ATTVAL_NONPOSITIVEINTEGER	java.sql.Types.INTEGER
SchemaSymbols.ATTVAL_NEGATIVEINTEGER	java.sql.Types.INTEGER
SchemaSymbols.ATTVAL_NONNEGATIVEINTEGER	java.sql.Types.INTEGER
SchemaSymbols.ATTVAL_UNSIGNEDLONG	java.sql.Types.BIGINT
SchemaSymbols.ATTVAL_LONG	java.sql.Types.BIGINT
SchemaSymbols.ATTVAL_UNSIGNEDINT, //4294967295	java.sql.Types.BIGINT
SchemaSymbols.ATTVAL_UNSIGNEDSHORT, //65535	java.sql.Types.INTEGER
SchemaSymbols.ATTVAL_UNSIGNEDBYTE, //255	java.sql.Types.SMALLINT
SchemaSymbols.ATTVAL_POSITIVEINTEGER	java.sql.Types.INTEGER
SchemaSymbols.ATTVAL_FLOAT	java.sql.Types.FLOAT
SchemaSymbols.ATTVAL_DOUBLE	java.sql.Types.DOUBLE
SchemaSymbols.ATTVAL_DECIMAL	java.sql.Types.DECIMAL
SchemaSymbols.ATTVAL_STRING	java.sql.Types.VARCHAR
SchemaSymbols.ATTVAL_DATE	java.sql.Types.DATE
SchemaSymbols.ATTVAL_TIME	java.sql.Types.TIME
SchemaSymbols.ATTVAL_DATETIME	java.sql.Types.TIMESTAMP
SchemaSymbols.ATTVAL_HEXBINARY	java.sql.Types.LONGVARBINARY

Note:

XSD structure supported by JReport Designer

In JReport Designer, not all XSD structures can be supported. The following diagrams show which structures are supported by JReport Designer.

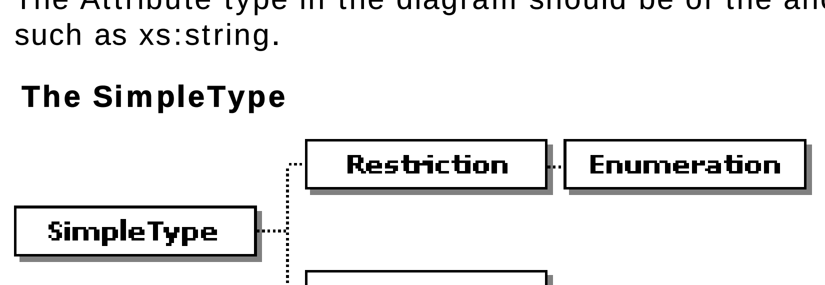
- **The ComplexType**



  The Element type in the diagram can be of simpleType, ref, or complexType (it is different from the ComplexType in the root of this diagram. It can be global complexType but cannot be the anonymous one. If you have defined a complexType named A, and in this complexType A redefined an element as complexType named B, then the elements belong to complexType B must be of the simpleType).

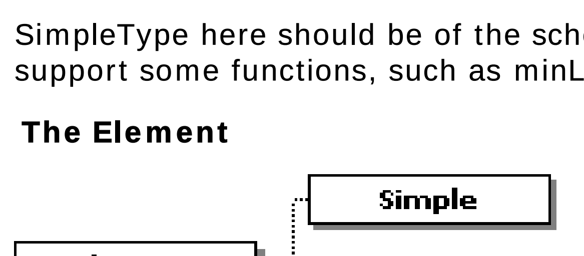
  The Attribute type in the diagram should be of the anonymous type or of the schema built-in type, such as xs:string.

- **The SimpleType**



  SimpleType here should be of the schema built-in type, such as xs:string. List type here cannot support some functions, such as minLength, and maxLength.

- **The Element**



  ComplexType here can include both global and anonymous complexType. The anonymous type means you do not give a name to the type, for example:

```
<xs:element name="aa">
  <xs:simpleType>
    <xs:restriction base="xs:string">
      <xs:enumeration value="Julie P. Adams"/>
    </xs:restriction>
  </xs:simpleType>
</xs:element>
```

  From the code above, you can see that the simpleType in the element aa has no name specified.

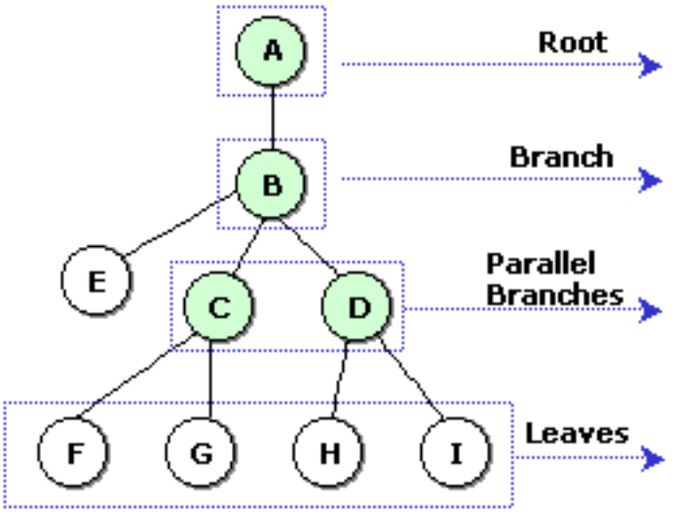
Notes:

- An XML data source file without any schema file is also allowed. That is, you can leave the XSD URI entry empty. However, the data type VARCHAR will be used for all the columns in the data source.
- When you import data of List type from an XSD file, you can define the delimiter through the property List Delimiter in the Report Inspector.
- JReport Designer also supports dynamic XML URI. The XSD file defines the structure of the XML format HDS and the XML file in fact only provides the data, so one XSD file can match more than one XML file. That enables you to develop reports with dynamic XML URI. For details, see Example 1: Developing a report from an HDS with dynamic XML URI.
- When specifying a schema file for the XML data source file, due to the schema file being complex, there are some limitations:
  - **Namespace**: Now, default namespace (3w) and target namespace are supported by JReport Designer. You can define a prefix for the default namespace, such as xs, or xsd, but the value of elementFormDefault must be qualified. The value of attributeFormDefault should be unqualified, but JReport Designer will not allow you to add prefixes before the attributes of the elements.
  - **Type**: If a customized complexType or simpleType is used, do not add the prefix to the value of the type, while if the built-in simpleType or complexType is used, the prefix must be added.

HDS with a parallel structure

Hierarchical data sources with a parallel structure are also supported by JReport Designer, and are hence called parallel HDS. The difference between the general HDS and the parallel HDS is that the parallel HDS contains more than one parallel branch, which will be presented in different Parallel Detail (PDT) panels in a banded object in JReport Designer.

Data in a parallel HDS is organized in a tree structure, which contains branches and leaves. Among these branches, there are some parallel branch nodes that split the trunk of the data source tree into deeper and more complex branches. The following is a figure illustrating the parallel HDS structure.



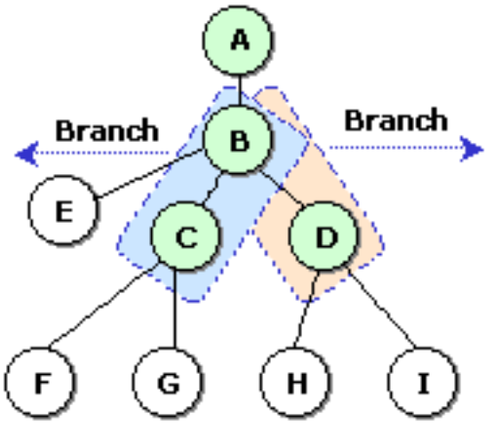
Note: Each node represents a table or a column(DBField). Nodes that contain no sub-nodes are columns,while nodes that hold one or more sub-nodes are tables.

Leaf

A leaf in parallel HDS refers to a column (DBField), nodes that contain no sub-nodes.

Branch

A branch in parallel HDS is a special table that contains leaves or sub-branches.



Parallel branches

If the contents of two branches have no parental relationships, these branches are called parallel branches.

Note: The way of using a parallel HDS is the same as with general HDS in JReport Designer. There are however, some key points that you should be aware of when you use a parallel HDS to develop reports. For details, see Example 2: Developing a report from a parallel HDS.

Developing reports from HDS

When you use hierarchical data sources to create reports, you need to be aware of its limitations and unique features. For example, reports created from an HDS are automatically grouped, and an HDS cannot be directly used to create charts and crosstabs. That is to say, if you want to create a chart or crosstab from an HDS, you will need to put it in a banded object, which is created on the HDS, and make the chart/crosstab inherit the dataset of the banded object.

To develop a report from an HDS:

1. Make sure the catalog where the HDS is located is open, then click the **New Page Report** button on the Standard toolbar.
2. In the New Page Report dialog, specify the title and layout of the first report tab in the report as required, then click **OK**.
3. In the Data screen of the report wizard, choose the HDS with which you want to build the report tab.
4. Follow the report wizard to create the report tab.

The following are two specific examples about developing reports from HDS.

Example 1: Developing a report from an HDS with dynamic XML URI

1. Click **File** > **Catalog Management** > **Open Catalog** to open an existing catalog.
2. In the Data tab of the Catalog Browser, expand the data source to which to add the HDS.
3. Right-click the **Parameters** node and select **Add Parameter** from the shortcut menu.
4. In the Create Parameter dialog, enter a name for the parameter, choose **String** as the Value Type, and in the Prompt Values box, type in the URI of the XML file from which you HDS will be imported (for details, see Creating a parameter).
5. Right-click the data source node in the Catalog Brower, and then select **Add XML Hierarchical Data Source**.
6. In the Import XML Hierarchical Data Source dialog, browse for the XML file. Then in the XML URI box, type in **@ParameterName**. For example, if you named the parameter in step 2 as XML_URI, here you should type in @XML_URI. In the XSD URI box, specify the URI of the corresponding XSD file. Then click **OK** (for details, see Importing the XML format HDS to a catalog).
7. Create a report with the imported HDS. When you view the report, type in the URI of the XML file which matches the XSD file in the parameter. The report will then display the corresponding data.

In addition, you can also use another way to develop report with dynamic XML URI as explained below:

1. In the Catalog Browser, import an XML HDS with an XSD file, and then create a report with the imported HDS.
2. Create a parameter using the above method.
3. In the Catalog Browser, go to the node of the imported HDS, expand the Properties sheet, and then change the property URI to @ParameterName.
4. Now you can view the report with dynamic XML file.

Example 2: Developing a report from a parallel HDS

The method for developing a report from a parallel HDS is the same as from a general HDS. However, take note of the following when building reports from parallel HDSs.

Groups in parallel HDS

When you insert a group by field, its children will also be inserted.

The parallel structure in the parallel HDS is represented as groups in JReport Designer. So, when you develop a report from a parallel HDS, it will automatically be grouped. By default, JReport Designer will use the first found field as the group by field. Note that you can neither add a new group by field nor remove the group by fields that are automatically added. However, if there is more than one field available to with which to group the sub-tables, you can replace the group by field with one of the other available fields.

Rules for inserting an object

- Objects in different branches should be placed in the banded object panel they belong to. That is, a banded object panel cannot hold objects which belong to different branches.
- The children of a branch, including the columns, formulas, summaries, sub-branches, and children of the sub-branches, can be inserted into any of their parent panels. If the parent panel is shared by more than one branch, that is, if the parent panel works as a trunk (global panel), it can be shared by the children of all the branches it holds. However, note that each instance of a panel, such as a report/page/group header/footer panel, and detail panel, can only hold the children from the same branch. If you want to insert the children of different branches into the shared global panel, you will have to create as many instances of this panel as per the number of branches, and then insert the children of each branch into these panels.
- Where a formula/summary can be inserted depends on the branch attributes of the formula/summary. Generally speaking, when a formula/summary creates relationships with a column, it will obtain the same branch attributes as those of the related column, and it can be placed in the panels where the column can be placed.

  A global formula, which is not associated with any branch columns, can be placed to any position of the report.

  - When you remove a branch using the report wizard, the related panel will also be removed.
  - A chart based on a field can only be placed in its own panels, which include most of the global panels, except for the detail panel.
  - A crosstab must be built upon data of the same branch, and it should be placed in the location where the branch data resides.
  - Where a text box object can be inserted depends on the branch attributes of the columns (formulas/summaries) that it contains. A text box object can hold only the data of the same branch.

Notes:

- As hierarchical data sources can only be used to create banded objects, in the New Page Report dialog, you can only choose to create a standard banded, horizontal banded or mailing label report using the report wizard.
- Since reports created from HDS are automatically grouped, in the Group screen of the banded wizard, you can neither add more groups by fields nor remove the existing ones. However, you can make changes to the existing group criteria. For example, you can replace a group by field with another one.
- When you want to publish reports created from XML format hierarchical data sources to JReport Server, the following method is recommended:

  1. In the Catalog Browser, select the HDS that is used by the report from the **Hierarchical** node, and then expand the Properties sheet.
  2. Change the URI property of the HDS to a relative path, for example, leaving only the name of the XML file.
  3. Create a new folder, and copy the report, the catalog, the XML file and the XSD file (if any) to the newly created folder.
  4. Click **File** > **Publish and Download** > **Publish to Server** to publish the folder (for details, see Publishing resources remotely).
  5. Start JReport Server. You will now be able to run the report on it.

Web service data sources

Web services provide a standard means of interoperating between different software applications, running on a variety of platforms and/or frameworks. It is a way to connect to a remote application and request data directly from the application without going directly to a DBMS. JReport Designer supports SOAP web services defined by WSDL 1.1 or WSDL 2.0 as data sources. You can add a SOAP web service data source to a catalog by importing a WSDL file.

This section describes the following tasks:

- Setting up a web service connection
- Adding tables to a web service connection
- Editing the operation used to create tables
- Example: Developing reports from web service data source

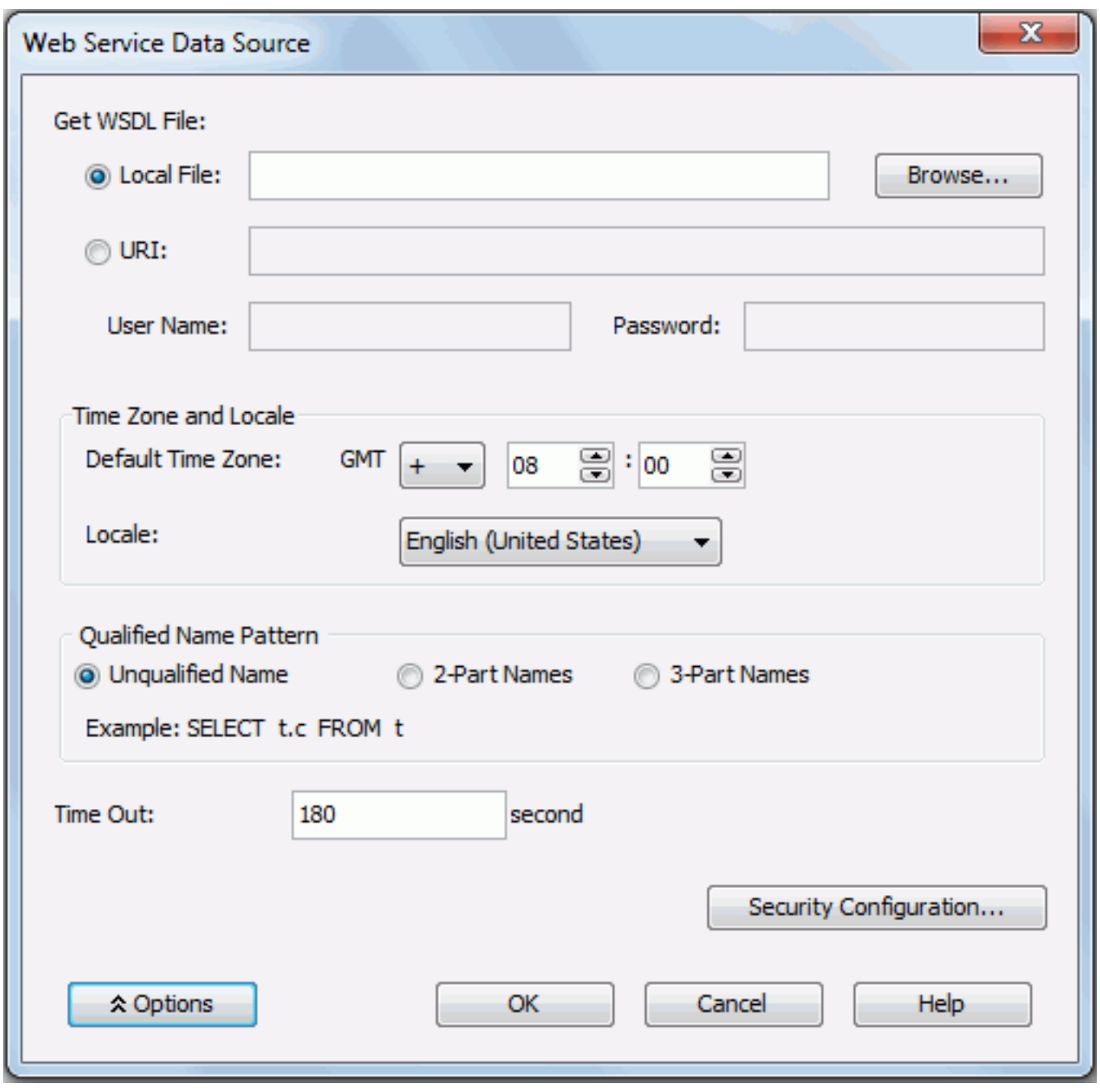
Setting up a web service connection

To set up a web service connection to connect a JReport catalog to a web service data source, follow the steps below:

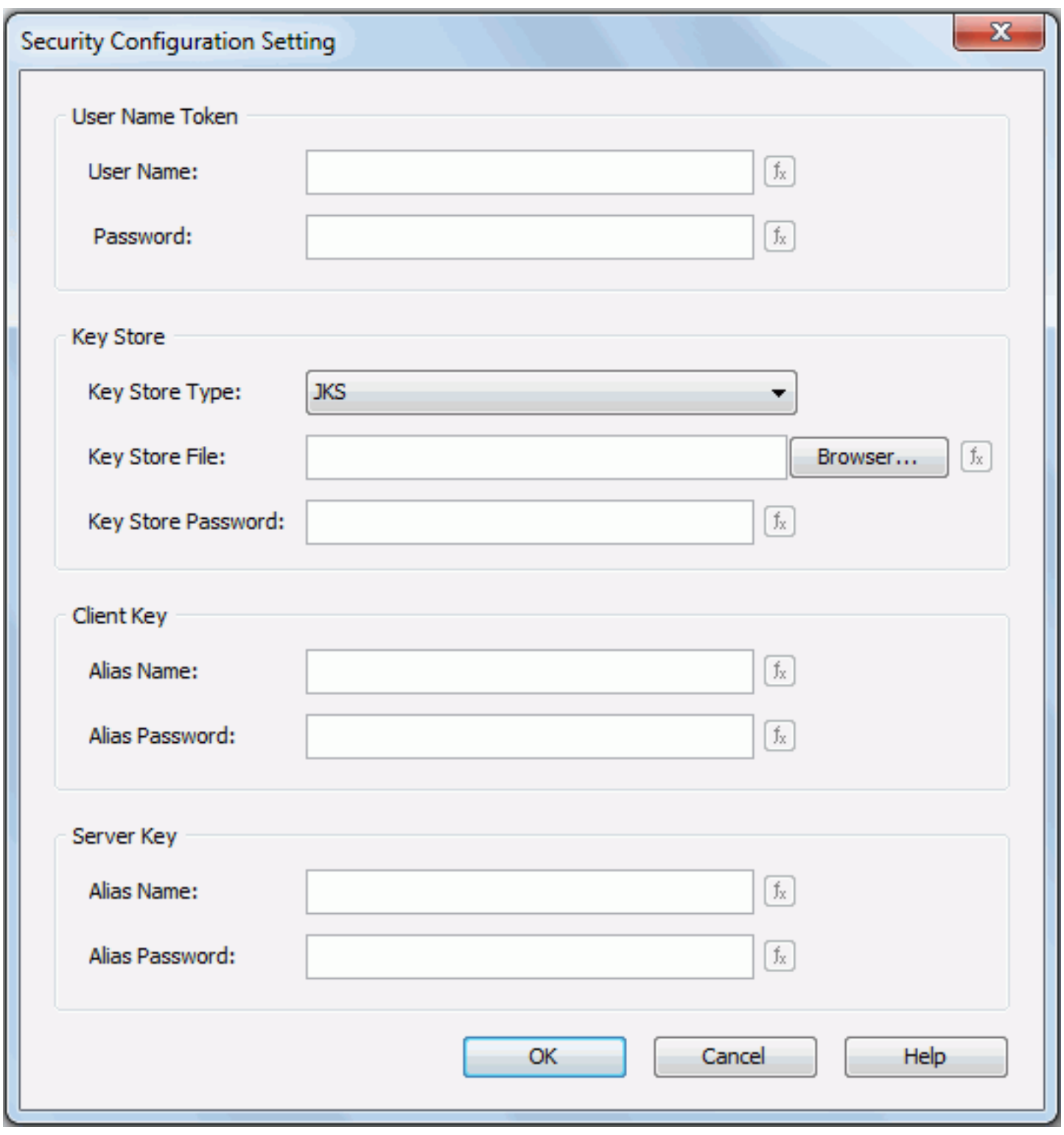
1. Create a catalog or open a catalog, then in the Data tab of the Catalog Browser, right-click the node of a data source and choose **Add SOAP Web Services Connection** from the shortcut menu.

   If you want to set up the connection by adding a data source to an existing catalog and then defining the connection on the data source, open the catalog, select any data source contained in it, click the **New Data Source** button on the Catalog Browser toolbar, then in the New Data source dialog, specify the name of the data source, select the **SOAP Web Service** connection type and click **OK**.

   The Web Service Data Source dialog appears.



2. Specify a WSDL file to create the web service connection.
   - If you want to use a WSDL file on your local disk, check the **Local File** radio button, then click the **Browse** button to browse for the file.
   - If you want to use a WSDL file through a URI, check the **URI** radio button, then input the URI string in the text field that follows and specify the user name and password for assessing the WSDL file.
3. Click the **Options** button to show the additional settings.
4. Click the **Time Zone and Locale** button to specify the time zone and the locale as required.
5. If you want the catalog or schema, or both of them to be used in data manipulation, make the selection in the Qualified Name Pattern box according to your requirements.
6. Input the number of seconds in the Time Out text field to specify how long to wait to get the WSDL file.
7. Click the **Security Configuration** button to display the Security Configuration Setting dialog to configure the security policy.



8. Specify the user name and password for the user name token to be used in the security policy.
9. Specify the type for the key store from the Key Store Type drop-down list.
10. Click the **Browse** button next to the Key Store File field to specify the key store file.
11. Input the password in the Key Store Password text field to get the access to the key store file.
12. In the Client Key box, specify the alias name to be used as client signature in the key store, and set the password for the name.
13. In the Server Key box, specify the alias name and password to be used to get the server-side certification or public key in the key store.
14. When done, click **OK** to go back to the Web Service Data Source dialog.
15. Click the **OK** button in the Web Service Data Source dialog to set up the web service connection.

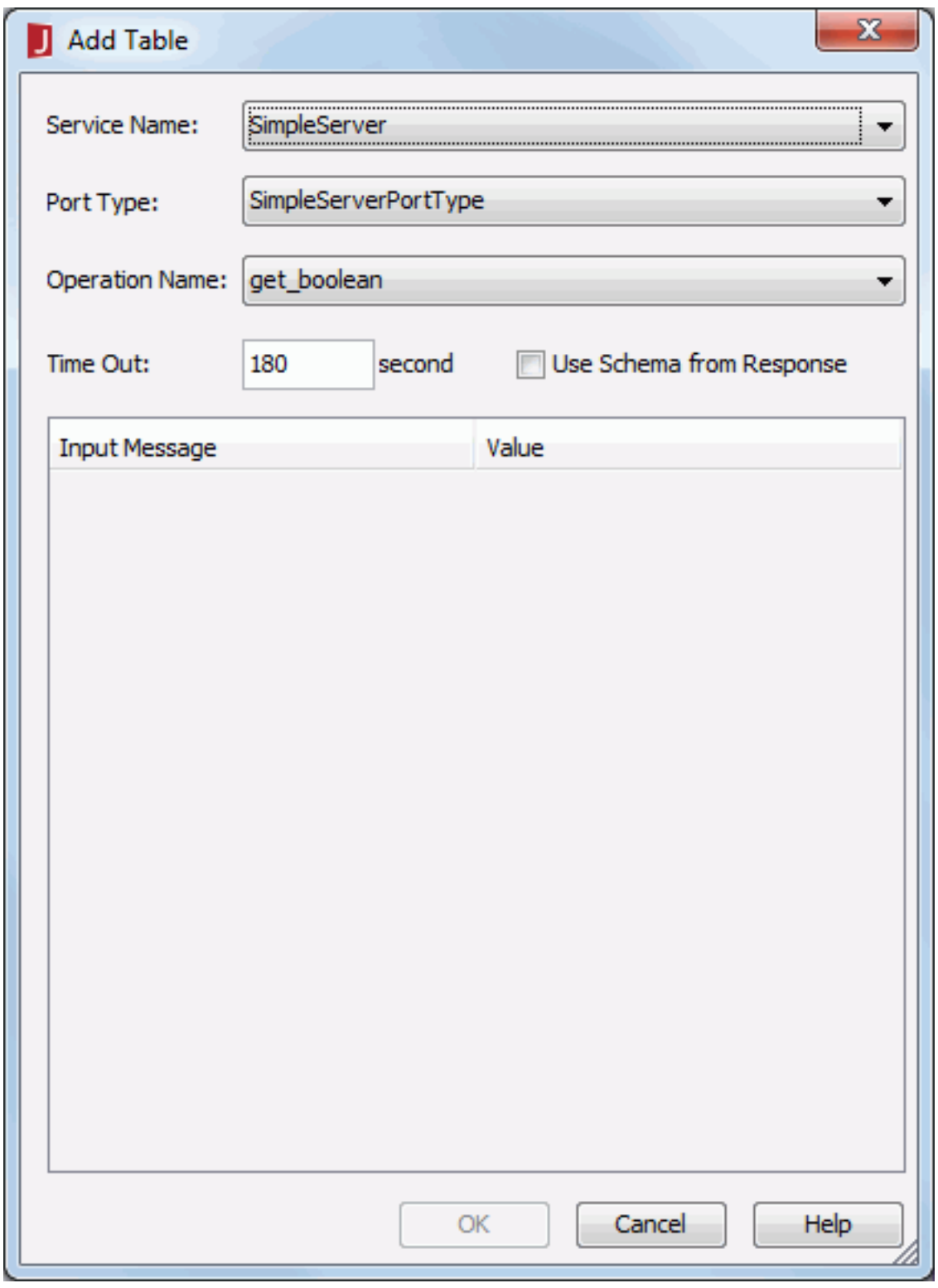
Note: When you configure the security policy for a web service connection, except for Key Store Type, all the other options in Security Configuration Setting dialog can be controlled by constant level formulas. However, the formula control is only available after the web service connection is already set up, that is when you edit an existing web service connection.

Adding tables to a web service connection

After setting up the WSDL connection, all the information of the web service defined in the WSDL file will be stored in the catalog, including information of service, operation, and the input message and output message in each operation. You can add tables to a web service connection based on these information. However, you can only define tables based on operations which support SOAP binding. Queries can be built on these tables and a report is developed from a query.

To add tables to a web service connection:

1. Select the **Tables** node of the web service connection, right-click it and then on the shortcut menu, click **Add Table**. The Add Table dialog appears.



2. Specify a service from the Service Name drop-down list.
3. If you use WSDL 1.1 to define the web service, you also need to specify the port type for the selected service from the Port Type drop-down list.
4. All the operations with SOAP binding that are included in the selected service will be listed in the Operation Name drop-down list. Select one as required, then the input message of the operation will be displayed. You can define the value for the input message if necessary by either typing in the value in the text field in the Value column, or clicking in the text field to select or define a constant level formula or parameter in the current catalog data source to return the value.

   If the Input Message column is empty, that is to say, the input message has its own defined values, so there is no need for you to provide it the value.
5. Type the number of seconds in the Time Out text field to specify how long to wait to get the specified service and operation information defined in the WSDL file.
6. If you use WSDL 1.1, you may sometimes find that the XML schema described by the output message does not match the concrete XML instance in the SOAP responded from the web service, as a result, JReport will be unable to read data properly from that XML instance. In this case, you can check the option **Use Schema from Response**, so as to ignore the XML schema in the output message and directly parse the XML schema from the specific XML instance included in the SOAP responded from the web service.
7. When done, click the **OK** button to accept the changes and exit the dialog.

By default, when you add a table, you add all the columns JReport finds in the table. However, JReport gives you the flexibility to remove columns in a table, leaving only the columns useful to your reports. To remove a column from a table, right-click the column and select **Delete** from the shortcut menu.

Also, you can add the columns back to a table after you have removed them. To do this:

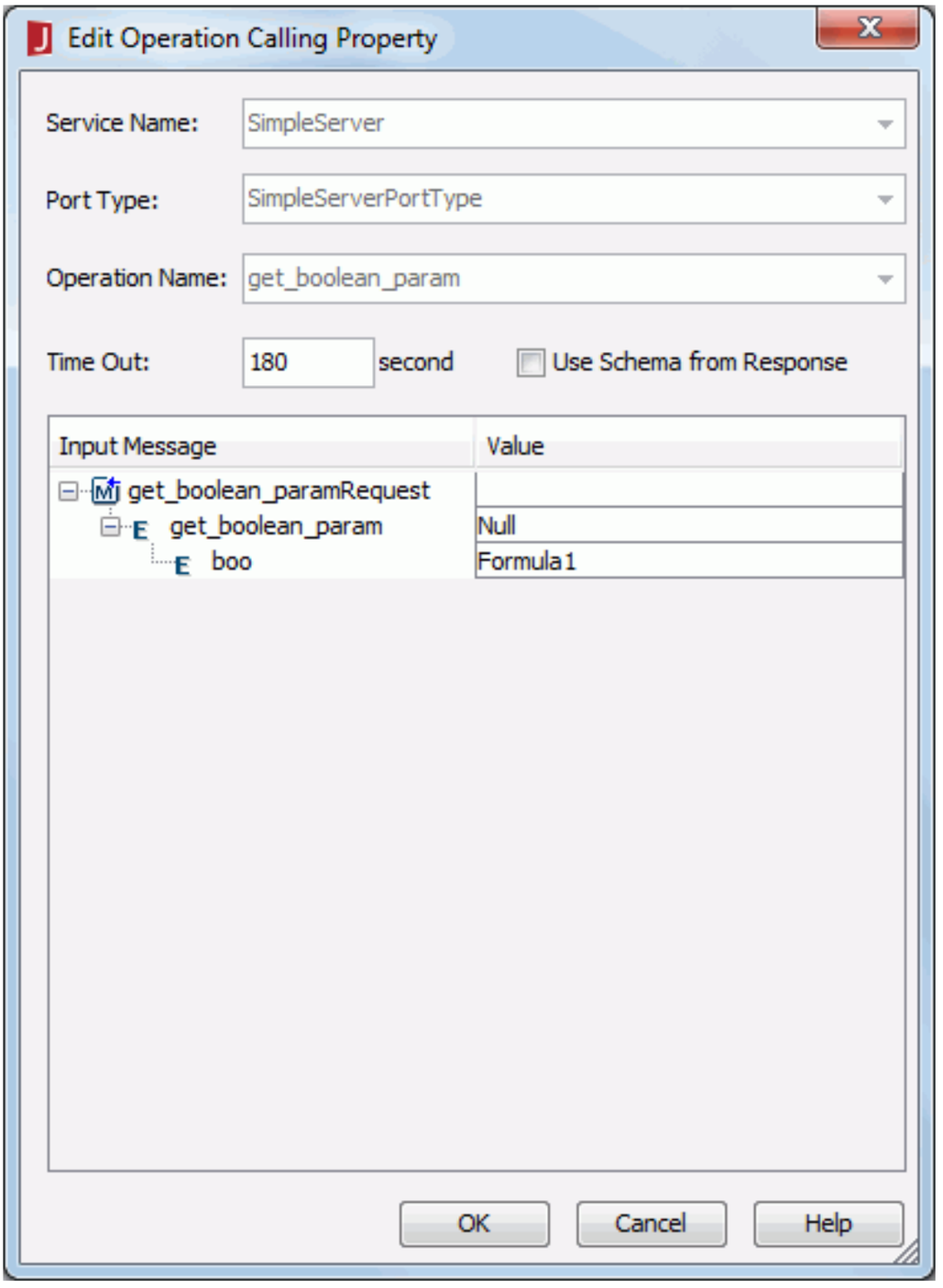
1. Right-click the table to which you want to insert columns, then select **Insert Column** from the shortcut menu.
2. In the Add Column dialog, select the required columns from the Columns box.

   There will be no available columns in the Columns box if you do not delete any in the table first since all columns are added automatically to start.
3. Click the **Done** button to add the columns to the table.

Editing the operation used to create tables

After creating the table, the service and operation you have selected to create the table will be displayed under the Tables node of the web service connection in the Catalog Browser. If you use WSDL 1.1 to define the web service, the defined port type will also be included. You can edit the information of the operation if necessary. To do this, follow the steps below:

1. Select the operation node, right-click it and then select **Edit** from the shortcut menu. The Edit Operation Calling Property dialog appears.



2. In the Input Message column, select the input message you want to edit, then modify its value in the text field of the Value column as required. You can either input the value, or click in the text field to select or define a constant level formula or parameter in the current catalog data source to return the value of the input message.

3. Type the number of seconds in the Time Out text field to specify how long to wait for the operation to complete.

4. If you use WSDL 1.1 to define the web service, the option Use Schema from Response will be available, using which you can decide whether to ignore the XML schema described in the output message and parse the XML schema directly from the specific XML instance in the SOAP responded from web service.

5. When done, click **OK** to accept the changes and exit the dialog.

Example: Developing reports from web service data source

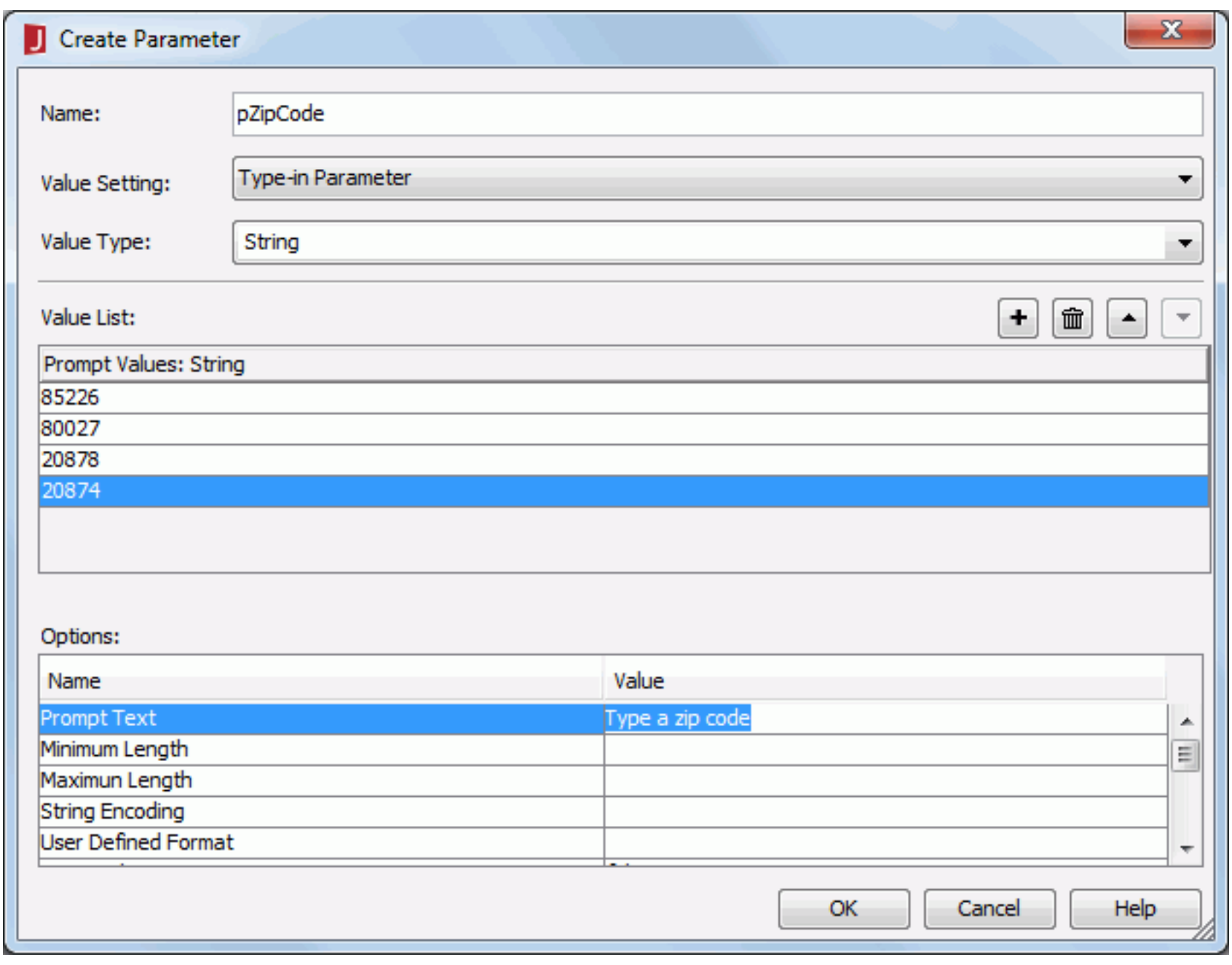
This example details how to use web service as a data source to create reports.

Setting up the web service connection

1. Open an existing catalog.
2. In the Data tab of the Catalog Browser, select any data source contained in it, click the **New Data Source** button on the toolbar, then in the New Data source dialog, specify the name of the data source as **Medicare Providers**, select the **SOAP Web Service** connection type and click **OK**.
3. In the SOAP Web Service Data Source dialog, check the **URI** radio button, and then type the URI string: **http://www.webservicex.net/medicareSupplier.asmx?WSDL** in the text field to get the WSDL file.
4. Keep the default settings for other options in the dialog, and then click **OK**.

Adding a table to the web service connection

1. Locate the **Tables** node in the web service connection, right-click it and select **Add Table** from the shortcut menu.
2. In the Add Table dialog, select **GetSupplierByZipCode** from the Operation Name drop-down list.
3. In the Input Message column, select **zip**, then in its value column, click and select **<Add Parameter...>** from the drop-down list.
4. In the Create Parameter dialog, input **pZipCode** as the parameter name in the Name text field.
5. Select **Type-in Parameter** from the Value Setting drop-down list.
6. Set the data type of the parameter value to **String** from the Value Type drop-down list.
7. In the Value List box, click to add a value line, click in it and then type in **85226**.
8. Repeat the above step to add three more values one by one: 80027, 20878, 20874.
9. In the Options box, enter **Type a zip code** as the prompt text.
10. Click **OK** to return to the Add Table dialog.



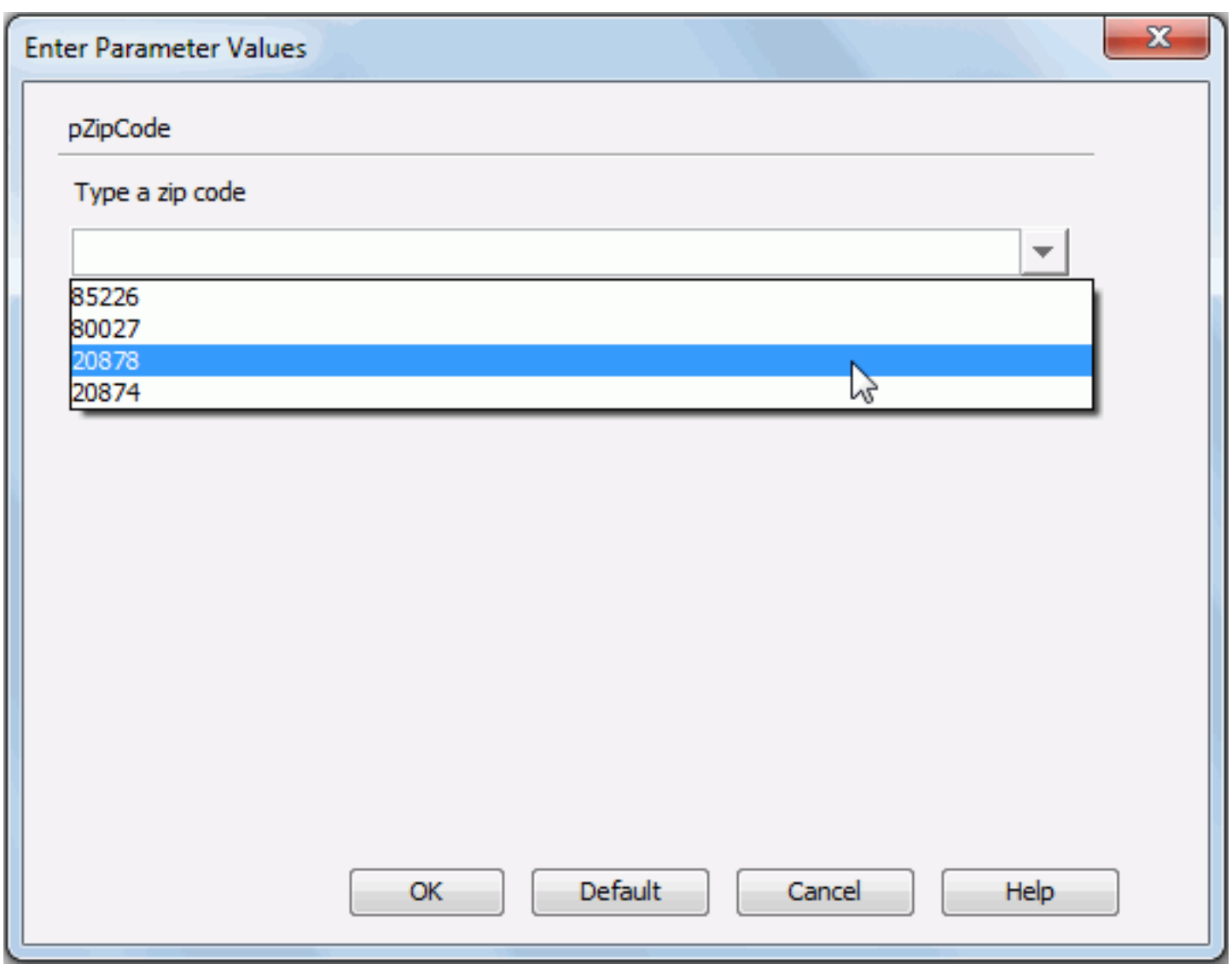
11. Click **OK** to add the table to the web service connection.

Building a query based on the table

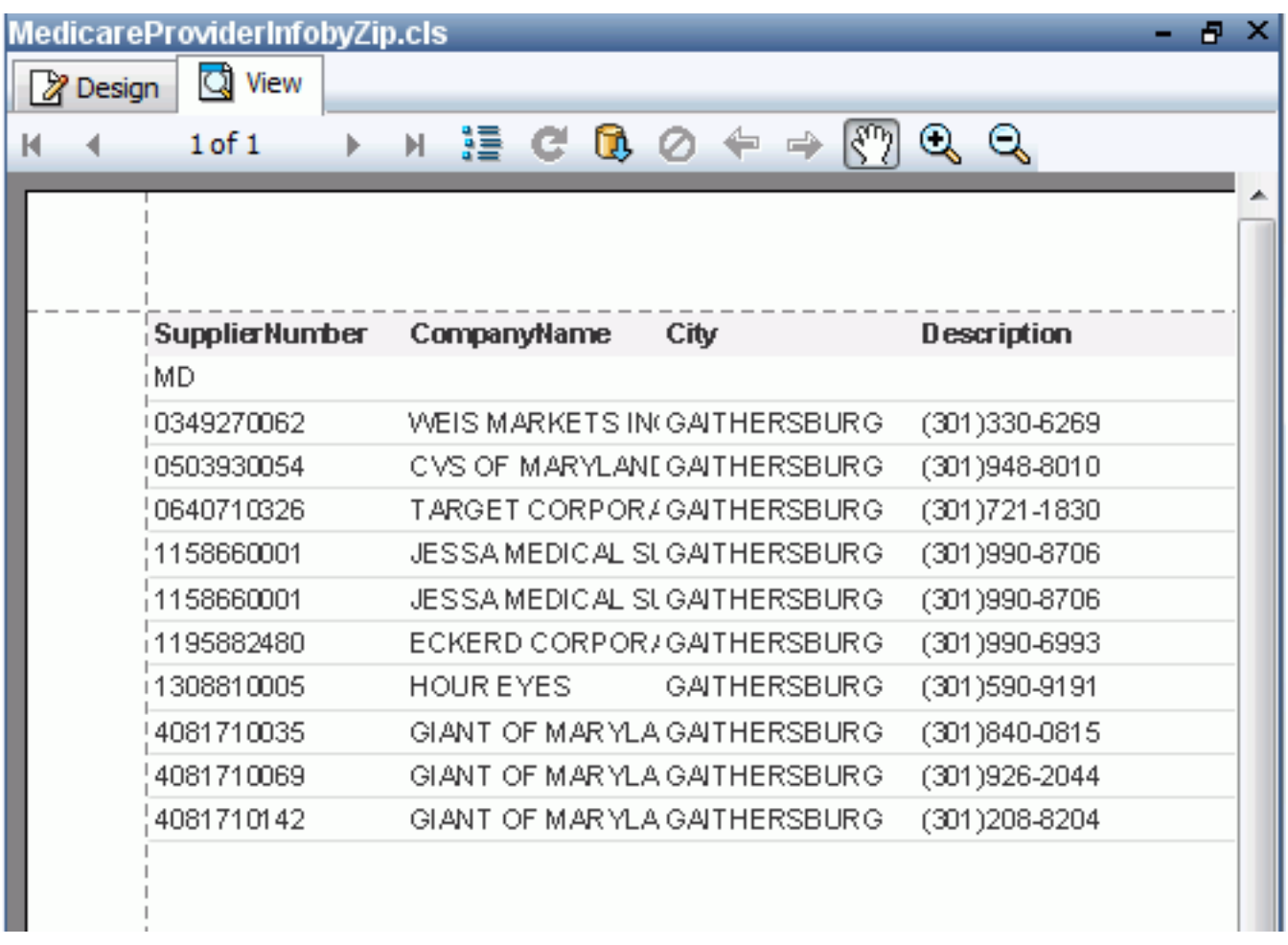
1. Locate the **Queries** node in the data source on which the web service connection is set, right-click it and select **Add Query** from the shortcut menu.
2. Specify the query name as **MedicareProvider** in the Input Query Name dialog, then click **OK**.
3. In the Query Editor window, click **Query** > **Add Table/View/Query** on the menu bar.
4. In the Add Table/View/Query dialog, expand the web service connection and select **GetSupplierByZipCodeResponse_SupplierDataLists_SupplierDatas_SupplierData** in the Tables node from the left box, then click the button to add it to the right box.
5. Click **OK**, the selected table will be displayed in the Query Editor window. Select all the columns in the table by selecting the ***** checkbox.
6. Click **OK** to save the query and leave the window.

Developing a report based on the query

1. In JReport Designer, click **File** > **New** > **Page Report** on the menu bar.
2. In the New Page Report dialog, clear the Report Title text box, and select **Banded** from the layout box, then click **OK**.
3. In the Table Wizard, select the query **MedicareProvider** in the data source Medicare Provider in the Data screen.
4. In the Display screen, from the Resources box, add the following fields one bye one: **SupplierNumber**, **CompanyName, City** and **Description**.
5. Click **Next** to go to the Group screen, add the field **Zip** as the group by field.
6. Click **Finish** to create the report.
7. Right-click the BandedHeader panel and select **Hide** from the shortcut menu to hide it from view. Repeat to hide the GroupFooter, BandedPageFooter and BandedFooter panels in the same way.
8. Click **File** > **Save** on the menu bar to save the report as **MedicareProviderInfobyZip.cls**.
9. Click the **View** tab to preview the report. The Enter Parameter Values dialog appears.



10. Type a zip code or select the one you want from the drop-down list, and then click **OK**. The report will be displayed according to your selected parameter value. For example, specify the zip code as 20878, then the report will be displayed as follows, showing information for zip code 20878 only:



MongoDB connections

A MongoDB connection contains relational schemas which are transformed from the collection schemas in a MongoDB data source. MongoDB uses BSON as the data storage and network transfer format for documents in collections. JReport Designer supports extracting of the metadata from the documents of a collection, and transforming schemas to relational tables. The collections and tables can then be added to the current connection from the transformed relational schemas and tables and accessed in the same way as JDBC-supplied tables. JReport Designer supports MongoDB 1.8 and above.

This section describes the following tasks:

- Transforming a MongoDB schema to a relational schema
- Setting up a MongoDB connection
- Managing schemas in a MongoDB connection
- Managing tables in a MongoDB connection

Transforming a MongoDB schema to a relational schema

You can transform a MongoDB schema to a relational schema. In the transformation process, elements in the MongoDB schema will be transformed to tables according to the ideographic transformation rules, and the hierarchical logic in the MongoDB schema will be maintained in the transformed relational schema.

Transformation rules

When a MongoDB schema is transformed to a relational schema automatically, tables will be built based on the schema, and the following rules will be applied:

- A database will be a name space for tables. The tables mapped from a collection in a database will belong to this name space.
- A top level document in a collection or an array element which contains documents will be mapped to a table.

After a document is mapped to a table, the following elements in the document will be mapped to the columns in the table:

- A simple element in the document or in the embedded document which does not belong to an array.
- An array element that contains a simple element in the document or in the embedded document which does not belong to an array.

However, as MongoDB databases use dynamic schemas, a heterogeneous set of documents may be stored within a collection, and only when the data in an array is of the same data type, can it be supported by JReport Designer. The metadata must be extracted from the documents of a MongoDB collection. The following rules and method for a given document will be applied for the extract. And before the data type defined in the MongoDB BSON document can function with JReport Designer, it should first be converted into a corresponding data type when the MongoDB schema is transformed, following the rules in the data type conversion table.

Extracting rules

- The documents are of the same structure.
- Some elements in the documents can be left blank.
- Documents with different structures will not interrupt the data process.

Extracting method

1. Merge all documents into a single document:

```
elementList=new List();
MergeDocument( Document[] ){
        for each document in Document[]
                for each element in document
                        if every E in the elementList does not have the same name as an existing element
                                then add the element to elementList.
}
If one element in elementList is a document, get all contents of the element in Document[] and merge them.
```

2. For the merged document, get each element's name, type, and sub type to build an element in the metadata.

Data type conversion table

BSON Data Type ID	BSON Data Type Name	Is Simple Data	SQL Type	SQL Type Name
"\x01"	Floating point	Y	8	Double
"\x02"	UTF-8 string	Y	12	VARCHAR
"\x03"	Embedded document	N	N/A	
"\x04"	Array (containing simple data only)	N	2003	ARRAY
"\x05"	Binary data	Y	-3	VARBINARY
"\x06"	Undefined — *Deprecated*	N	N/A	
"\x07"	ObjectId	Y	1	CHAR (hex string presents 12 bytes)
"\x08"	Boolean	Y	16	BOOLEAN
"\x09"	UTC datetime	Y	93	TIMESTAMP
"\x0A"	Null value	N	N/A	
"\x0B"	Regular expression	Y	12	VARCHAR
"\x0C"	DBPointer — *Deprecated*	N	N/A	
"\x0D"	JavaScript code	Y	12	VARCHAR
"\x0E"	Symbol — *Deprecated*	N	N/A	
"\x0F"	JavaScript code w/ scope	Y	12	VARCHAR
"\x10"	32-bit Integer	Y	4	INTEGER
"\x11"	Timestamp	Y	93	TIMESTAMP
"\x12"	64-bit integer	Y	-5	BIGINT
"\xFF"	Min key	N	N/A	
"\x7F"	Max key	N	N/A	

Sub types for binary

BSON Data Type ID	SQL Data Type Name
\x00"	Binary / Generic
"\x01"	Function
"\x02"	Binary (Old)
"\x03"	UUID (Old)
"\x04"	UUID
"\x05"	MD5
"\x80"	User defined

MongoDB hierarchical logic in relational schema

The XML hierarchical logic will be maintained in the transformed relational schema. The parent-child relationship in the MongoDB schema can be maintained by primary key and foreign key in tables, and this relationship can be reproduced by applying the join of the primary key and foreign key. The business view is based on query. The join will be pushed down to MongoDB with Map/Reduce. If MongoDB contains huge detail data, it is suggested that you set the Prefetch property of the business view to false.

Setting up a MongoDB connection

To set up a MongoDB connection to connect a JReport catalog to a MongoDB data source, follow the steps below:

1. Create a catalog or open a catalog, then in the Data tab of the Catalog Browser, right-click the node of a data source and choose **Add MongoDB Connection** from the shortcut menu.

   If you want to set up the connection by adding a data source to an existing catalog and then defining the connection on the data source, open the catalog, select any data source contained in it, click the **New Data Source** button on the Catalog Browser toolbar, then in the New Data source dialog, specify the name of the data source, select the **MongoDB** connection type and click **OK**.

   The MongoDB Connection Wizard appears.

2. In the MongoDB Connection Information screen, type in the host and port of the MongoDB server, and also the user ID and password to connect to the MongoDB server. Click **Next**.

3. In the Connection Options screen, click to add a line for specifying the conditional replica set member for the MongoDB connection. If necessary, click to add a new replica set member. Click to remove a selected replica set member. Then specify the options for the connection. Click **Next**.

4. In the Add Database screen (this screen will be displayed only when you have no right to access all the databases in the MongoDB data source), click to add a line for specifying the databases you can access in the MongoDB data source. If necessary, click to add a new database. Click to remove a selected database. Click **Next**.

5. In the Add Schema screen, add the collection schemas that will be transformed to relational schemas. Click **Next**.

6. In the Add Table screen, add the required tables that are transformed from the relational schemas to the connection.

7. Click **Finish** to confirm the transformed result and finalize the transformation process.

Note: When setting up the connection via SSL, you need to import the SSL certificate into the default JKS, and use the *-D* parameter to specify the system parameter in the file JRServer.bat in <server_install_root>\server\bin (The MongoDB Java driver mongo-2.12.0.jar is already within JReport. For MongoDB 3.0, the MongoDB Java driver mongo-2.13.1.jar is supported.). Then add *ssl=true* in the Options box of the Connection Options screen of the MongoDB Connection Wizard.

Managing schemas in a MongoDB connection

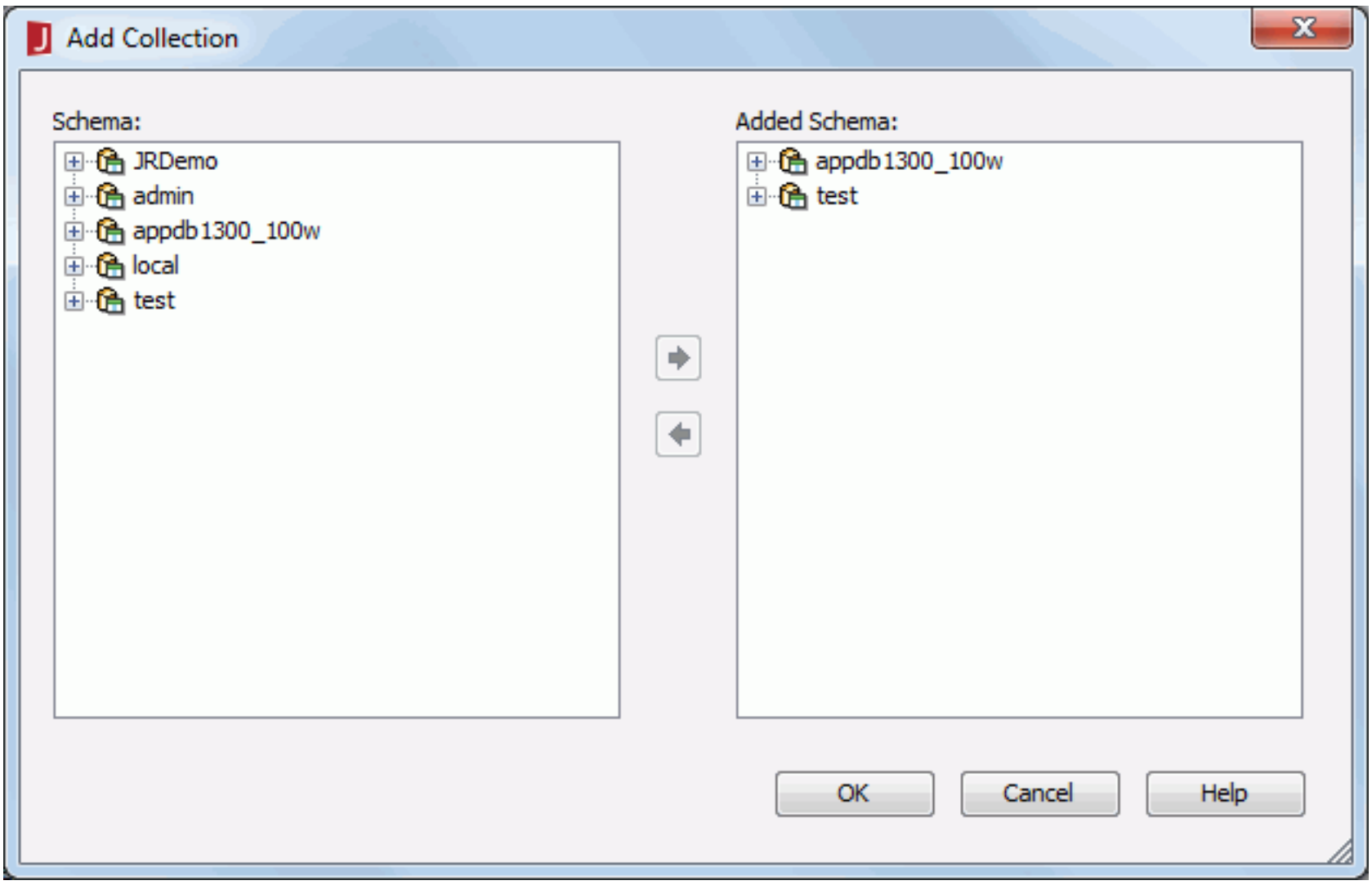
The relational schemas are transformed from the collection schemas in a MongoDB data source, and then you can add tables that are transformed from the relational schemas to the connection.

This document shows how to add more collections in the schemas to a MongoDB connection, refresh the schemas, and so on.

Adding schemas

To add relational schemas which are transformed from the collection schemas in a MongoDB file to a MongoDB connection, follow the steps below:

1. Right-click the **Schemas** node or an existing schema in the MongoDB connection and select **Add Collection** from the shortcut menu to display the Add Collection dialog.



2. Select the required collections in the schemas that are transformed from the collection schemas in the Schema box and then click add them to the Added Schema box.
3. After adding the required collection schemas, click **OK** to close the dialog.

Refreshing schemas

The schemas in your catalog are a temporary cache to improve performance when you design and test your report. Your data source will probably change over the time; however, these changes will not be reflected automatically in your catalog. To refresh all of the collection schema metadata from the MongoDB database, you can choose to refresh the schema information using the Refresh command on the shortcut menu of the schema. Then when the refreshing job is done, a reporting dialog will be shown, summarizing the changes and operations that have been taken.

Renaming and deleting databases of schemas

- To rename the database of the schemas, right-click the database and select **Rename** from the shortcut menu, and then input the required database name in the text box.
  - If the renamed database exists in the MongoDB data source, it will connect to the new database.
  - If the renamed database does not exist in the MongoDB data source, no data will be retrieved.
- To delete the database of the schemas, right-click the database and select **Delete** from the shortcut menu.

Tip: If you want to add all the deleted or missing databases at one time, you can choose to refresh the schema, which refreshes all of the collection schema metadata from the MongoDB database.

Managing tables in a MongoDB connection

A table contains fields mapped to attributes, top level documents, simple elements, array elements and other nodes in MongoDB documents. Queries can be built on these tables and a report is developed from a query or something else which is functionally similar.

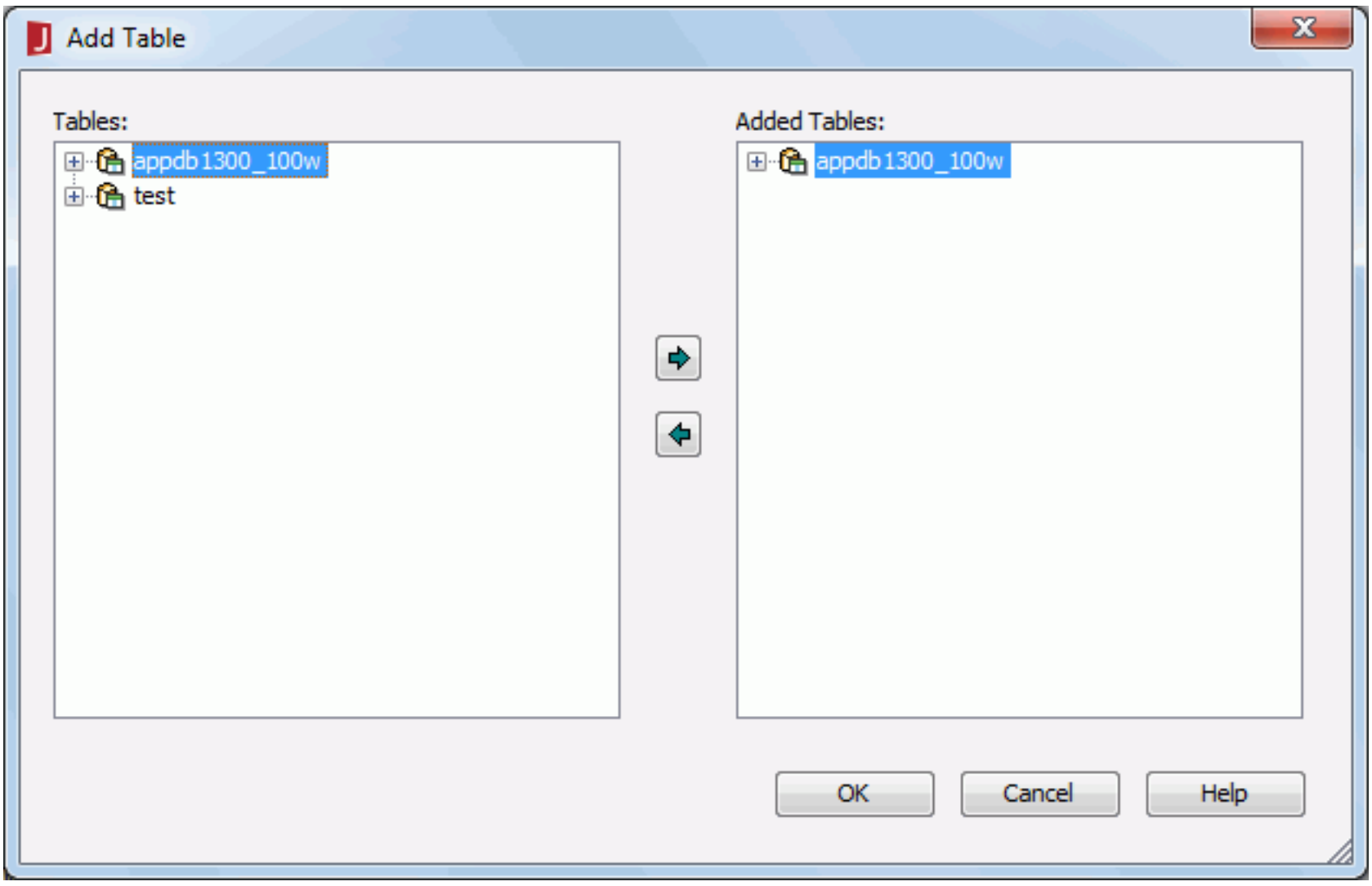
This document shows how to add more tables to a MongoDB connection, refresh the tables, and so on.

Adding tables

To add tables which are transformed from a MongoDB file to a MongoDB connection, follow the steps below:

1. Do either of the following:
   - Right-click the MongoDB connection and select **Add Tables** from the shortcut menu.
   - Right-click the **Tables** node of the MongoDB connection and select **Add Table** from the shortcut menu.
   - Right-click an existing table in the MongoDB connection if there is and select **New Table** from the shortcut menu.

   The Add Table dialog appears.



2. Select the required tables that are transformed from the relational schemas in the Tables box and then click add them to the Added Tables box.
3. After adding the required tables, click **OK** to close the dialog.

Refreshing tables

The tables in your catalog are a temporary cache to improve the performance when you design and test your report. Your data source may change over the time; however, these changes will not be reflected automatically in your catalog. To synchronize your tables in the connection and data source, you can choose to refresh the table information using the Refresh command on the shortcut menu of the table. Then when the refreshing job is done, a reporting dialog will be shown, summarizing the changes and operations that have been taken.

Removing and adding columns to tables

By default, when you add a table, you add all the columns that reside in the table. However, JReport gives you the flexibility to remove columns from a table, leaving only the columns useful to your reports. To remove a column from a table, right-click the column and select **Delete** from the shortcut menu.

Also, you can add the columns back into a table after you have removed them. To do this:

1. Right-click the table to which you want to insert columns, then select **Insert Column** from the shortcut menu.
2. In the Add Column dialog, select the required columns from the Columns box and click **Add** to add the columns to the table.

   There will be no available columns in the Columns box if you have not previously deleted columns.
3. Click the **Done** button to close the Add Column dialog.

Tip: If you want to add all the columns in a table at one time, you can also choose to refresh the table, which synchronizes the table in the catalog with the one in your database.

HIVE connections

The HIVE connection functions the same as a JDBC connection. It also supplies the JDBC driver and metadata information which are the same as JDBC. However, it expands many native functions to support non SQL and huge data, which are not supported by JDBC. The libraries from CDH4.1.1 that support HIVE 0.9 are listed as below. You can add them in `<install_root>\lib` when needed:

- hadoop-common-2.0.0-cdh4.1.1.jar
- hadoop-core-2.0.0-mr1-cdh4.1.1.jar
- hive-exec-0.9.0-cdh4.1.1.jar
- hive-jdbc-0.9.0-cdh4.1.1.jar
- hive-metastore-0.9.0-cdh4.1.1.jar
- hive-service-0.9.0-cdh4.1.1.jar
- libfb303-0.7.0.jar
- libthrift-0.7.0.jar
- slf4j-api-1.6.1.jar
- slf4j-simple-1.6.1.jar

If you use higher HIVE versions, you can replace the libraries with new jars.

For more information about HIVE connections, refer to JDBC connections.

JSON connections

A JSON connection contains relational data which is transformed from a JSON data source.

JReport Designer can parse JSON data to extract the schema including JSON metadata (JSON objects and the relation between the objects), then transform the schema to logic RDBMS tables, namely, map JSON object classes to tables, and the relation to primary/foreign key columns in tables. The tables can then be added to the current connection from the transformed relational tables.

This section describes the following tasks:

- Extract metadata from JSON data
- Transforming a JSON schema to a relational schema
- Setting up a JSON connection
- Managing tables in a JSON connection

Extract metadata from JSON data

A JSON metadata contains a root element whose element type can be object class or object array. An object class or object array element contains some elements whose element type can be object class, object array, attribute or simple data array with the data type of string, number or Boolean. JSON objects in an array should have the same structure, the members (name/value pairs) in an object cannot be of the same name, and for nested array, only the first layer will be kept.

Extracting method

When a JSON schema is transformed to a relational schema automatically, RDBMS tables will be built based on the schema, and the following rules will be applied:

- Elements should be extracted from the root value of JSON data hierarchically.
- When creating the root element, the element name is null, and the root element type is object class or object array if the root value of JSON data is an object class or an object array, else the root element type is an empty object class.
- If the value of an object member is a simple value, an attribute element will be created under current element. If it is a simple value array, a simple data array element will be created. If it is an object array, an object array element will be created. If it is an object class, an object class element will be created. The element type will be set according to the conversion rules in the Data type conversion table below. The element name is the object member name.
- All elements of the same name in object members of an object array will be merged into an element.

Data type conversion table

Before the JSON data type defined in the JSON file can function with JReport Designer, it should first be converted into a corresponding SQL data type when extracting metadata from JSON data, following the rules below.

JSON Data Type	SQL Data Type
String (following the format Combined date and time representations in ISO 8601)	TIMESTAMP
String (following JDBC timestamp escape format)	TIMESTAMP
String (following the format Calendar dates in ISO 8601)	DATE
String (following the format Times with time zone designators in ISO 8601)	TIME
String	VARCHAR
Number (excluding fraction and exponent)	INTEGER
Number (including fraction and exponent)	NUMERIC
Boolean	BOOLEAN

Notes:

- If the data type of all values for members of the same name in an object array is Number, including not only integer, but also fraction or exponent, the data type of all values will be converted to NUMERIC.
- If the values for members of the same name in an object array are of different data types and at least one of them is String, the data type of all values will be converted to VARCHAR.
- The values for the members of the same name in a JSON schema cannot be mixed with simple data array and single value, such as string, number,Boolean, or null, otherwise, the extracted schema maybe incorrect.

Transforming a JSON schema to a relational schema

A JSON schema can be transformed to a relational schema. In the transformation process, elements in the JSON schema will be transformed to either tables or columns in tables according to the ideographic transformation rules and be named according to the naming rules.

Transformation rules

When a JSON schema is transformed to a relational schema automatically, RDBMS tables will be built based on the schema, and the following rules will be applied:

- If an object class or object array element has at least one attribute element or simple data array element, or its embedded element can be mapped to tables, the object class or object array element will be mapped to a table, and its attribute or simple data array element will be mapped to columns in the table.
- When mapping an attribute element to a column, the value of its Mapped SQL Type will be set as the value of the SQL Type of the column.
- For an embedded object class or object array element, a built-in column called foreign key will be created for the mapped table. If there is no built-in column called primary key in its parent element table, the column will be created. The SQL type of the columns primary key and foreign key is SQL type 4 (64 bit Integer).

Naming rules

The RDBMS tables and columns in tables are named based on the rules below:

- For the root element, the table name is ROOT. If the name is not unique, the character _ will be added before the name one by one till it is unique.
- For elements other than the root element, the table name is the element name. If the table name is not unique, the names of the ancestor elements will be added before the table name one by one and be separated by _ till it is unique.

Setting up a JSON connection

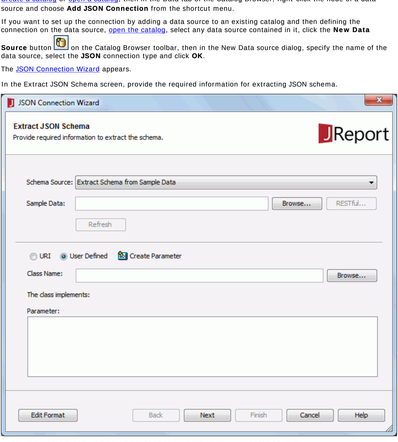
To set up a JSON connection to connect a JReport catalog to a JSON data source, follow the steps below:

1. Create a catalog or open a catalog, then in the Data tab of the Catalog Browser, right-click the node of a data source and choose **Add JSON Connection** from the shortcut menu.

   If you want to set up the connection by adding a data source to an existing catalog and then defining the connection on the data source, open the catalog, select any data source contained in it, click the **New Data Source** button on the Catalog Browser toolbar, then in the New Data source dialog, specify the name of the data source, select the **JSON** connection type and click **OK**.

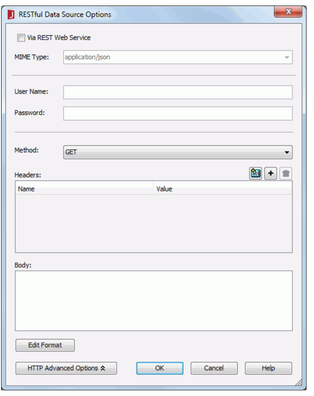
   The JSON Connection Wizard appears.

2. In the Extract JSON Schema screen, provide the required information for extracting JSON schema.



   - If Extract Schema from Sample Data is selected from the Schema Type drop-down list,
     a. Input the URI string of the sample JSON data file directly in the Sample Data text field or click **Browse** to specify the sample data file.

        If the file is from a remote URL, click **RESTful** to specify the RESTful data options for the sample data in the RESTful Data Source Options dialog.



        i. Specify whether to receive remote data via the REST web service. If the Via REST Web Service option is unchecked, the remote data will be received via the protocol in the URL you specify in the Sample Data text field. If it is checked, the remote data will be provided by the REST web services on the application server, and the REST web service client API (such as JAX-RS client API of Java EE) will be used to get the remote data. The URL, the HTTP method (HTTP.GET), MIME types and so on need to be specified to set up the connection.

        ii. Specify the MIME type for the REST web service data sources. You can input it in the MIME Type text field directly or select one from the drop-down list. This option is available only when Via REST Web Service is checked.

        iii. Input the user name and password for remote data authentication.

        iv. Click **HTTP Advanced Options** to specify the advanced options for HTTP. Select an HTTP method from the Method drop-down list to send the request, which can be GET or POST. Specify the user defined HTTP headers and body. Click **OK** to apply the specified RESTful data source options and close the dialog.

        To add a header line, click and if a header is not required, select it and click to remove it. If needed, click to create a parameter for use. You can also click the **Edit Format** button to define the format of the values referenced in the HTTP headers or body.

     b. Specify the source of the data, which can be from a URI or a user defined interface. To use a parameter to specify the file at runtime, you can also click the **Create Parameter** button to create one.

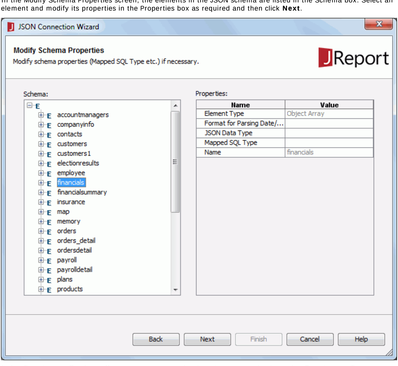
        - If URI is selected, input the URI string directly in the Instance text field or click **Browse** to specify the instance file. If the file is from a remote URI, click **RESTful** to specify the RESTful data options for the instance data in the RESTful Data Source Options dialog. For details about the dialog, refer to the step above.
        - If User Defined is selected, provide the class name with package name in the Class Name field. You can also click **Browse** to find the class file. The class you entered should actually exist and can be found by JReport Designer, which means the class should be in the class path of the system environment. After filling in this field, the class name of the interface that the class implements will be displayed automatically behind The class implements:. Then specify the parameter for the user defined interface in the Parameter box.

        You can use parameters or constant level formulas predefined in the current catalog data source or the special field User Name in the URI or parameter string, and the format of the predefined parameters used in the string is "@" + parameter name. For example, if the parameter string of a class is SQL=select * from employee, and you want to use the parameter sql in a catalog to replace the part after "=" in the string, then the parameter string of the class will be SQL=@sql. Moreover, if a parameter string contains characters, such as @, ', or double quotation marks, or other strings that do not need to be parsed, quote the whole string with double quotation marks. You can also click the **Edit Format** button to edit the format of the referenced values.

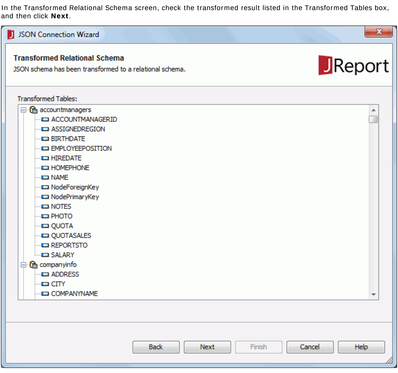
   - If Extract Schema from Instance Data is selected from the Schema Source drop-down list, specify the instance file and the RESTful data options for the instance data as required.

   When done, click **Next** to go to the next screen.

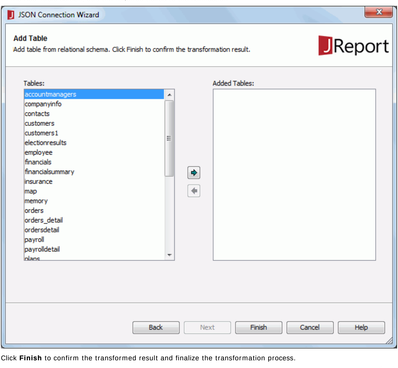
3. In the Modify Schema Properties screen, the elements in the JSON schema are listed in the Schema box. Select an element and modify its properties in the Properties box as required and then click **Next**.



4. In the Transformed Relational Schema screen, check the transformed result listed in the Transformed Tables box, and then click **Next**.



5. In the Add Table screen, add the required tables to the connection.



6. Click **Finish** to confirm the transformed result and finalize the transformation process.

Managing tables in a JSON connection

A table contains fields mapped to attributes, simple data array elements and other nodes in JSON files. Data will be fetched from the JSON data according to the table information and JSON metadata. Queries can be built on these tables and a report is developed from a query or something else which is functionally similar.

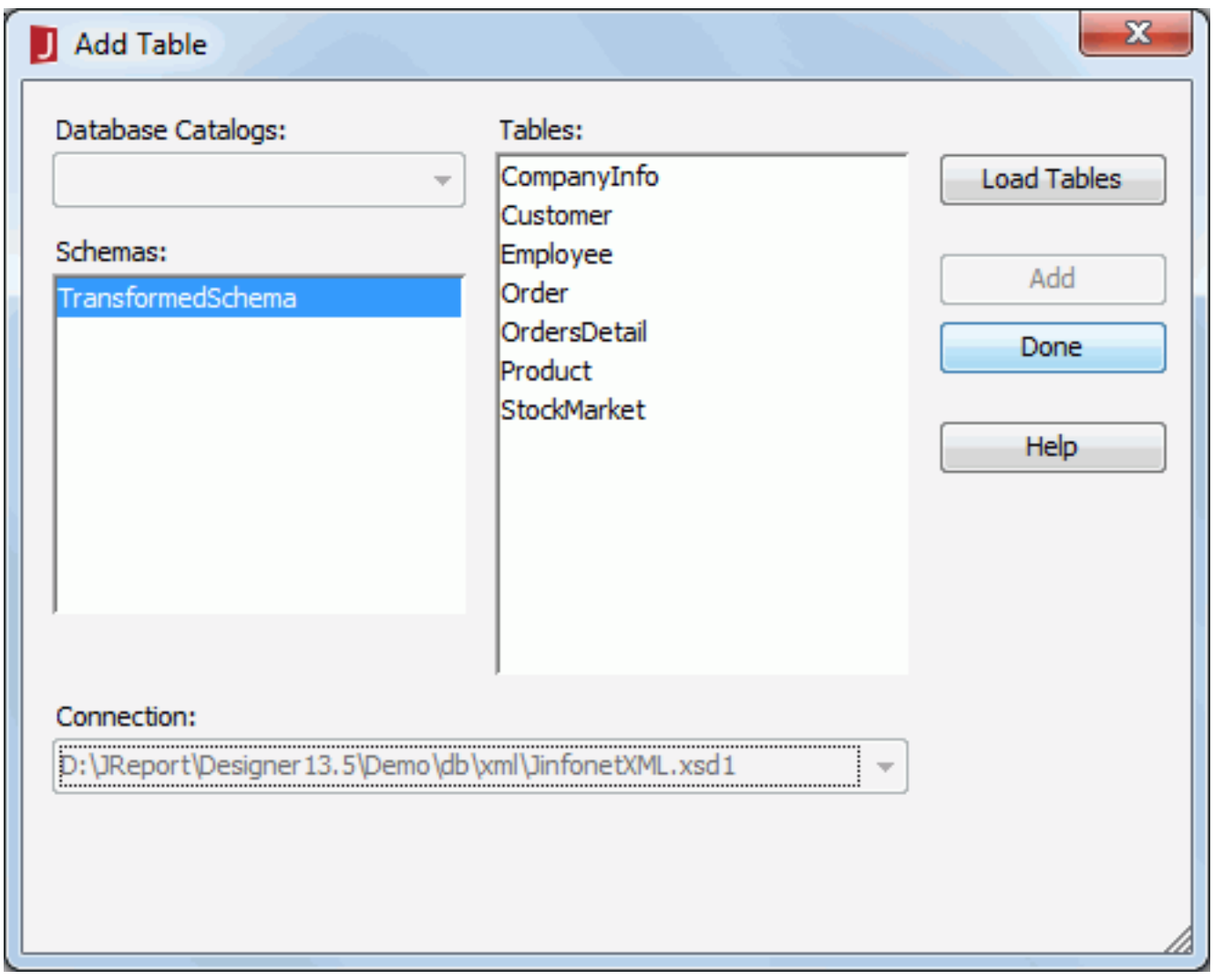
This document shows how to add more tables to a JSON connection, organize the tables, and so on.

Adding tables

To add tables which are transformed from a JSON file to a JSON connection, follow the steps below:

1. Do either of the following:
   - Right-click the JSON connection and select **Add Tables** from the shortcut menu.
   - Right-click the **Tables** node of the JSON connection and select **Add Table** from the shortcut menu.
   - Right-click any folder in the Tables node of the JSON connection if you have already created some and select **Add Table** from the shortcut menu.
   - Right-click an existing table in the JSON connection if there is and select **New Table** from the shortcut menu.

   The Add Table dialog appears.



2. Click the **Load Tables** button. The tables contained in the schema that is transformed from the JSON file will then be displayed in the Tables box.
3. Choose the required tables in the Tables box, and then click **Add**.

   To choose consecutive tables, click the first table, press and hold down the **SHIFT** key, and then click the last table. To choose tables that are not consecutive, press and hold down **CTRL**, and then click each table.
4. After adding the required tables, click **Done** to close the dialog.

Refreshing tables

The tables in your catalog are a temporary deposit to improve the performance when you design and test your report. Your data source keeps changing over the time. However, these will not be reflected automatically in your catalog. To synchronize your tables in the connection and data source, you can choose to refresh the table information using the Refresh command on the shortcut menu of the table. Then when the refreshing job is done, a reporting dialog will be shown, summarizing the changes and operations that have been taken.

Organizing tables

You can organize the tables in a JSON connection by arranging them in different folders in the Catalog Browser.

To add a folder:

1. Right-click the **Tables** node, or any existing folder and select **Add Folder** from the shortcut menu.
2. A new folder is added to the Tables node, or in the selected folder. Type a name for the folder in the editing area, and then click outside to confirm the change.

To move tables to a folder:

1. Right-click the table you want to move in the Tables node and select **Move To** from the shortcut menu.
2. In the Move Table dialog, select the folder to which the table will be moved, then click **OK**.

Removing and adding columns to tables

By default, when you add a table, you add all the columns reside in the table. However, JReport gives you the flexibility to remove columns from a table, leaving only the columns useful to your reports. To remove a column from a table, right-click the column and select **Delete** from the shortcut menu.

Also, you can add the columns back to a table after you have removed them. To do this:

1. Right-click the table to which you want to insert columns, then select **Insert Column** from the shortcut menu.
2. In the Add Column dialog, select the required columns from the Columns box and click **Add** to add the columns to the table/view.

   There will be no available column in the Columns box if you have not previously deleted columns. That is to say, you cannot add columns that do not belong to the table.
3. Click the **Done** button to close the Add Column dialog.

Tip: If you want to add all the columns in a table at a time, you can also choose to refresh the table, which synchronizes the table in catalog with the one in your database.

Queries

Relational databases typically contain many rows of data, with each row constituting a separate record. Most relational database queries retrieve only a portion of the records contained in a table. The WHERE clause qualifies the query command statement to limit the data to specific records from the tables.

Queries are a higher-level object in a catalog. Its concept is similar to that in the database. The working objects a query can select include tables/views (DBFields), formulas or summaries, and parameters, and the tables/views it can select from are the mapped tables/views defined in a JReport catalog. In this way, a query is independent from the raw database. You can use queries to view, change and analyze data in different ways, and JReport can help you with the building of various professional reports based on queries. When making a query, you can place criteria or restrictions on the data to extract only the required data from the database. For example, instead of having to view all the customers of your company, you can view just the customers from Japan. To do this, you specify conditions that limit the results to the records with Country field = "Japan".

JReport Designer provides you with an interactive query designer - the Query Editor, to build any queries as you like. It also enables you to build queries using the pre-join feature. However, note that only the queries built in JDBC connections can support all of the functions. Due to the unique characteristics of other connections, only part of the query functions are supported for them.

In addition, JReport Designer provides you with the Data Manager which allows you to control the data retrieval of your queries, including the number of rows to be displayed and the duration required for the retrieval. It can also keep access information from previous runs of a query (including imported SQL, stored procedure and UDS which function as queries).

The following topics describe queries:

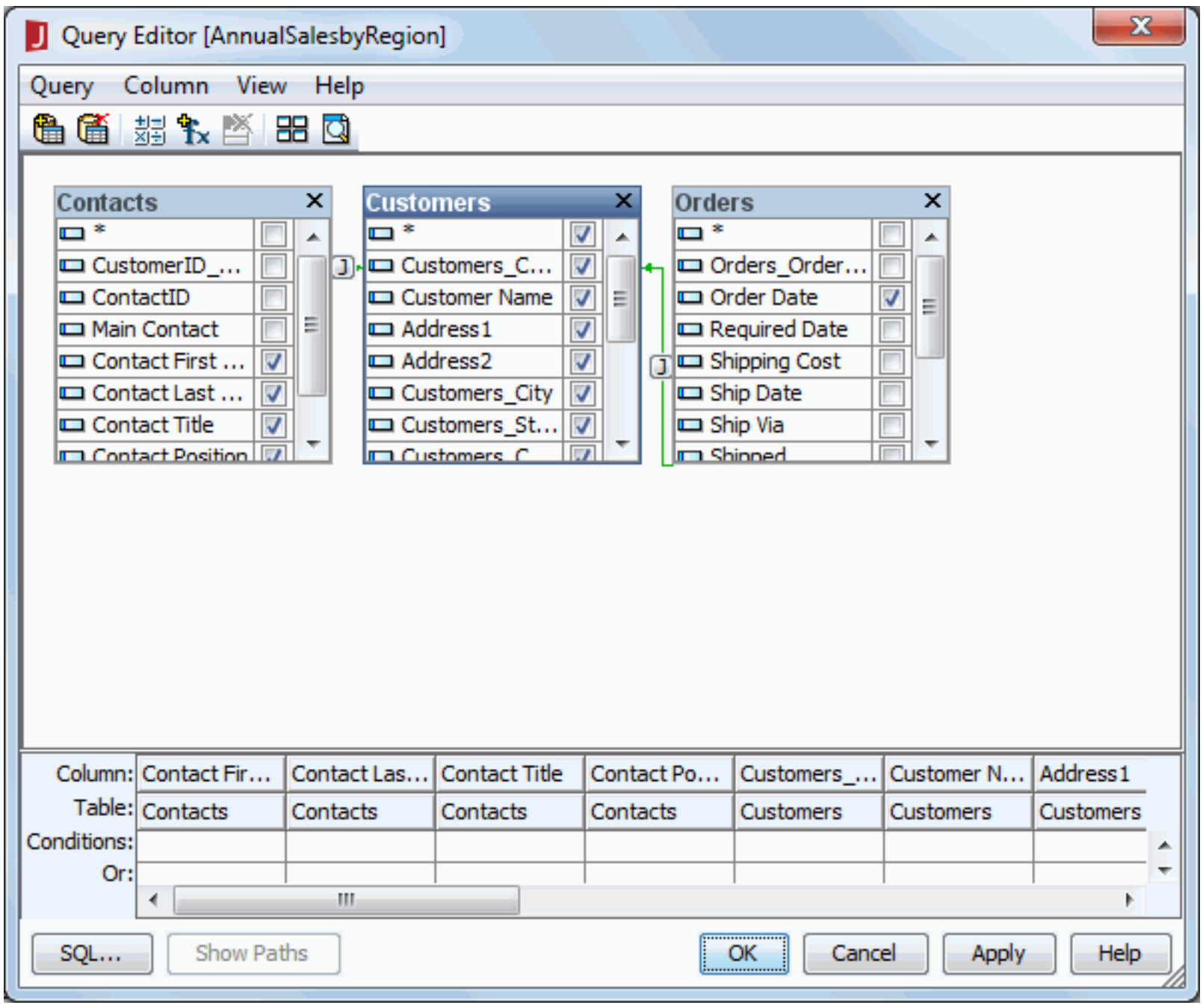
- Creating a query
- Using pre-joins in queries
- Editing a query
- Previewing a query
- Creating and modifying a union query
- Locking queries
- Developing reports from queries
- Dynamic queries
- Query modifiers
- Cached query results
- Data Manager

See an example: The SampleComponents catalog, included with JReport Designer, contains reports that have examples of how queries could be used in a report. For the query example, open `<install_root>\Demo\Reports\SampleComponents\Query.cls`.

Creating a query

To create a query in a catalog, follow the steps below:

1. In JReport Designer, open the required catalog, then do either of the following:
   - In the Catalog Browser, expand the data source in which you want to create the query, right-click the **Queries** sub node and select **Add Query** from the shortcut menu.
   - Click **File** > **New** > **Query** on the menu bar, specify the data source in which the query will be created, then click **OK**.
2. In the Input Query Name dialog, give a name for the query and click **OK**. The Query Editor window appears.



3. On the Query menu, click **Add Table / View / Query** to add data to the query. If the catalog data source is connected with multiple connections, you can mash-up multiple data resources in the query that come from all these connections, including tables, views, synonyms, queries, imported SQLs, stored procedures and user defined data sources. When a query, imported SQL, stored procedure or user defined data source is added, it will be added as a single table with all of its columns.
4. Select the required columns in each table. To select all, click *****. The columns will be displayed in the criteria panel of the Query Editor.

   For tables in query of a MongoDB connection, you cannot select all columns by clicking *****. The PrimaryKey and ForeignKey columns in each table cannot be selected to a query.
5. Join the tables as required.

   Based on the Auto Join options that are selected in the Query Editor category of the Options dialog, JReport will join the tables automatically. However, if the pre-join feature is enabled for the data source where the query is created, the auto join settings will not take effect, instead, JReport applies the pre-joins to the added tables. You can also add more joins among these tables.
6. Create filters on the query.
7. Add some computed columns and formula fields to the query if required.
8. Select **Apply** from the Query menu to save the query.

When a query has been added to a catalog, you can further edit it to suit your requirements.

Note: Normally, a query returns all the records that match its search criteria without considering whether there are duplicated ones. You can decide to get only one copy for each record by checking Select Distinct on the Query menu. When this option is enabled, SQL SELECT statements are treated as SELECT DISTINCT statements. The query will search for identical records and ensures to return them only once instead of returning duplicate records from the database. However, this feature is only supported on queries built in JDBC connections.

Using pre-joins in queries

JReport Designer provides you with the option of choosing whether or not to use the Pre-join feature when creating queries. If you want to use it, you must have this option enabled.

To turn on the Pre-join feature:

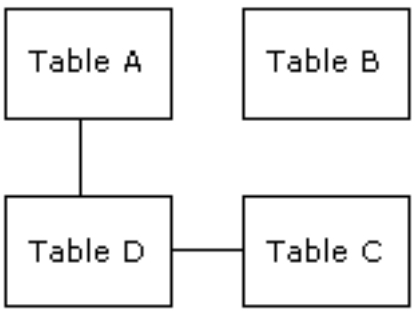
1. Make sure Forbid editing data object properties in the Catalog category of the Options dialog is unchecked.
2. In the Catalog Browser, right-click the data source in which you want to create queries, then select **Properties** to expand the Properties sheet.
3. Set the Pre-join property of the data source to **true**.

Then, when you add tables contained in the data source to queries in the same data source, the pre-joins defined on the data source will be automatically applied to the tables.

Creating queries using pre-join

When you create a query using the pre-join feature, the following cases may appear:

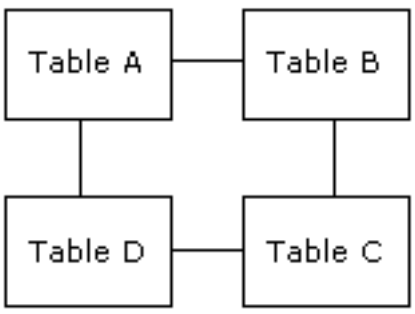
- If JReport Designer needs other tables to bridge the tables you have added, you will be prompted to decide whether to add another table to the query. For example,



  Suppose you've added table A and C to your query, a message box will then appear asking you whether you want to add table D in order to create links between tables A and C.

  Note: The prompt message appears only when the option "Show warning message when adding tables" is selected in the Query Editor category of the Options dialog. Otherwise, JReport will add the bridged table automatically without any prompt.

- If there is more than one paths available in the pre-join, as with this case,



  You've added table A and C to the query, and there are two paths available - ABC and ADC. JReport Designer will ask you for the path ABC or ADC.

  JReport Designer provides two methods for seeing the path information of a query:

  - In the Catalog Browser, expand the Properties sheet and select the query that uses the pre-join path. You will then see a property called Path Name which shows the name of the path used.
  - In the Query Editor window, click the **Show Paths** button.

Editing queries using pre-join

One query is bound with at most one path. Once a path has been specified for a query, when you modify the relationships among the tables, you can make changes to a specific query based on that path, but the path itself will not be changed. When you edit a query, the following cases may appear:

- Adding tables to a query that has already been specified a path - If more than one path is suitable after the tables have been added, the specific path will still be used. That is, you cannot choose another path to replace the original one.
- If the path specified for a query is deleted - In the Query Editor, the Show Paths button will be disabled.

Editing a query

You can edit queries added to a catalog at any time. To edit a query, in the Data tab of the Catalog Browser, right-click the query and select **Edit Query** from the shortcut menu, then in the Query Editor window, edit the query as required. For example, you can add some extra tables to the query and define join relationship among these tables.

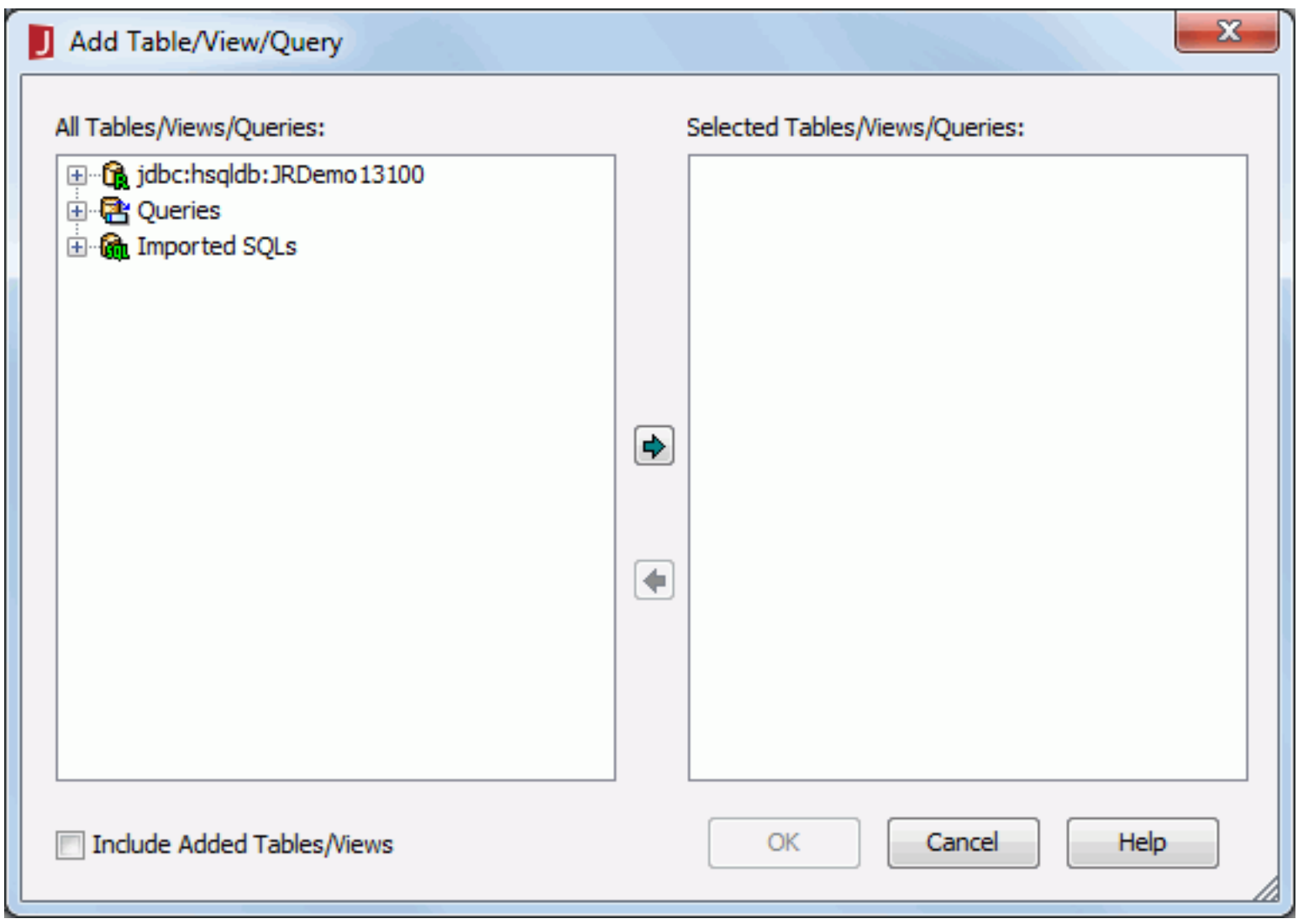
This section details how to edit an existing query as follows:

- Adding and deleting tables in a query
- Joining tables in a query
- Filtering the fields of a query
- Editing the SQL statements of a query
- Creating computed columns in a query
- Adding formula fields to a query

Adding and deleting tables in a query

- **To add some extra tables to a query:**

  1. In the Query Editor window, select **Add Table / View / Query** from the Query menu to bring up the Add Table/View/Query dialog.



  2. In the All Tables/Views/Queries box, all the tables, views, synonyms, queries, imported SQLs, stored procedures and user defined data sources in the current catalog data source in which the query is added are listed. Select the resources you want to use in the query and click to add them into the Selected Tables/Views/Queries box. You can mash-up multiple data resources in a query if you want. When two resources (for example, a table and a view) use the same name, they cannot be added to the right box at the same time, and when a table is already contained in a query, you cannot add the table and the query at the same time.

     To remove unwanted resources from the Selected Tables/Views/Queries box, select them and click .

  3. If the current catalog data source contains JDBC connections, the Include Added Tables/Views option is available. Check it if you want to show the tables, views and synonyms that have already been added to the query in the All Tables/Views/Queries box. You can then add the tables, views and synonyms of the JDBC connections to the query as many times as you want by providing different names for the tables, views and synonyms each time you add them in the Input Table Name dialog.

  4. Click **OK** to add the selected resources to the current query. When a query, imported SQL, stored procedure or user defined data source is added, it will be added as a single table with all of its columns.

  5. Click **Arrange Table / View / Query** on the Query menu to organize the tables.

  6. Select the required columns in each table. To select all, click *****. The columns will be displayed in the criteria panel of the Query Editor.

     For tables in query of a MongoDB connection, you cannot select all columns by clicking *****. The PrimaryKey and ForeignKey columns in each table cannot be selected to a query.

- **To remove tables from a query:**

  1. Select the table that you want to delete in the Query Editor window.

  2. Click **Delete Table / View / Query** on the Query menu.

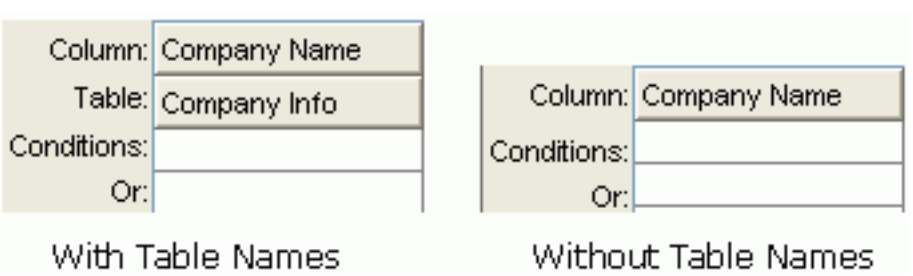
  3. Repeat the above steps to delete other tables.

Notes:

- For tables added to a query, you can decide whether or not show the full name of the fields (with table names) by checking the option Show Mapping Names on the Query menu.

- After deleting a table from a query, any joins based on the table will be moved.

- In the criteria panel, if you do not want to show the table names for the selected columns, you can uncheck Show Table Names on the Query menu.



- To delete a column in a table, select the column in the criteria panel and click **Delete Column** on the Column menu. You can also undo the deletion by finding the field in the table and placing a check mark beside it.

Joining tables in a query

You have two ways to create joins between tables in a query, automatically or manually. Any joins defined between two tables are represented as arrows connecting the key fields from the two tables. Besides joining tables in the same data connection, JReport also allows you to join tables that come from different connections and this kind of joins are called distributed joins.

Joining tables automatically

The Auto Join option on the Query menu enables you to join tables together automatically based upon the default criteria. You can choose to join tables in a query by any of these criteria. However, if you have specified to use pre-joins in your queries, the Auto Join feature will not take effect.

- **Foreign Keys**
  If there is a column that is defined as foreign key in one table and a primary key in another at the same time, these two tables will be joined together.
- **Primary Keys with Same Names**
  If a column is defined as a primary key in one table and appears in another at the same time, these two tables will be joined together.
- **Same Column Names**
  Tables with the same column name will be joined together.

Joining tables manually

To create manual joins, take one of the following two ways:

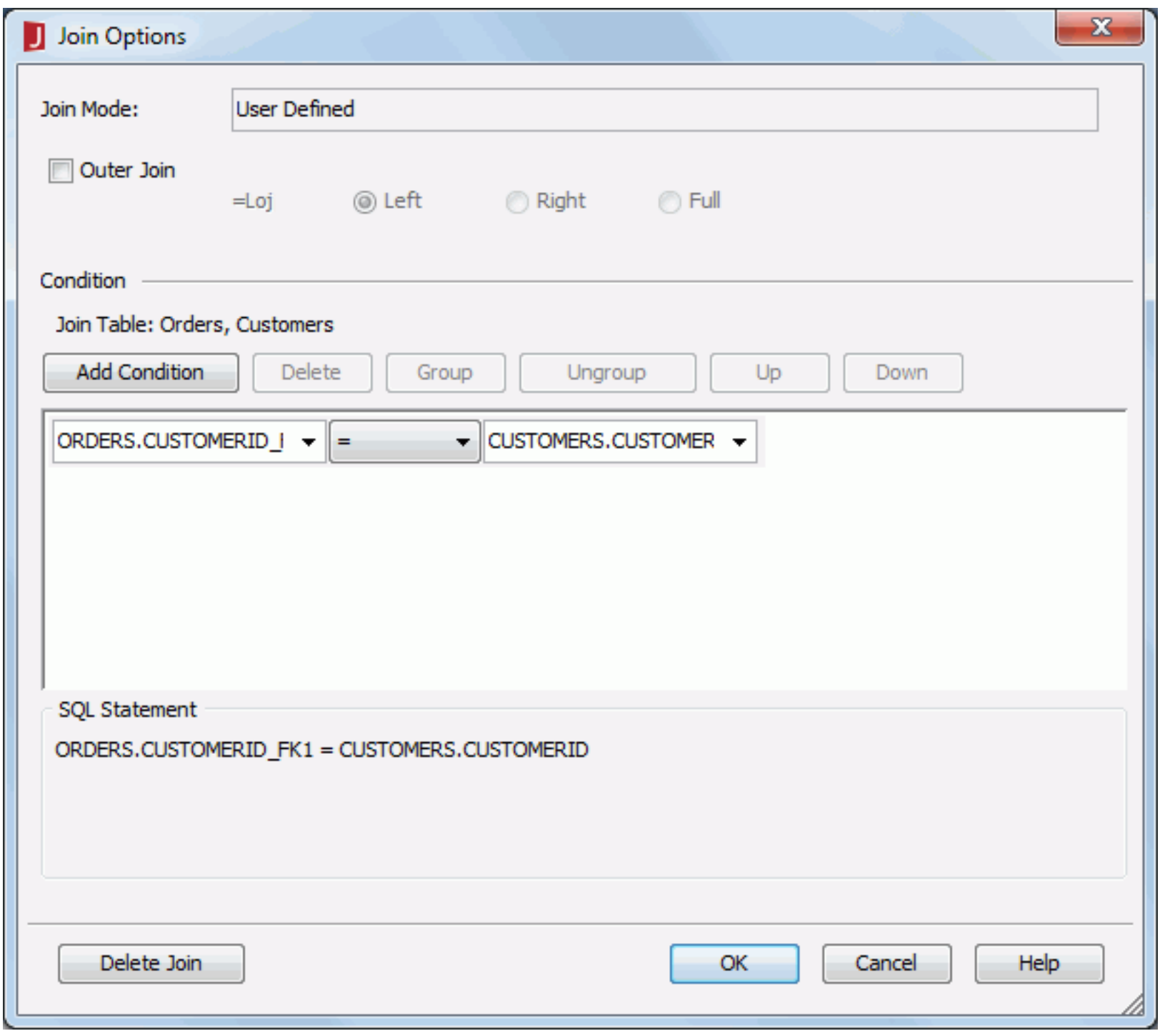
- Position the mouse pointer over the column that will be the source of the join, then click and hold the left mouse button while dragging the join away from the source column to the destination column.
- Select the columns from two tables while pressing CTRL, and then click **Query** > **Join Columns**.

The join relationship is then established between the tables. When more than one relationship is required between two tables, you can create multiple joins between them.

Editing a join

No matter whether a join is created automatically or manually, you can further edit it if required. To do this:

1. Double-click the join button in the join line. The Join Options dialog appears.



2. Check the **Outer Join** option to set the join an outer join.
3. Click either the **Left**, **Right** or **Full** radio buttons. Regardless of where the tables are placed in the Query Editor, **Left** is where the arrow starts and **Right** is where the arrow points.
4. Edit the join conditions in the Condition panel according to your requirements.
5. When done, click **OK** to accept the changes.

Notes:

- The feature of multiple joins (more than one join existing between two tables) is supported.
- In JReport Designer, the joins in one path should never form a loop (any table in this path will have direct or indirect joins with all the other tables). If you specify a path which forms a loop, JReport Designer will prompt you to re-select the joins.
- Not all forms of joins are supported by all database systems; for example, MySQL does not support Full Outer Join so be sure to check your DBMS manuals.

Alerting when Cartesian product is used

A Cartesian product is used when tables link together with no specifications.

For example, Table A has three values: A, B and C. Table B has three values: 1, 2 and 3. Value A matches value 1, value B matches value 2, and so on. This is a specific match. However, a Cartesian product could have value A matching with 1,2 and 3, and value B matching with 1, 2 and 3, and so on. Depending on the data values, Cartesian products can produce a large and complicated dataset as unnecessary information may be contained.

You can specify whether to alert when this happens as follows:

1. On the Query menu of the Query Editor, click **Current Query Options**.
2. In the Query Option dialog, check or clear Warning When Cartesian Exists.

Filtering the fields of a query

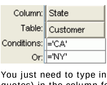
You can specify certain criteria for the fields of a query to be retrieved from the database via a filter, so that when you build reports on the query, the returned results will be narrowed down. In addition, you can also apply sub-queries to a filter. You can type your filter criteria both in the format of a filter and a QBE (Query By Example). The overall conditions will include them both. Always click the SQL button to view your conditions and parse the query to ensure the syntax is valid.

Filtering with the QBE format

The criteria panel in the lower part of the Query Editor window is for you to filter out some unnecessary records in a query. This filter is in the format of Query by Example (QBE).

When filtering with the QBE format, you are prompted to type the search criteria into a template resembling the data record. The advantage of query-by-example retrieval is that you don't need to learn a query language to frame a query. All the data fields are shown to you, and all you need to do is to enter the information that restricts the search to the required criteria. Any fields left blank, however, will match everything.

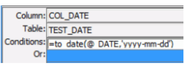
The columns and menus allow you to specifically define criteria for the fields in JReport Designer. For example, if a field is labeled REGION, and this field is a list of all 50 states in the United States. If you only want to see information from California (CA) and New York (NY), in the criteria menu, you can pick out CA and NY by placing their names in the column as seen below.



You just need to type in NY and CA. JReport Designer automatically places the ='xx' (equal sign and quotes) in the column for you. Below is a list of the syntax available:

- Comparison predicates (=, >, <, >=, <=, <>).
- BETWEEN predicate (example: BETWEEN 1 AND 100).
- IN predicate (example: NOT IN (1, 3, 5)).
- LIKE predicate (example: LIKE '%apple%').
- NULL predicate (example: IS NOT NULL).

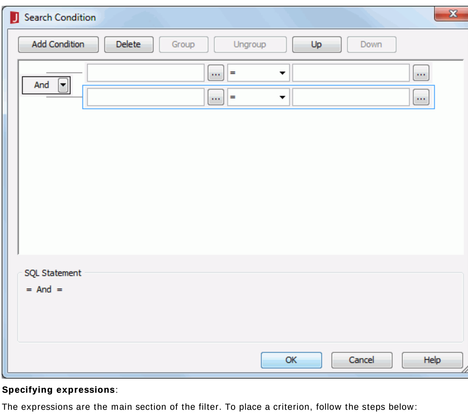
Note that in an Oracle database, if you want to use a Date or DateTime type parameter or a specific date or time to filter the field of a query, you need to apply the to_date() or to_timestamp() function in the filter condition, for example:



Filtering with the filter format

Compared with QBE, the filter provides you with more flexibility with entering your conditions. The expression includes not only the DBField selected, but also formulas and parameters. You can also manually type in strings that are supported by the database.

To filter with the filter format, select **Filter** from the Query menu. The Search Condition dialog appears.



Specifying expressions:

The expressions are the main section of the filter. To place a criterion, follow the steps below:

1. Click the **Add Condition** button to add a condition line.
2. In the field text box, specify the field to be filtered. You can either type in the name of the field manually or click ... to specify the field in the Expressions dialog.
3. From the operator drop-down list, select an operator as required.
4. In the value text box, click ... to specify the value of how to filter the field in the Expressions dialog or input the value manually. You can also use sub-queries to narrow down the result with the Subquery tab.
5. If you want to add more conditions, click the **Add Condition** button, define the conditions and then set the relationship between the condition lines as required.
6. When done, click **OK** to accept the settings.

See also Dynamically filtering queries for additional information about how parameters can be used to filter queries dynamically.

Tip: You can make use of the symbols at the bottom of the Expressions dialog to modify the expression according to your requirements.

Using sub-queries in filters

When filtering the fields of a query with the filter format, you are also enabled to use sub-queries to narrow down the result. The following are syntaxes that can be used in sub-queries.

Quantified predicate

A quantified predicate compares a value with a set of values.

- **Syntax**
```
expression----+- = --+- SOME --+-- ( subselect )
                       +- <>  -+  ANY    ---+
                       +- ! = --+  ALL   ----+
                       +- <   --+
                       +- >   --+
                       +- <=  --+
                       +- ! >  --+
                       +- >=  --+
                       +- ! < --+
```

  The subselect must specify a single result column and can return any number of values, whether they are null or not.

  - When ALL is specified, the result of the predicate is:
    - True if the result of the subselect is empty, or if the specified relationship is true for every value returned by the subselect.
    - False if the specified relationship is false for at least one value returned by the subselect.
    - Unknown if the specified relationship is not false for any values returned by the subselect and at least one comparison is unknown because of a null value.
  - When SOME or ANY is specified, the result of the predicate is:
    - True if the specified relationship is true for at least one value returned by the subselect.
    - False if the result of the subselect is empty, or if the specified relationship is false for every value returned by the subselect.
    - Unknown if the specified relationship is not true for any of the values returned by the subselect and at least one comparison is unknown because of a null value.

- **Examples**
```
SELECT qty FROM sales WHERE qty>= ALL (SELECT qty FROM sales)
SELECT BUYERID, ITEM FROM ANTIQUES WHERE PRICE != ANY (SELECT PRICE FROM ANTIQUES);
```

EXISTS predicate

The EXISTS predicate tests for the existence of certain rows.

- **Syntax**
```
- [ NOT ] EXISTS--(subselect)
```

  The subselect may specify any number of columns and,

  - The result is true only if the number of rows specified by the subselect is not zero.
  - The result is false only if the number of rows specified by the subselect is zero.
  - The result cannot be unknown.

- **Example**
```
SELECT DISTINCT pub_name FROM publishers WHERE EXISTS (SELECT * FROM titles WHERE
pub_id = publishers.pub_id AND type = 'business')
```

IN predicate

The IN predicate compares a value with a set of values.

- **Syntax**
```
expression----+-- [NOT] IN --+-- ( subselect )
```

  In the subselect form, the subselect must identify a single result column and may return any number of values, whether null or not null.

- **Example**
```
SELECT distinct pub_name FROM publishers WHERE pub_id IN (SELECT pub_id FROM titles
WHERE type = 'business')
```

The following example explains how to apply a subquery when filtering a field:

1. Create a query named mainin in the catalog, add the table Customers and select the following columns: Customers_Customer ID, Customer Name, Customers_City, and Customers_Region.
2. Click **Filter** on the Query menu to open the Search Condition dialog.
3. Click the **Add Condition** button to add a condition line.
4. Click ... beside the field text box. In the Expressions dialog, select the column **Customers_Customer ID**, then close the dialog.
5. Select **in** as the operator from the operator drop-down list.
6. Click ... beside the value text box. In the Expressions dialog, click the **Subquery** tab. Select an existing query in the catalog to be the subquery. If you want to edit the selected query, click the **Edit Subquery** button. To create a new subquery, click the **New Subquery** button.

   Here, we create a new query named subin, add the table Orders, select the column Orders_Customer ID, and add a condition "Ship Via=Express Delivery" in the Search Condition dialog.
7. Click **OK**. The subquery subin will then be added into the value text box. Click **OK** to close the Search Condition dialog.

Now, the subquery subin will be applied to the filter when you build a report that uses the Customers_Customer ID column.

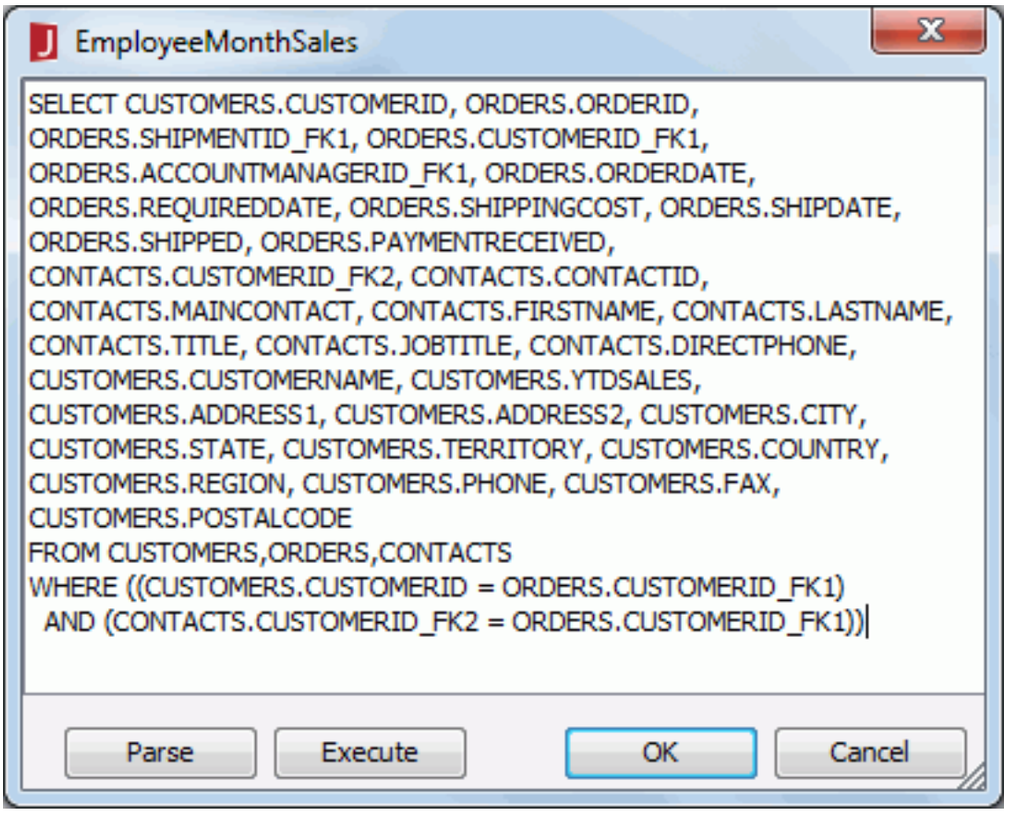
Notes:

- If a query uses a parameter in its search criteria, you can decide what the query does in case that the parameter value is NULL at runtime (for example, not provided). You can choose to remove the parameter condition from the query's criterion, or treat the parameter value as a default value (0) or an empty string, which can cause great difference in your report result.

  To remove the parameter condition from a query if this happens, check **Ignore Predicate If Parameter Value Is Null** on the Query menu.
- For string type parameters, when the value is left blank, if the Ignore Predicate If Parameter Value Is Null is checked, the value of this parameter will be considered as NULL, and this predicate will not appear in the where clause; if not checked, it will be treated as an empty string ("").
- The following SQL type of data cannot be sorted: Db.SQL_BINARY, Db.SQL_BLOB, Db.SQL_CLOB, Db.SQL_LONGVARCHAR, Db.SQL_LONGVARBINARY, Db.SQL_VARBINARY and Db.SQL_OTHER.

Editing the SQL statements of a query

In the Query Editor window, if you click the SQL button (or the View > Edit SQL menu command), a window will then be displayed showing the SQL statements used to execute the query like below.



You can edit the query statements in the text area as follows:

- To check whether it can be parsed back to the structured PSQL query correctly, click **Parse**. Whether the query statement is valid or not, a message box will be displayed to show you the result.
- To see whether it can be successfully processed, click **Execute**.
- To accept the changes you have made, click **OK**. JReport Designer will then parse and execute the query statements and dispose the Query Editor window.

Notes:

- If there are parameters in this query, their default values will be used to construct statement.
- If JReport Designer fails in parsing or restoring the query statements, a message box will be displayed asking you whether to apply the changes anyway. If you choose yes, a statement string will then be recorded in the PSQL query. However, after that, the structured statement may be not tally with the statement string.

Customized SQL mode

When creating a query using the Query Editor, JReport allows you to copy and paste your own SQL statement directly in the SQL window as explained above. However, because JReport query parser follows the SQL 92 standard, sometimes JReport is not able to parse the customized SQL statement. In this case, a warning message is displayed after you click the OK button in the SQL window asking whether to continue using the customized SQL statement. If you choose Yes, the customized SQL mode is enabled, in which all the functions in the Query Editor will be disabled, and you can only edit the query by modifying the SQL statement in the SQL window.

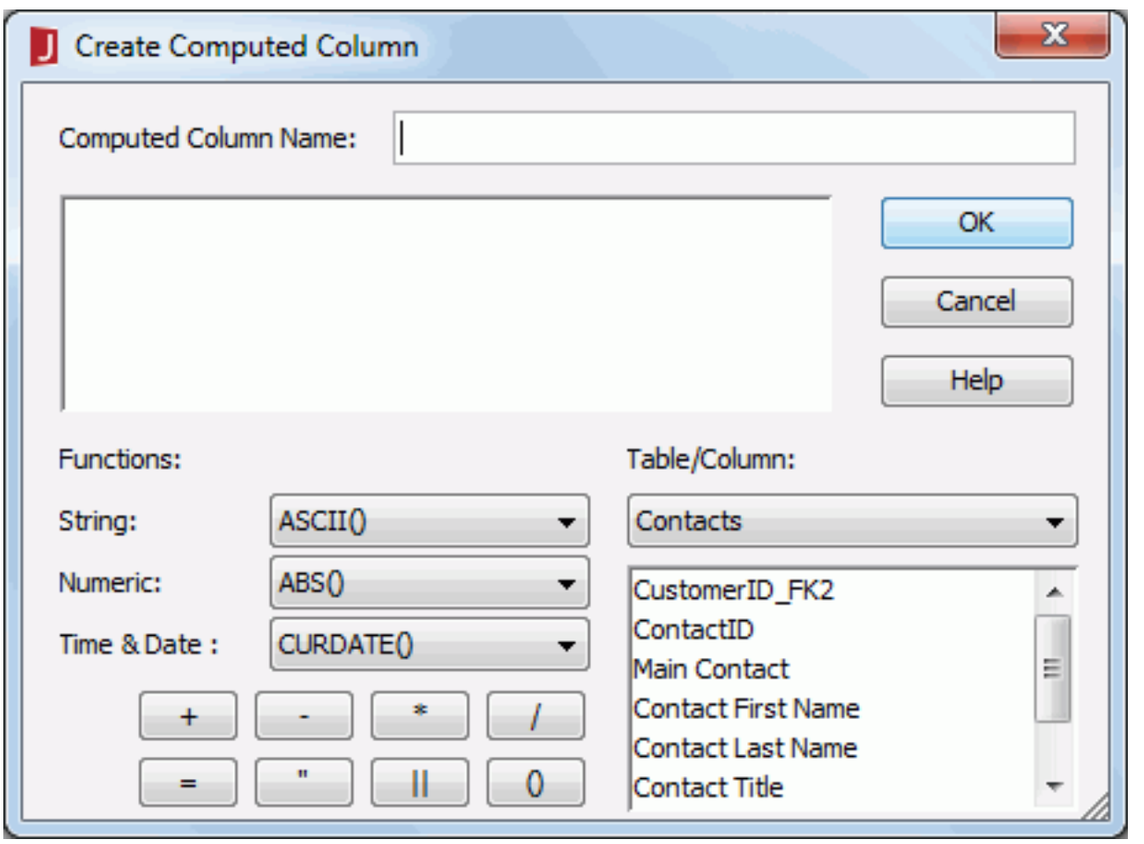
During the customized SQL mode, you can see the Customized SQL Mode option in the Query menu of the Query Editor is enabled and checked. By unchecking this option, you can go back to use functions of the Query Editor, in which case, the customized SQL statement will be lost and the SQL statement generated by JReport will be applied instead.

Note that you are required to guarantee the correctness of the customized SQL statement since JReport will not parse it.

Creating computed columns in a query

To create a computed column in a query, follow the steps below:

1. In the Query Editor window, select **Create Computed Column** from the Column menu. The Create Computed Column dialog appears.



2. Input a name for the column in the Computed Column Name text box.
3. Compose your functions for the column.
4. When done, click the **OK** button to create the computed column.

Notes:

- In the lower part of the Create Computed Column dialog, you can see some functions and tables/ columns of the query. These are just for your reference. You can specify the expression by yourself in the editing text box, only if the expression can be accepted by your database.
- The functions in this dialog are not from the JReport system. They are from the database you are connected to.
- If you change your data source, some of these functions may no longer exist. For each database, a different set of supported functions will be returned. The following functions will help you with writing an expression.
  - **+**: Add numbers or fields together in the Expression menu.
  - **-**: Subtract numbers or fields together in the Expression menu.
  - *****: Multiply numbers or fields together in the Expression menu.
  - **/**: Divide numbers or fields together in the Expression menu.
  - **=**: Equate fields together.
  - **"**: Place quotations on long character strings or names that have blanks in them (example: place quotes on values such as "New York" or "Washington DC").
  - **||**: Place fields together in the same Expression menu. (Example: "New York" || "Washington DC").
  - **()**: Place your fields in parentheses.

Adding formula fields to a query

Besides the table columns, you can also add some formula fields to a query.

To add a formula field to a query:

1. In the Query Editor window, click **Column** > **Add Formula Field** on the menu bar.
2. In the Add Formula Field dialog, choose the required formula in the Formulas box and then click **Add**. The formula will then be placed into the criteria panel in the lower part of the Query Editor, which means that you can reference the formula in the query.
3. Repeat the above step until you have added all the required formula fields, and then click **Close**. The formulas are added to the current query.

After a formula field has been added to a query, if you want to replace it with another one, you can double-click its name in the criteria panel, and then choose the required field in the Replace Formula dialog.

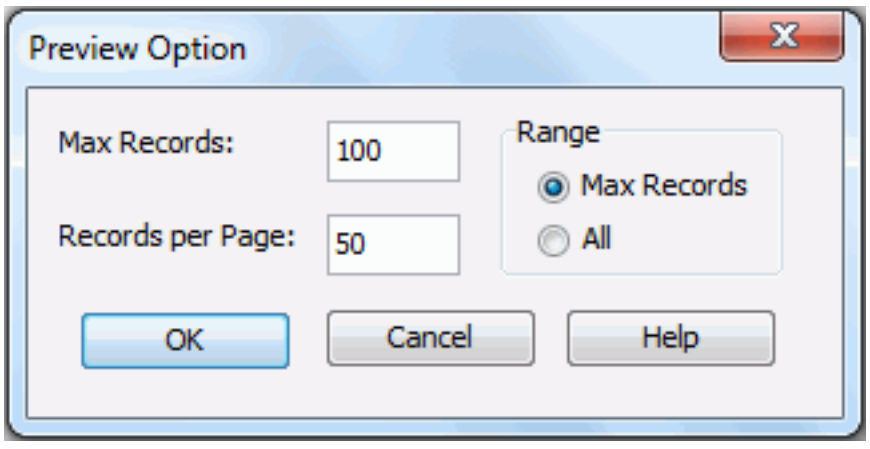
Notes:

- When you add a formula field, if the table that contains the field doesn't exist in the query, this table will be automatically added to the query.
- You are not able to edit a formula in the Add Formula Field dialog.

Previewing a query

To preview a query, follow the steps below:

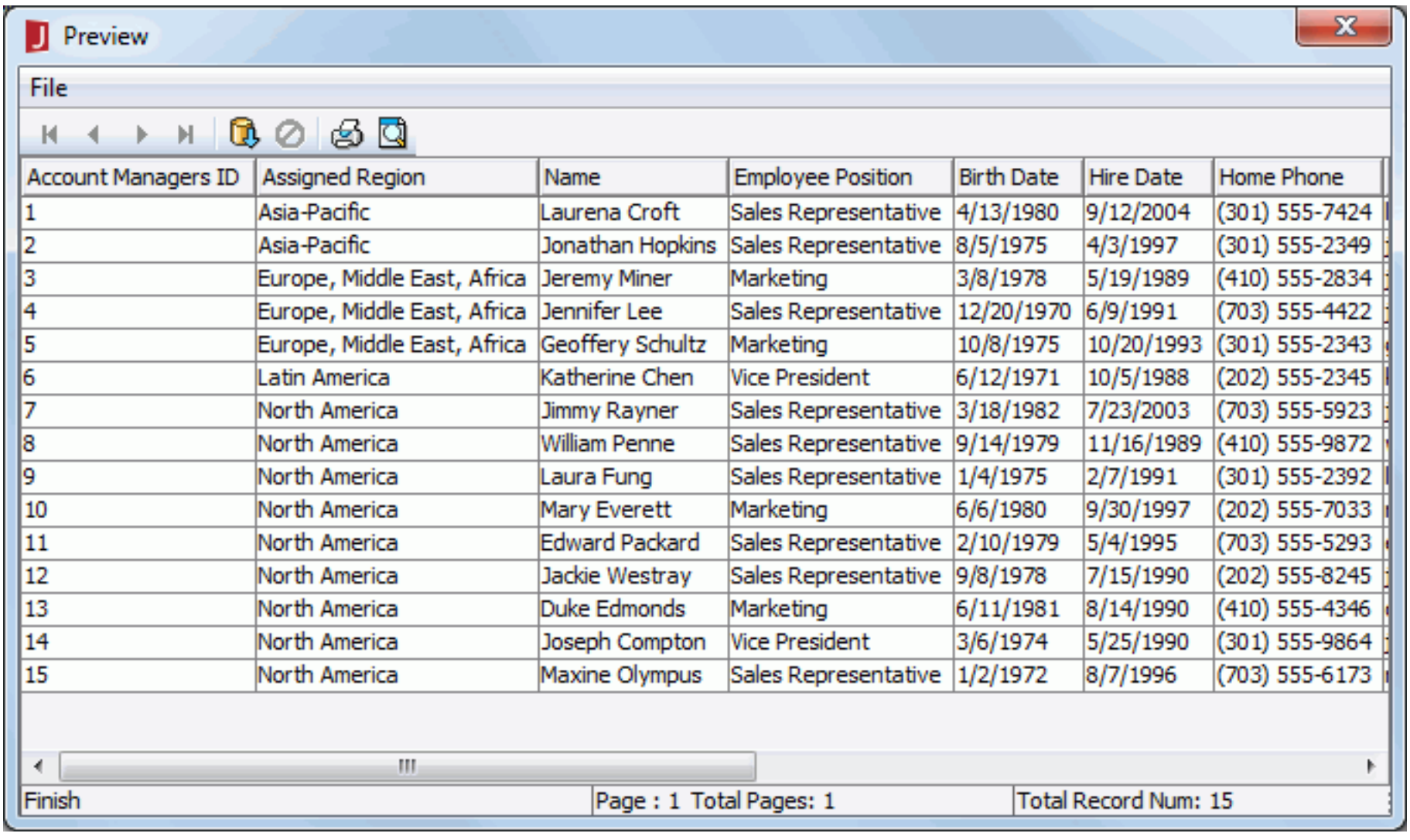
1. In the Query Editor window, click the **Preview** button on the toolbar. The Preview Option dialog appears.



2. Specify the options in the dialog to suit your requirements.

   Tip: The values shown in the text boxes as soon as the Preview Option dialog appears are the default values. These values can be specified in the Query Editor category in the Options dialog.

3. Click **OK**. The Preview window will then appear.



Account Managers ID	Assigned Region	Name	Employee Position	Birth Date	Hire Date	Home Phone
1	Asia-Pacific	Laurena Croft	Sales Representative	4/13/1980	9/12/2004	(301) 555-7424
2	Asia-Pacific	Jonathan Hopkins	Sales Representative	8/5/1975	4/3/1997	(301) 555-2349
3	Europe, Middle East, Africa	Jeremy Miner	Marketing	3/8/1978	5/19/1989	(410) 555-2834
4	Europe, Middle East, Africa	Jennifer Lee	Sales Representative	12/20/1970	6/9/1991	(703) 555-4422
5	Europe, Middle East, Africa	Geoffery Schultz	Marketing	10/8/1975	10/20/1993	(301) 555-2343
6	Latin America	Katherine Chen	Vice President	6/12/1971	10/5/1988	(202) 555-2345
7	North America	Jimmy Rayner	Sales Representative	3/18/1982	7/23/2003	(703) 555-5923
8	North America	William Penne	Sales Representative	9/14/1979	11/16/1989	(410) 555-9872
9	North America	Laura Fung	Sales Representative	1/4/1975	2/7/1991	(301) 555-2392
10	North America	Mary Everett	Marketing	6/6/1980	9/30/1997	(202) 555-7033
11	North America	Edward Packard	Sales Representative	2/10/1979	5/4/1995	(703) 555-5293
12	North America	Jackie Westray	Sales Representative	9/8/1978	7/15/1990	(202) 555-8245
13	North America	Duke Edmonds	Marketing	6/11/1981	8/14/1990	(410) 555-4346
14	North America	Joseph Compton	Vice President	3/6/1974	5/25/1990	(301) 555-9864
15	North America	Maxine Olympus	Sales Representative	1/2/1972	8/7/1996	(703) 555-6173

   Note: If the query has parameters, when you click the OK button, the Enter Parameter Values dialog will be displayed. After you enter the parameter value, the Preview window will appear showing the corresponding records.

4. Click , , and to browse the records. If the type of the result set is TYPE_FORWARD_ONLY, the last page button will be disabled until you have browsed the last page.

5. To refetch the result set, click the **Refetch** button on the toolbar. Click the **Stop** button to stop JReport from refetching the result set is required.

6. To print the result set, click the **Print** button. If you want to preview the result set before printing, click the **Print Preview** button.

7. If you want to set up the page parameters, click **File** > **Page Setup**.

8. Click **File** > **Close**, or the close button of the window to close the Preview window.

Note: In the Preview window, JTable is used to display the result set. When you open the window, the result set is cached, and when you close the window, the result set is released.

Creating and modifying a union query

The Union option in the Query menu of the Query Editor enables you to combine specified records from more than one query into a result set.

To create a union query:

1. Right-click on a selected query (here we call it primary query) and select **Edit Query**.
2. In the Query Editor, select **Union** from the Query menu, and then click **Create**.
3. Specify a name for the query in the Input Query Name dialog, and then click **OK**.
4. A new Query Editor will be opened. Create the query as required in the editor.

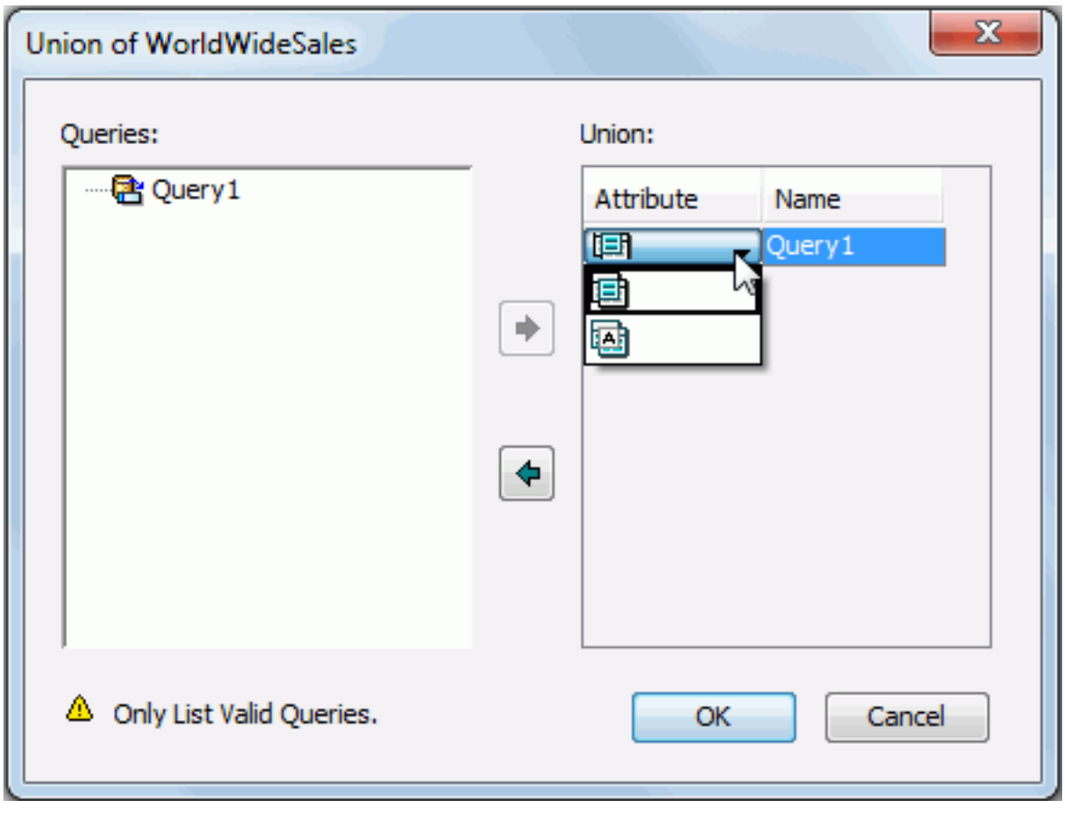
Notes:

- When creating a union query, note that the selecting order should refer to the order in the primary query. The SQL type and the number of the selected columns should match those in the primary query. For example, if you select two columns in the primary query, the first one is of Integer type, and the second one is of String type. Then, in the union query, you should also first select an Integer column, and then a String type column.
- A union query cannot support formulas and parameters.

After a union query is set up, you can also modify it at any time.

To modify a union query:

1. Right-click on the primary query and select **Edit Query**.
2. In the Query Editor, click **Query** > **Union** > **Select** to bring up the Union dialog.



3. To add a union to the query, select a union from the Queries box, and then click . To remove a union from a query, select a union from the Union box, and then click .
4. In the Union box of the dialog, select the Attribute column to specify the type of the union.
   - : No duplicate records will be returned.
   - : All records will be returned.
5. When done, click **OK** to confirm the changes.

Locking queries

Normally in JReport Designer, queries are free to be shared among reports, which could increase the potential risk of being changed unexpectedly. When a query is shared by more than one report, it can be changed by any of those reports, which means you must be very careful to avoid causing impacts to other reports when you modify the shared query for one report. However, JReport provides you with the flexibility to lock a query to prevent it from being changed by unauthorized person.

Creating a user to a query

Before you can lock or share a query, you need to claim the query by creating a user to it.

To creating a user to a query:

1. Right-click the query you want to claim and select **Create User** from the shortcut menu.
2. In the Create Query's Username and Password dialog, type in the user name and password.
3. Click **OK** to confirm the information.

When a user has been created for a query, you can modify the user information or remove the user as required.

- To modify the user information, right-click the query and select **Modify User** from the shortcut menu. In the Modify User Information dialog, provide the old user name and password and click **OK**. Then, in the Modify Query's Username and Password dialog, type in the new user name and password as required and click **OK**.
- To remove a user, right-click the query and click **Delete User** on the shortcut menu. In the Remove User dialog, provide the user and password information and click **OK**. The query is then free to be shared.

Note: A query can be claimed by only one user, that is to say, you can create only one user to a query.

Locking or sharing a query

Using the Share property of a query, you can lock or unlock it according to your requirements. By default, the Share property of a query is true, which means that all queries in a catalog can be shared among multiple reports, until you lock them.

To lock or unlock a query:

1. Select the query that you want to lock or unlock.
2. Click the **Expand** button on the Catalog Browser toolbar to show the Properties sheet.
3. Locate the Share property and set its property to true or false. If it is true, the query can be shared among reports and can be modified. Otherwise, the query is locked and can be neither used in new reports nor modified by unauthorized person.

   Make sure that you have enabled the editable mode for the Properties sheet before editing the property values, that is to say, you have unchecked the option "Forbid editing data object properties" in the Options dialog.

You can also lock or unlock a query in the following way:

1. Right-click the query that has already been claimed by a user and select **Set Share Property** from the shortcut menu.
2. In the Set Query's Share Property dialog, provide the user information and check or clear the Share option as required.
3. Click **OK** to confirm the change.

Notes:

- If a query has not yet been claimed by any user, when you set its Share property in the Properties sheet, you will be prompted to claim the query by creating a user to it.
- If a query is locked, it is password protected. Unless you provide valid user information, you will neither be able to create a report directly using that query nor make changes to the query, including setting its Share property.

•

Developing reports from queries

JReport Designer helps you build various professional reports based on queries. When building reports on queries, you can place certain criteria or restrictions on the data in order to extract only the required data from the database.

Note that sometimes after you have designed a report, you may want to use another query to run it. Instead of building another report from scratch, JReport Designer allows you to change the query name directly in the Report Inspector. In which case, you should make sure that the new query uses the same tables as the original one. You are not recommended to use this method until you are familiar with JReport Designer.

To develop a report from a query:

1. Make sure the catalog where the query locates is opened, then click the **New Page Report** button on the Standard toolbar.
2. In the New Page Report dialog, specify the title and layout of the first report tab in the report as required, then click **OK**.
3. In the Data screen of the report wizard, choose the query with which you want to build the report tab.
4. Follow the report wizard to create the report tab.

Dynamic queries

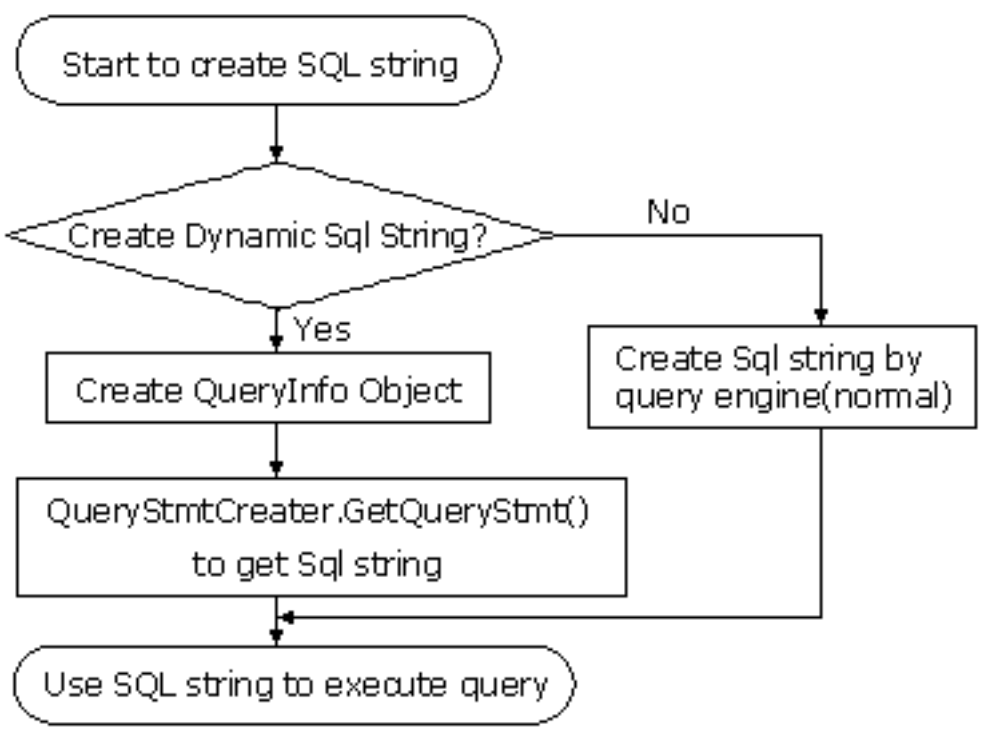
You can build queries using the Query Editor or from an imported external SQL file. Either way, the queries should be predefined. However, sometimes you may want to specify the query at runtime. For example, in the catalog, you build the query "select customers_id from customers", but at runtime, you may want to fetch data from another table - customers1. The query should be updated according to the table index, such as "select customers_id from customers1". JReport provides a new feature called dynamic query. With this, queries can be dynamically generated, which allow you to fetch data from different tables at runtime.

In this section, you will be shown how to make use of dynamic queries via the dynamic query interface.

The dynamic query interface

The dynamic query interface SQLStmtCreator is stored in the archive file - JREngine.jar in `<install_root>\lib`. It is contained in the package toolkit.db.api, and can be applied to any existing query in a catalog.

The following flowchart illustrates how the interface works when creating a dynamic query. The QueryInfo object is passed from the JReport Engine to the interface as an input. Then, the completed SQL statement is returned from the interface. Finally, the completed SQL statement is sent to the database to get the result set for the report.



Method of the dynamic query interface

This interface is very simple with only one method: getSQLStmt(QueryInfo queryInfo);

It receives information of a query and returns an SQL string. QueryInfo is a container that contains all information to build an SQL string. Users can call getXXX() methods to get all information step by step.

The structure of QueryInfo is as follows:

- **ConnectionInfo**
  Driver, URL, User, Password, DateFormat, TimeFormat, TimestampFormat, TransactionIsolation level, ReadOnly, QualifiedNamePattern, ExtraNamePattern, EncodingPattern
- **Column array**
  All the selected columns. Elements in this array are ColumnInfo object. ColumnInfo contains Mapping Name, Real Name, Table Info and Expression (if this is a computed column).
- **Tables(array)**
  Selected tables. Elements in this array are TableInfo objects. TableInfo contains Mapping Name, Real Name, Correlation Name, Schema, and Catalog.
- **Joins(array)**
  Elements in this arrays are JoinInfo objects. JoinInfo includes Column from, Column to, Operator, and Join type.
- **QBEs(array)**
  Part of the where condition. It is retrieved from the query builder. Elements in this array are QBEInfo objects. QBEInfo contains ColumnInfo with QBE condition bound to this column.
- **Ands(array)**
  Part of the where condition. It is retrieved from the advanced search condition in the query builder. Elements in the array are AndInfo objects. AndInfo contains Left expression, Operator, Right expression, and Logic.
- **SubLinks(array)**
  If the dynamic query is for a subreport, the SubLinks is used for the additional WHERE clause to filter data, and this is the way to link the dynamic query to the primary report. Elements in this array are composed of SubLinksInfo objects. SubLinkInfo contains Column, Operator and Value.
- **Parameters (array)**
  Parameters used for creating a query. Users are required to encode parameters that the database can recognize. The elements in this array are ParameterInfo objects. ParameterInfo contains Name, Type and Value.
- **OrderBys(array)**
  Elements in the array are OrderByInfo objects. OrderByInfo contains Column and sorting direction.
- **Other Information**
  Query name, IsDistinct, and WherePortionString. The value of WherePortionString is set via JRengine. setWherePortionString().

Reference: See JReport Javadoc toolkit.db.api.SQLStmtCreator interface in `<install_root>\help\api`.

Using dynamic queries

Before you can use dynamic queries, you are required to first make some preparations.

1. Compile the Java file that you write and append the path of the compiled Java file with a valid path to the ADDCLASSPATH variable of setenv.bat.
2. Set the value of the connection property SQL Statement Creator.

   The dynamic query interface is set as a property in the JDBC connection object in a catalog.

   In JReport Designer, launch the Catalog Browser, expand the data source node, select the JDBC connection, right-click it and select **Properties** from the shortcut menu to display the properties of the connection. You will then see the SQL Statement Creator property, which is used to set the real class name of the dynamic query object.

   Here are a couple of examples:

   - You implemented the interface by the class UserSQLStmtCreatorImpl, then input the property value as UserSQLStmtCreatorImpl;@para1;@para2. Para1 and Para2 are parameters used to specify criteria while creating the query.
   - School year which changes at runtime, requires to be inserted into the report template so that the input value for the runtime parameters can be passed to the dynamic query interface via QueryInfo object, in order for a corresponding SQL statement to be returned.

     Notes:

     - Binding parameters is not supported by this feature.
     - By default, you are not permitted to change the property values in the Catalog Browser. To make the property values editable, on the Catalog Browser toolbar, click the **Options** button . In the Options dialog, select **Catalog** in the **Category** box and uncheck **forbid editing data object properties**. Then, go back to the Catalog Browser, and highlight the column name that you want to edit to change or set the property values for.
3. Set the value of the query property Enable SQL Statement Creator.

   In the Catalog Browser, in the Properties sheet of a highlighted query, there is a property named Enable SQL Statement Creator, which indicates whether or not the query uses the dynamic query interface to get the result set. When it is set to true, the query can be re-generated at runtime using the dynamic query interface.

An illustrative example

You are provided with a demo program, SQLStmtCreatorImpl.java in `<install_root>\help\samples\APIDynamicQuery`, which implements the dynamic query interface. This demo is for changing the table name of the query sent to the database. Specifically, when you run the report, if you enter 1 as the tableIndex parameter, the query will dynamically change to Customers1. If you do not enter anything, you will get the result set from the Customers table.

The following example explains how to compile the required files and use dynamic queries in a report:

1. Compile SQLStmtCreatorImpl.java.

   Assume that JReport Designer has been installed in `C:\JReport\Designer`, and the class files of the MappingNameFinder.java are in `C:\JReport\Designer\help\samples\APIDynamicQuery`:

   ```
   javac -classpath c:\jreport\designer\lib\JREngine.jar;c:\jreport\designer\help
   \samples\APIDynamicQuery SQLStmtCreatorImpl.java
   ```

   Note: To compile SQLStmtCreatorImpl.java you will need another file MappingNameFinder.java in `<install_root>\help\samples\APIDynamicQuery`.
2. Modify the batch file setenv.bat in `<install_root>\bin` by appending the SQLStmtCreatorImpl.java path into the batch file's ADDCLASSPATH variable:

   `set ADDCLASSPATH=%JAVAHOME%\lib\tools.jar;`**`C:\jreport\designer\help\samples\APIDynamicQuery;`**
3. Start up JReport Designer.
4. Click **File** > **Open**. In the Open Report dialog, click the **Browse** button to open the catalog file **SampleComponents.cat** in `<install_root>\Demo\Reports\SampleComponents`, then open the sample report **BandedObjectReport.cls**.
5. Launch the Catalog Browser, expand the **Data Source 1** node, right-click the **Parameters** node, click **Add Parameter** on the shortcut menu, then create a type-in parameter of String type named tableIndex (leave the other settings to their default).
6. Expand Data Source 1 and then select the JDBC connection.
7. Unfold the property list of the connection by clicking on the toolbar, then set the value of the SQL Statement Creator property as **SQLStmtCreatorImpl;@tableIndex**. The parameter tableIndex is used to specify which table is to be selected at runtime.
8. Select the query **QueryForBandedObject** in the Catalog Browser, and set the value of its property Enable SQL Statement Creator to **true**.
9. Save the catalog and view this report.
10. The Enter Parameter Values dialog appears. When you input 1 as the value of the tableIndex, the report will run on the table Customers1, and when you input nothing as the value of the tableIndex, the report will run on the table Customers.

Query modifiers

Query modifiers, also called WHERE portions in JReport, are the WHERE clauses in SQL select statements. In JReport, the SQL select statements are called queries. Both queries and query modifiers are stored in the catalog, and query modifiers are saved in the catalog independently (not related to queries). When a report is running, JReport will construct the SQL statement based on the query used in the report and may substitute the WHERE clause in the SQL statement if there is a given query modifier.

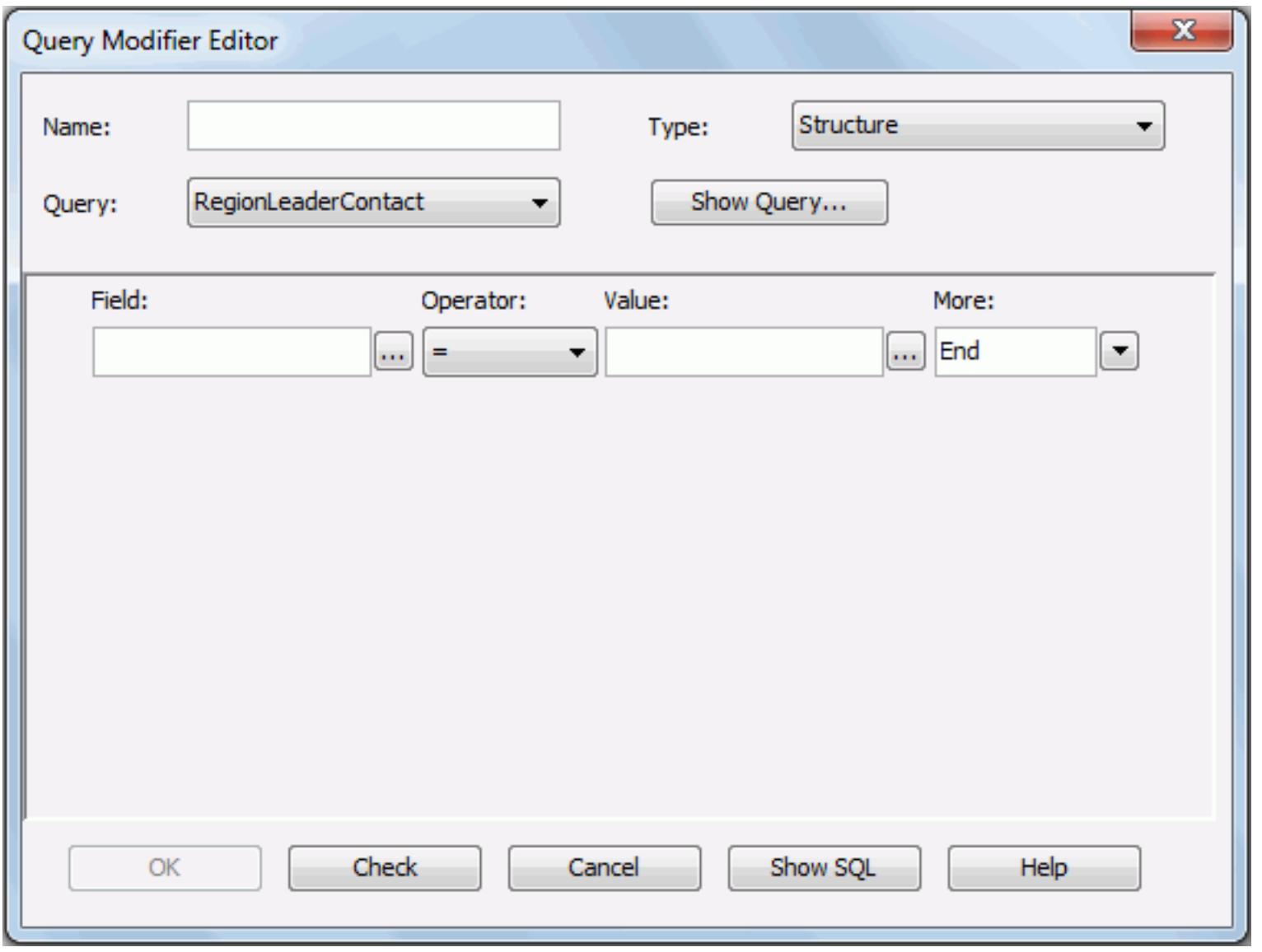
The query modifier can be stored in the catalog in two formats: structure and string. In the structure format, the groups of conditions are stored. The query modifier consists of many condition groups and the groups are connected by a logical operator. Each condition consists of many conditions and the conditions are also connected by a logical operator. Each condition consists of a left expression, the relationship operator and a right expression. In string format, the text of the query modifier is stored.

Before running a report, you can specify a query modifier to the JReport system via the Engine Bean, so that you can generate the report on different data.

Adding a query modifier

To add a query modifier, follow the steps below:

1. In JReport Designer, open the catalog to which you want to add the query modifier.
2. In the Data tab of the Catalog Browser, expand the desired data source and then the **Queries** node.
3. Right-click the **Query Modifiers** node and select **Add Query Modifier** from the shortcut menu. The Query Modifier Editor dialog appears.



4. Enter a name for the query modifier in the Name text box and specify its type in the Type drop-down list, Structure or String.
5. Select a query from the Query drop-down list to build the query modifier. You can click the **Show Query** button to view the query if required.
6. In the editing area, edit the query modifier as required.

   If String type has been selected, a text editor will be shown where you can key in the WHERE clause. If the Structure type has been selected, a condition editor will be displayed where you can construct the query modifier by clicking buttons (for details about how to set the condition, see Filtering with the filter format).
7. Click the **Check** button to check the validity of the query modifier according to the query.

   The validity of a query modifier is defined as: all the fields and formulas referred to by the query modifier are available to the chosen query. This validation check also ensures that the query modifier will work with the specified query.
8. Click the **Show SQL** button to view the string of the query modifier.
9. When done, click **OK.**

Cached query results

By default, when you run a report, JReport Engine fetches data from the database using the JDBC driver. For reports that are built on queries (and imported SQL files, stored procedures and UDSs which function as queries), JReport Designer enables you to create cached result files and save them somewhere in your machine. Then, when you view these reports, you can choose to use the data from the cached query result file as opposed to the database. This is very useful for working on the design of a report while you are off-line and do not have a DBMS connection. Additionally, it is helpful when you need to send data and reports to Jinfonet Support for assistance.

Creating a cached query result

JReport Designer provides you with two ways to create cached query results, the first is with the Catalog Browser panel and the second is with the Resource View panel.

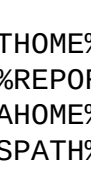
- **To create a cached query result file in the Catalog Browser panel:**
  1. In the Data tab of the Catalog Browser, select the data object for which you intend to create the cached query result file.
  2. Right-click the data object, and select **Create Cached Query Result** from the shortcut menu.
  3. In the Save Cached Query Result dialog, specify the file name with or without an extension and the folder to where you want to save the result file.
  4. Click **OK** and the cached query result file will be saved to the specified folder.
- **To create a cached query result file in the Resource View panel:**
  1. Open a report, in the Resource View panel, right-click the query on which the report is built and select **Create Cached Query Result** from the shortcut menu.
  2. In the Save Cached Query Result dialog, specify the file name with or without an extension and the folder to where you want to save the result file and then click **OK**.

Using a cached query result to run a report

By setting the Data Driver property appropriately, you can make a report run using cached query result. However, for web reports, in order to make the property shown in the Report Inspector, you need to add the parameter *-pd* to the file JReport.bat which is located in `<install_root>\bin` as follows before starting JReport Designer:

```
Set CLASSPATH=%REPORTHOME%\lib\SwingDesigner.jar;%REPORTHOME%\lib\report.jar;%REPORTHOME
%\lib\resource_en_US.jar;%REPORTHOME%\lib\JREngine.jar;%REPORTHOME%\lib\JRResultViewer.
jar;%REPORTHOME%\lib\*;%REPORTHOME%;%ADDCLASSPATH%"%JAVAHOME%\bin\java.exe" -Xms40m -
Xmx512m "-Dinstall.root=%REPORTHOME%" -classpath "%CLASSPATH%" -Dreporthome="%REPORTHOME
%" -Djreport.url.encoding="UTF-8" com.jinfonet.designer.JReport -pd %1 %2 %3 %4 %5 %6 %
7 %8 %9
```

To make a page report or web report run using cached query result:

1. Open the report, in the Report Inspector, click the **Up** button  on the Report Inspector toolbar to reveal the higher levels of the report structure.

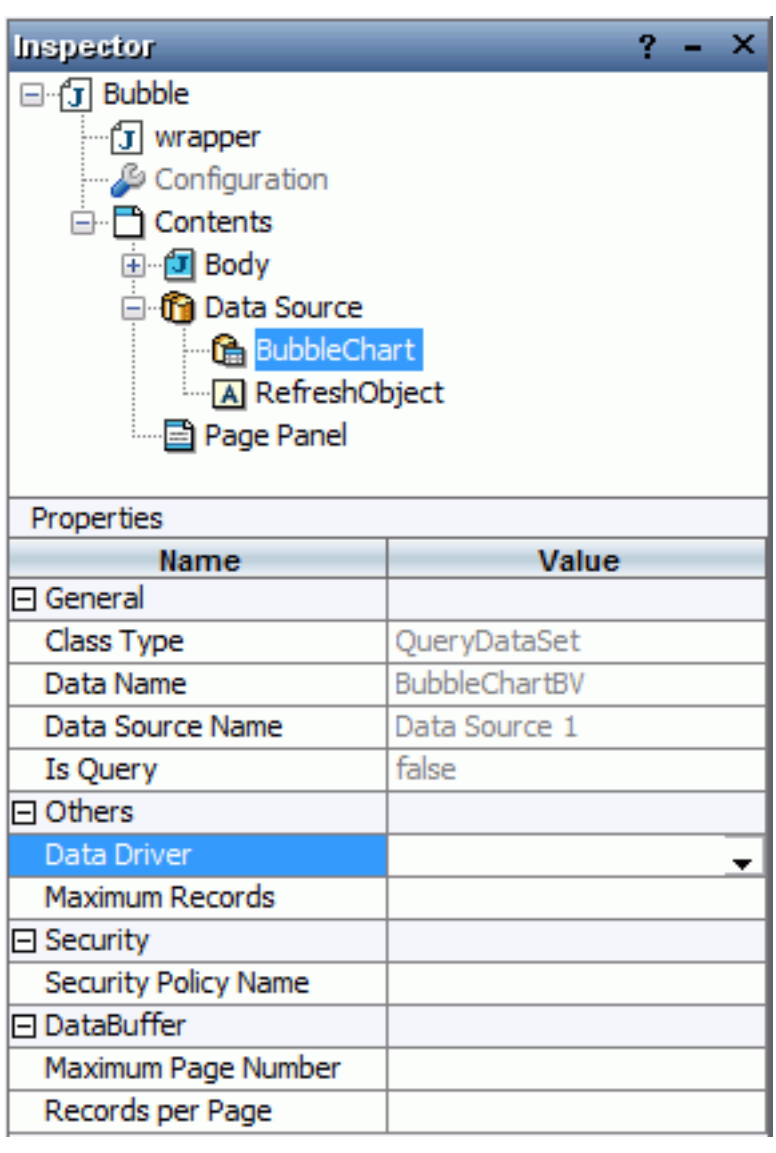
   Note: This step is required for both page reports and web reports when *-pd* is added. For page reports when *-pd* is not added, ignore the step.
2. Find the Datasets node, then select a dataset used in the report and specify its Data Driver property in the Properties panel as follows:

   `jrquery://jet.universe.resultfile.UResultFileResultSet;Full path of the cached query result`

   For example, if the cached query result has been saved to `C:\JReport\Designer\Cached` with the file name test, the property value should be `jrquery://jet.universe.resultfile.UResultFileResultSet;C:\JReport\Designer\Cached\test`
3. Repeat the above step to set the Data Driver property for other datasets the report uses.
4. Click the **View** tab to view the report. You will now see that the report data has been retrieved from the cached query result file.

To make a library component run using cached query result:

1. Open the library component, in the Report Inspector, select the node that represents the dataset used by the library component under the Data Source node (the node name is LCDataSource when *-pd* is added) in the report structure tree.
2. Specify the Data Driver property in the Properties panel in the above format.

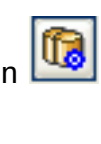


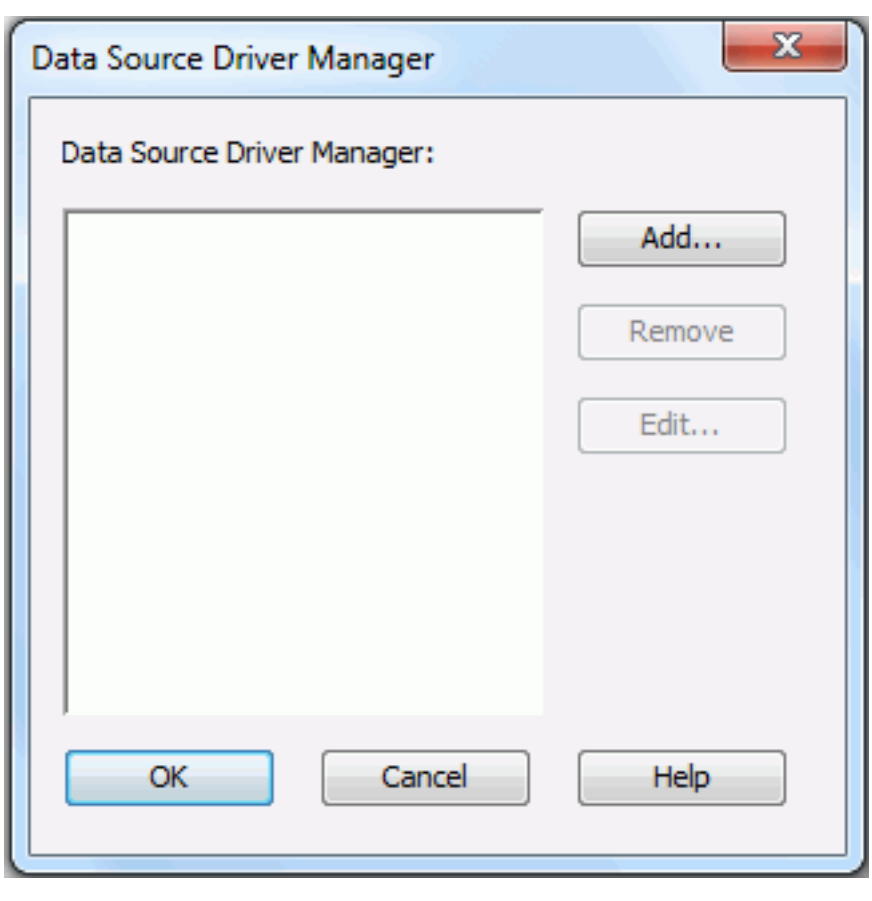
Note: If you check the option "Use cached query result" in the Catalog category of the Options dialog, whenever you view a report, a dialog will prompt you to choose a cached query result to run the report.

Managing cached query results

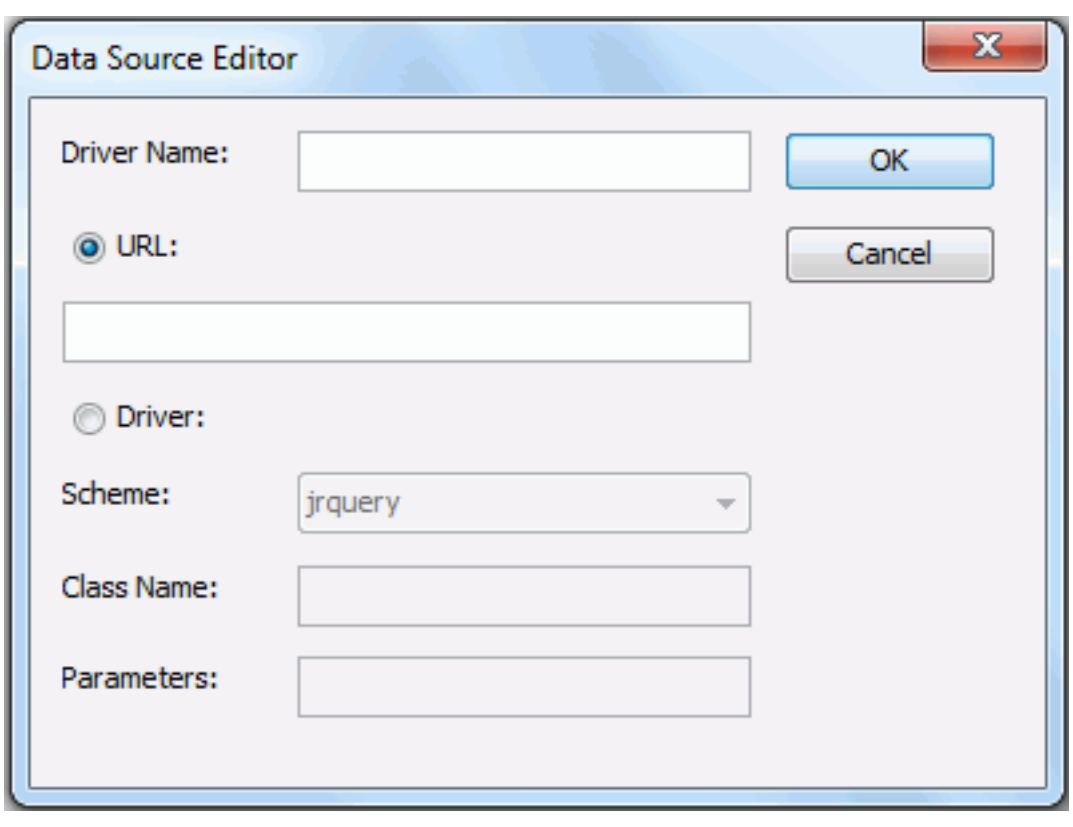
You may find it inconvenient to input a property value that is long and complicated. To help you with this, JReport Designer provides a tool for managing your cached query results, the Data Source Driver Manager.

Assume that a cached query result file orderstat_cached for the query OrderStat in the catalog file SampleReports.cat has been created and saved in `C:\JReport\Designer\Cached`. To use the Data Source Driver Manager, follow the steps below:

1. Click **File** > **Catalog Management** > **Open Catalog** in JReport Designer.
2. In the Open Catalog File dialog, select to open the catalog file SampleReports.cat.
3. In the Catalog Browser, click the **Data Source Driver Manager** button  on the toolbar to open the Data Source Driver Manager dialog.



4. Click the **Add** button, and the Data Source Editor dialog appears.



5. Input a name for the driver in the Driver Name box. Here, we name it **OrderStat**.
6. Check the **URL** radio button and input the URL as follows:

   `jrquery://jet.universe.resultfile.UResultFileResultSet;Full path of the cached query result`

   In the example, the URL is `jrquery://jet.universe.resultfile.UResultFileResultSet;C:\JReport\Designer\Cached\orderstat_cached`

   Note: If you check the Driver radio button to specify the driver, you will need to input the class name and parameter of the driver as follows:

   - **Class Name**: jet.universe.resultfile.UResultFileResultSet
   - **Parameter**: Full path of the cached query result (`C:\JReport\Designer\Cached\orderstat_cached` in this example)

   Then, a URL will be generated according to your input information.
7. Click **OK** to add the driver to the Data Source Driver Manager.
8. Repeat the above steps to add more drivers to the manager.
9. In the Data Source Driver Manager dialog, all the added drivers will be listed in the Data Source Driver Manager box. If you want to remove a driver, select the driver and click the **Remove** button. To edit a driver, select the driver and click the **Edit** button.
10. Click **OK** to close the dialog.

Upon running a report using a cached query result, you can choose an appropriate driver from the Data Driver property's drop-down list directly without inputting the long value manually.

Data Manager

The Data Manager in JReport Designer allows you to control the data retrieval of your queries, including the number of rows to be displayed and the duration required for the retrieval. It can also keep access information from previous runs of a query (including imported SQL, stored procedure and UDS which function as queries).

This topic shows the usage of the Data Manager as follows:

Limiting the query run time and number of records

There may be many records in a table on which you are to build a report. After you have generated the query and designed the rough report, you may only want to preview several lines of records for testing, in which case, you can take advantage of the two query properties - Maximum Rows and Maximum Duration which allow you to limit the run time and number of records, saving time and resources. Taking InvoiceReport.cls as an example:

1. Click **File** > **Open** in JReport Designer.
2. In the Open Report dialog, click the **Browse** button to open the catalog file **SampleReports.cat** in `<install_root>\Demo\Reports\SampleReports`, then open **InvoiceReport.cls** in the catalog.
3. Adjust the Maximum Rows and Maximum Duration properties of the query Invoice on which the report is built to suitable levels. You can adjust the values in either of the following ways:
   - In the Catalog Browser, right-click the query and select **Properties** from the shortcut menu. In the Properties sheet, change the property values as required, then press **Enter** on your keyboard to apply the change.
   - In the Resource View panel, right-click the query and click **Properties** on the shortcut menu. In the Properties dialog, edit the property values as required and then press **Enter** to confirm.

   Here, modify the value of Maximum Rows to **10** and change the value of Maximum Duration to **1** second. This specifies the maximum elapsed time to be one second, and that within this duration no more than ten records will be fetched.

   Tip: If you find that you are unable to edit the properties, make sure that you have unchecked the Forbid editing data object properties option in the Options dialog (**File** > **Options** > **Catalog** > **Forbid editing data object properties**).
4. View the report. You will find that only ten rows are fetched within one second.

Notes:

- You can decide where the two properties will be applied, on JReport side or on the database side, by configuring the JdbcDriversConfig.properties file. For details, see Specifying where to implement the maximum query run time and number of records.
- If the value of the Maximum Rows/Maximum Duration property is 0 or a negative number, it means the number will be unlimited.
- If the number you specify for Maximum Rows/Maximum Duration is equal to or more than that required for running all the records in your report, Maximum Rows and Maximum Duration will be unlimited.
- It is not recommended to specify both of the properties at the same time. Sometimes the time information for large-data reports may not be very accurate. For example, if you specify 5 seconds, you may find that it takes 7 seconds, since this time also includes the time for writing the history information to a file.
- Click the Refetch Data button on the toolbar of the View tab to run the report again if you find that the specified properties have not taken effect.
- These two properties apply to the specified query, not the report. For a report with a subreport, specifying the properties for a query used by the primary report will not have any effect on the subreport's query, and vice versa. For example, Report A.cls uses query Q1, and Report B.cls uses query Q2. B.cls is inserted to A.cls as a subreport. Maximum Rows for Q1 is modified to 3. The primary report will now only fetch three records, whereas Q2 for subreport B.cls will not be affected. To limit the subreport records, you will have to restrict Q2.

Viewing a query's running history

An important function of the Data Manager is to keep track of the run information of queries. By default, the information is not recorded and shown in the Data Manager to improve performance. If you want to show the history information, you need to first add the command line `alwaysShowDMRecord=true` in the file report.ini in `<install_root>\bin`, and then specify the Maximum Duration and Maximum Rows properties of the queries in JReport Designer. Then, the history information will be recorded in a file in `<install_root>\dm`. In this folder, JReport generates a subfolder for each catalog. For example, if the JReport sample report is run, a folder SampleReports will be created in `<install_root>\dm`. In addition, a file with a `.dmt` extension corresponding to the query will be created in the subfolder to store the run history information.

To change the Data Manager information output location (the default is `<install_root>\dm`), edit report.ini in `<install_root>\bin`. Change the location in the argument `dmPath=C\:\\JReport\\Designer\\dm` to the location where you would like the information to go to. For instance, `dmPath=C\:\\temp\\dm`. Make sure that the folder you specify in the report.ini argument already exists.

Viewing the Designer-side run history information of a query

1. Open the file **report.ini** in `<designer_install_root>\bin` and add the command line **alwaysShowDMRecord=true** as follows. Here, it is assumed that JReport Designer has been installed to `C:\JReport\Designer`.

   ```
   #paths
   #Thu Apr 19 13:57:30 GMT+08:00 2007
   dmPath=C\:\\JReport\\Designer\\dm
   helpPath=C\:\\JReport\\Designer\\help
   templatePath=C\:\\JReport\\Designer\\template
   tempPath=C\:\\JReport\\Designer\\temp
   stylePath=C\:\\JReport\\Designer\\style
   alwaysShowDMRecord=true
   ```

2. Start JReport Designer and open a report and adjust the Maximum Rows and Maximum Duration properties of the query on which the report is built to suitable values.
3. Click the **View** tab to view the report.
4. In the Resource View panel (or the Catalog Browser panel), right-click the query the report uses and choose **Data Manager** from the shortcut menu. You will now find that the run information of the query is displayed in the Data Manager dialog. You can sort the items by Date, Time, Rows, Duration, Report, simply by clicking the column title.

The following is an example to demonstrate how to use the Data Manager with the sample report InvoiceReport.cls. Here assumed that the command line `alwaysShowDMRecord=true` has been added to the file report.ini in `<designer_install_root>\bin`:

1. Click **File** > **Open** on the menu bar.
2. In the Open Report dialog, click the **Browse** button to open the catalog file **SampleReports.cat** in `<designer_install_root>\Demo\Reports\SampleReports`, then open the sample report **InvoiceReport.cls** but do not view it.
3. Click **View** > **Catalog Browser**.
4. In the Catalog Browser, right-click the query **Invoice** and select **Data Manager**. The Data Manager dialog is shown without any history information, since the report has not yet been run.
5. Close the Data Manager, then specify Maximum Rows as **5** and Maximum Duration as **3** in the Catalog Browser for the Invoice query.
6. Click the **View** tab to run the report. After the report result is shown, go to the Data Manager. You will now see the history of this first run of the query. The history information will include when the report is run - date and time, the running duration, the number of rows displayed, and the report that uses the query.
7. Close the report, and use the query Invoice to create another report report1.cls.
8. Run report1.cls, then the run information will be appended as a record in the Data Manager, but the Report field it shows will be report1.

Viewing the Server-side run history information of a query

1. Open the **report.ini** file in `<server_install_root>\bin` and add the command line **alwaysShowDMRecord=true** to it.
2. Publish a report that has had its query properties Maximum Rows and/or Maximum Duration set in JReport Designer to JReport Server (for details, see Publishing resources remotely).
3. Run the report on JReport Server. The Server Engine will save the query run information, but does not have a GUI to display it. In order to view the information, you will need to use the Data Manager in JReport Designer.
4. Open the report on JReport Server and start JReport Designer. Make sure that you have pointed the dmPath argument in the file report.ini in `<designer_install_root>\bin` to the output location (containing the dm folder) of the server. Then in JReport Designer, open the corresponding query in the Catalog Browser to show its server-side run history information.

Parameters

A parameter is a variable whose value is input at runtime. If a report contains a parameter, you will need to specify a value for the parameter when running the report. For example, if each time a report is run, an editor for the report would like to have his/her name placed in the report. You can simply insert a parameter directly into the report, and then each time you run the report, the value of the editor's name can be inserted. The runtime parameter dynamically controls your report result.

Also, you may want to input order date as an input parameter when running a report so that only orders on that date are displayed. In this way, parameters can be used as part of the query condition (search criteria), so that only the records which satisfy the specified parameter values are retrieved. In addition, JReport parameters can also be used for dynamic grouping or sorting.

Parameters defined in a catalog can be used in different reports. They can also be edited at any time.

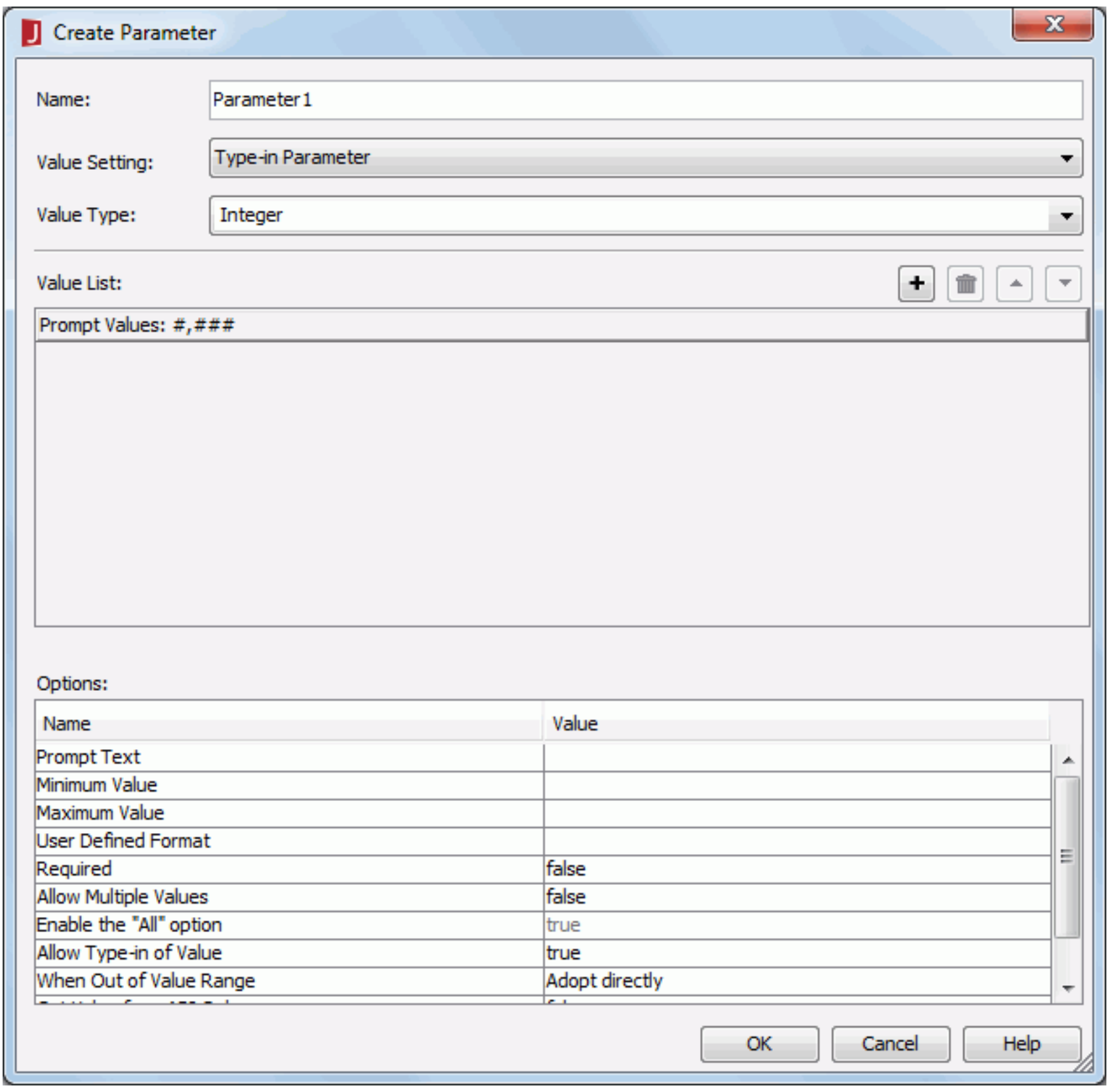
Click the following links to get details:

- Creating a parameter
- User defined format for parameters
- Parameter application cases

Creating a parameter

To create a parameter in a catalog, follow the steps below:

1. In JReport Designer, open the catalog file in which you want to create the parameter.
2. In the Data tab of the Catalog Browser, expand the data source that will include the parameter, right-click the **Parameters** node and select **Add Parameter** from the shortcut menu. The Create Parameter dialog appears.



3. In the Name field, input a name for the parameter.
4. Select a parameter type from the Value Setting drop-down list. For parameters in XML connection, it can only be Type-in Parameter.
5. In the values section, specify the parameter values as required. The section varies with the type you select from the Value Setting drop-down list.
   - **For Type-in Parameter:**
     a. Specify the data type of the parameter value from the Value Type drop-down list.
     b. Click to add a value line, double-click in it and then type in a value of the specified data type.

        If the parameter is of Date, DateTime, or Time type, you can also click the calendar button to set a date and time value using either calendar or built-in formula functions.
     c. Repeat the second step to add more values.

        To adjust the order of the values, click or ; to remove any unwanted value, select it in the list and then click .
     d. Specify the value which will be set as the default selected value in the Enter Parameter Value dialog by selecting it from the list.
   - **For Bind with Single Column:**
     a. From the Source drop-down list, select the required data source type: Tables and Views (including synonyms), Stored Procedures, Imported SQLs, or User Defined (if the specified catalog data source to which the parameter is added contains only a MongoDB connection, only Tables and User Defined are available in the drop-down list). Then, all fields in the specified catalog data source of this type will be available for value selection.
     b. Select a field from the Bind Column drop-down list. This field is used to filter the query when running a report with the parameter.
     c. If you want the values of another field to be displayed in the Enter Parameter Values dialog, which makes more sense to report end users than the values of the Bind Column field, select that field from the Display Column drop-down list.
   - **For Bind with Cascading Columns:**
     a. From the Data Source drop-down list, select the required data source from which to get the columns. You can get data from a table, view, synonym, stored procedure, imported SQL, or user defined data source. However, if the specified catalog data source to which the parameter is added contains only a MongoDB connection, only tables and user defined data sources are available.
     b. Click to add a parameter row.
     c. Click in the Bind Column cell and select a field from the drop-down list.
     d. If you want the values of another field to be displayed in the Enter Parameter Values dialog, which makes more sense to report end users than the values of the Bind Column field, click in the Display Column cell and then select that field from the drop-down list.
     e. Click in the Parameter cell to create the parameter. A name for the parameter will then be automatically added by JReport.
     f. Repeat steps b to e to create more parameters. Make sure that the selected Bind Column fields are of cascading relationship one level by one level down. In this way, a group of cascading parameters are created.

        To adjust the order of the parameters, click or ; to remove any unwanted parameter from the cascading group, select it and then click .
6. In the Options box, set options for the parameter according to your requirements.
7. When done, click **OK** to create the parameter.

User defined format for parameters

How can you define formats for different types of parameters? The most commonly used are the Date and Time formats.

Date format syntax

To specify the time format, use a time pattern string. In this pattern, all ASCII letters are reserved as pattern letters, and are defined as follows:

Symbol	Meaning	Presentation	Example
G	Era designator	Text	AD
y	Year	Number	1996
M	Month in a year	Text & Number	July & 07
d	Day in a month	Number	10
h	Hour in am/pm (1~12)	Number	12
H	Hour in a day (0~23)	Number	0
m	Minute in an hour	Number	30
s	Second in a minute	Number	55
S	millisecond	Number	978
E	Day in a week	Text	Tuesday
D	Day in a year	Number	189
F	Day of the week in a month	Number	2 (2nd Wed in July)
w	Week in a year	Number	27
W	Week in a month	Number	2
a	Am/pm marker	Text	PM
k	Hour in a day (1~24)	Number	24
K	Hour in am/pm (0~11)	Number	0
z	Time zone	Text	Pacific Standard Time
'	Escape for text		
''	Single quote		

Notes:

- The count of pattern letters determines the format.
- Text: 4 or more pattern letters -- use full form, < 4 -- use short or abbreviated form if one exists.
- Number: the minimum number of digits. Shorter numbers are zero-padded to this amount. However, Year is handled in a different way. If the number of letter 'y' is 2, then the year will be truncated to 2 digits.
- Text & Number: 3 or over, uses text, otherwise uses number.
- Any characters in the pattern that are not in the ranges of ['a'..'z'] and ['A'..'Z'] will be treated as quoted text. For example, characters such as ':', '.', ' ', '#' and '@' will appear in the resulting time text even if they have not been single quoted.
- A pattern containing any invalid pattern letter will result in a thrown exception during formatting or parsing.

Examples

- Format Pattern: "yyyy.MM.dd G 'at' hh:mm:ss z"
  Result: 1996.07.10 AD at 15:08:56 PDT
- Format Pattern: "EEE, MMM d, ''yy"
  Result: Wed, July 10, '96
- Format Pattern: "h:mm a"
  Result: 12:08 PM
- Format Pattern: "hh 'o''clock' a, zzzz"
  Result: 12 o'clock PM, Pacific Daylight Time
- Format Pattern: "K:mm a, z"
  Result: 0:00 PM, PST
- Format Pattern: "yyyyy.MMMMM.dd GGG hh:mm aaa"
  Result: 1996.July.10 AD 12:08 PM

Decimal number format syntax

Symbol	Meaning	Notes
0	A digit	
*	A digit, zero shows as a star	Can't mix 0, *, and _ in same format
_	A digit, zero shows as a space	Can't mix 0, *, and _ in same format
#	A digit, zero shows as absent	
.	Placeholder for decimal separator	
,	Placeholder for grouping delimiter	Shows the interval to be used
;	Separates formats	positive and negative.
-	If there is no explicit negative sign, - is prefixed	"0.00" -> "0.00;-0.00"
%	Divides by 100 and shows as a percentage	
X	Any other characters can be used in the prefix or suffix	

String format syntax

Symbol	Meaning	Example
%	Any string of zero or more characters.	WHERE title LIKE '%computer%' finds all book titles with the word 'computer' anywhere in the book title.
_ (underscore)	Any single character.	WHERE au_fname LIKE '_ean' finds all four-letter first names that end with ean (Dean, Sean, and so on).
[ ]	Any single character within the specified range ([a-f]) or set ([abcdef]).	WHERE au_lname LIKE '[C-P]arsen' finds author last names ending with arsen and beginning with any single character between C and P, for example Carsen, Larsen, Karsen, and so on.
[^]	Any single character not within the specified range ([^a-f]) or set ([^abcdef]).	WHERE au_lname LIKE 'de[^l]%' all author last names beginning with de and where the following letter is not l.

Parameter application cases

In JReport, parameters can be used in many situations. That is, you can apply parameters to many complicated tasks. The usage of parameters in JReport Designer can greatly facilitate your reporting work.

The following topics detail the usages of parameters:

- Using parameters in reports
- Referencing parameters in a formula
- Dynamically filtering queries
- Supplementing WHERE portions
- Binding a parameter to a DBField
- Applying RLS to a parameter
- Filtering a parameter with another parameter
- Dynamically grouping/sorting report data
- Using built-in functions to set date and time parameter values
- Editing the display sequence of parameters
- Controlling multiple parameters
- Importing parameter values
- Applying all values to a parameter in SQL statement

Reference: If you would like to reference parameter features showed in sample reports, you can open the report Parameter_*.cls in the catalog file SampleComponents.cat, which is located at `<install_root>\Demo\Reports\SampleComponents`.

Using parameters in reports

When you have created some parameters in a catalog, you can then use them as components in reports. For details, refer to the topic Parameter fields.

Referencing parameters in a formula

A formula which references parameters will return values according to the parameter values. To reference a parameter in a formula, follow the syntax: @ParameterName.

Here is an example of a formula which displays the URL for a company. The URL is based on the value for the parameter Company.

1. Open the demo catalog SampleReports.cat, in the Catalog Browser, expand the desired data source and create a type-in parameter named p_Company of String type (leave the other settings to their default).

2. In the same data source, create a formula named CompanyURL in the Formula Editor window as follows:

   ```
   "http://www." + @p_Company + ".com"
   ```

   Here, you can either manually type @p_Company, or you can select it from the Parameters node in the Fields panel (if you choose this option, the @ sign will automatically be inserted).

3. Insert the formula into a report.

4. View the report result, and type the string **Jinfonet** when prompted to enter the parameter value for p_Company. You will then find that the value of the formula becomes http://www.jinfonet.com.

This is a simple example of referencing a parameter in a formula. Here, the formula is placed directly into a report. You will better understand the benefits of referencing parameters in formulas which are used to control object properties.

Dynamically filtering queries

You can use parameter in a WHERE clause so that the query result can vary each time according to the entered parameter value. See the following example.

1. Open the demo catalog SampleReports.cat, then in the Catalog Browser, expand the desired data source and create a type-in parameter named IDSet of Integer type with the default values 1, 10, and 20, and enable the Allow Multiple Values option.

2. In the same data source, create a query CustomersInfo on the table Customers. Select all the fields contained in the table.

3. Filter the records of the query by adding a condition as follows in the Search Condition dialog (for details, see Filtering with the filter format):

4. Create a table report based on the query as follows: have the fields Customer ID, Customer Name, City and Phone displayed in the table and apply the Commercial style.

5. View the report. In the Enter Parameter Values dialog, click ... next to the value field of IDSet.

6. In the Enter Values dialog, uncheck the **All** option, add the listed values 1, 10, and 20 to the right box, and then click **OK**. You will find that the records with Customer ID equal to 1, 10 and 20 have been retrieved.

Customer ID	Customer Name	City	Phone
1	Absolute Java	New York City	(212) 555-3462
10	Cafe Italy	Indianapolis	(317) 555-1274
20	International Coffee	Los Gatos	(650) 555-9879

7. View the report again. This time, in the Enter Values dialog, remove 1, 10, and 20 from the Selected Values box first. Enter the values 3, 7, 9, 11, 15, and 25 in the Enter Values field and add them into the Selected Values box one by one. Then click **OK**. Now, the records for the specified Customer IDs are displayed.

Customer ID	Customer Name	City	Phone
3	Automatic Java Mac	Palo Alto	(650) 555-4690
7	Cafe Brazil	Colorado Springs	(719) 555-1745
9	Cafe International	Irvine	(714) 555-9548
11	Cafe Kenya	Boston	(617) 555-5998
15	Gourmet Coffee Inte	Tacoma	(253) 555-9834
25	Java House	Melbourne	(407) 555-9111

To view the report with full data, select the **All** option and then click **OK** in the Enter Values dialog.

Notes:

- The parameter's data type should be set according to the field type and the manner in which you use the parameter. For example, if the field is a number, define a parameter to be Number type also.

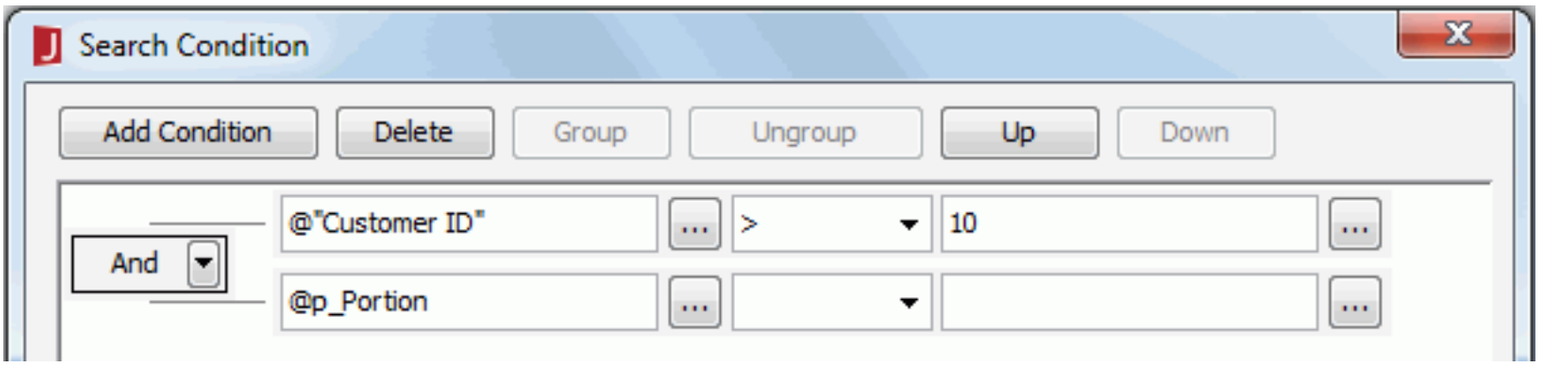
- If you set the parameter format to be @ParameterName, to specify a String type value, your typed value should be without single quotations, for example, USA. Once you input a parameter value with single quotation marks, the quotation marks will be recognized as a part of the parameter value.

  However for the format :ParameterName, to specify a String type value, your typed value should be with single quotations, for example, 'USA'. If the data type is Number, the parameter value should be like: 1234.

Supplementing WHERE portions

JReport allows you to filter a query at runtime. To do this, you can call the method setWherePortion(). However, by calling this method, the original WHERE portion of the query will be replaced. If you want to append the runtime WHERE portion to the original one instead of replacing it/them, you can do it in the following way:

1. Open the demo catalog SampleReports.cat, then in the Catalog Browser, expand the desired data source and create a type-in parameter named p_Portion of String type (leave the other settings to their default).
2. In the same data source, create a query CustomersInfo on the table Customers. Select all the fields contained in the table.
3. Create a table report based on the query, and have the fields Customer ID, Customer Name, City and Phone displayed in the report, then apply the Commercial style.
4. In the Resource View panel, right-click the node that represents the query, and then select **Edit Query** from the shortcut menu.
5. In the Query Editor, click **Query** > **Filter**.
6. In the Search Condition dialog, add two condition lines (WHERE portions) for the query as follows (for details, see Filtering with the filter format):



The colon is used to append the parameter value (of String type) to the first condition row instead of replacing it.

7. Click **OK** to apply the change to the query.
8. View the report, and in the Enter Parameter Values dialog, write a WHERE portion in standard SQL syntax as value of p_Portion, for example, CUSTOMERID<20. Then, both of the two WHERE portions will be used to filter the query.

Customer ID	Customer Name	City	Phone
11	Cafe Kenya	Boston	(617) 555-5998
12	Cafe San Francisco	San Francisco	(415) 555-9090
13	Coffee House of Ha	Fairfax	(703) 555-0981
14	Java Nike Space	Silver Spring	(301) 555-9384
15	Gourmet Coffee Inte	Tacoma	(253) 555-9834
16	Gourmet Java Club	London	01 (202) 555846
17	Green Mountain Jav	Cleveland	(216) 555-4348
18	Guatemala Coffee H	Dallas	(972) 555-9452
19	International Coffee	Englewood	(201) 555-0001

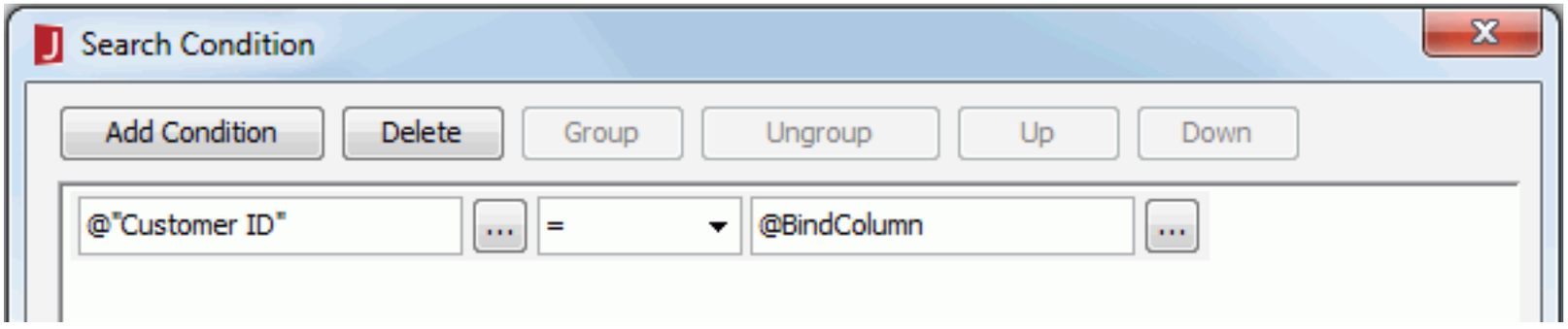
Binding a parameter to a DBField

JReport provides a mechanism which links each field value of a "display column" with the exact value of a "bind column". When you select the field value from the "display column", the field value of the "bind column" is sent to the query to filter the query result.

For example, if you specified customer ID to query data, but you would like to select customer name not customer ID in the parameter specifying dialog which appears at runtime, you can do so by using the functionality - Parameter Binding.

1. Open the demo catalog SampleReports.cat and in the Catalog Browser, expand the data source where to create the parameter.
2. Right-click the **Parameters** node and then click **Add Parameter**.
3. In the Create Parameter dialog, enter **BindColumn** in the Name field, select **Bind with Single Column** from the Value Setting drop-down list, select **Tables and Views** from the Source drop-down list, then select **Customer ID** from the Bind Column drop-down list, and choose **Customer Name** from the Display Column drop-down list.

   Note: Selecting a display column different from the bind column will establish a link between the two columns (DBFields). When you select the field value from the display column, the corresponding field value of the bind column is actually assigned to the parameter.
4. In the same data source, create a query CustomersInfo on the table Customers. Select all the fields contained in the table.
5. Create a standard banded report based on the query, and have the fields Customer ID, Customer Name, City and Phone displayed in the report.
6. In the Resource View panel, right-click the node that represents the query, and then select **Edit Query** from the shortcut menu.
7. In the Query Editor, click **Query** > **Filter**.
8. In the Search Condition dialog, add a condition for the query as follows (for details, see Filtering with the filter format):



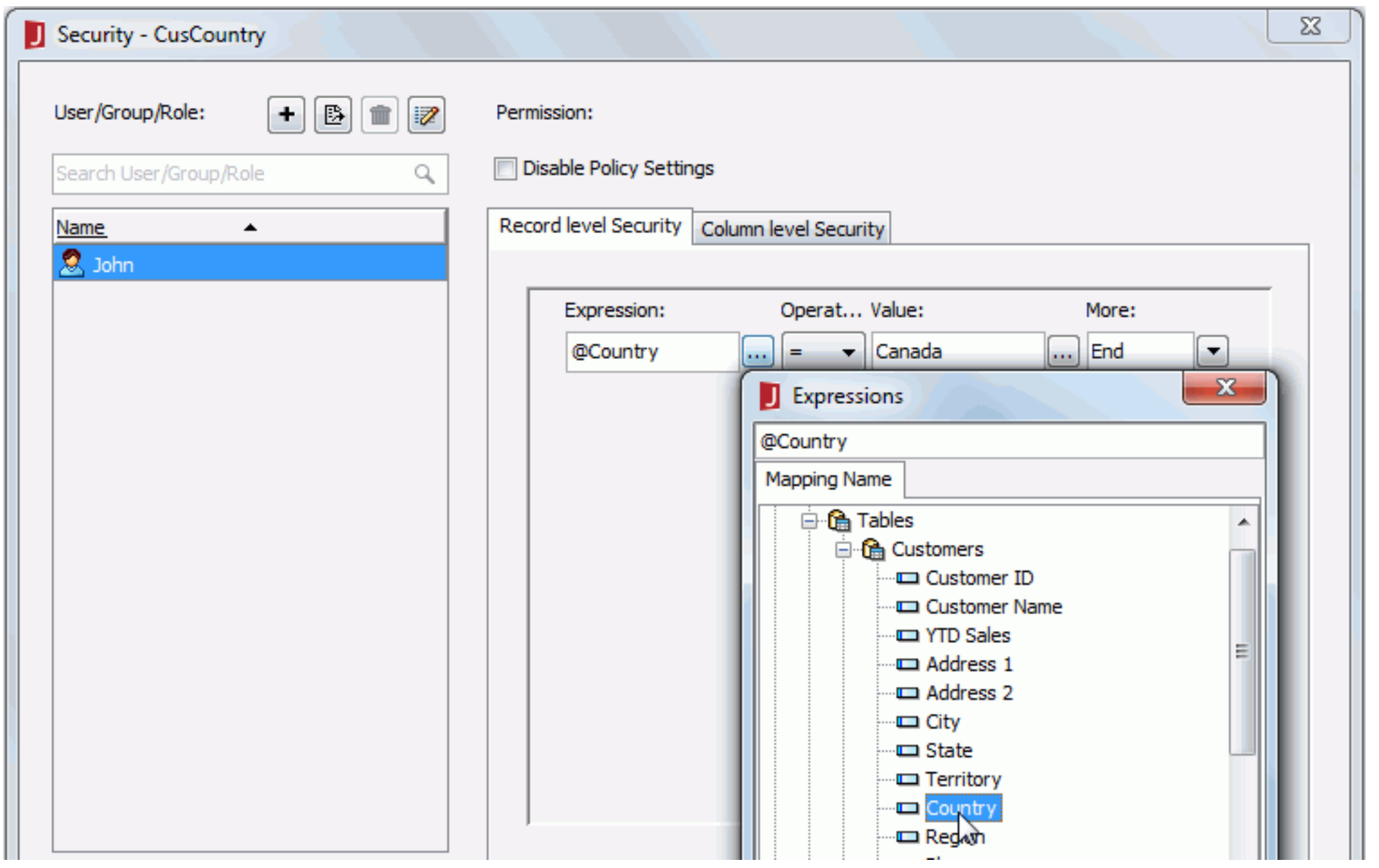
9. Click **OK** to apply the changes to the query.
10. Click the **View** tab to view the report. All the customer names will be listed in the Enter Parameter Values dialog. You can select a value from the list to query data with the customer ID instead of customer name.

Applying RLS to a parameter

When a parameter is bound with another column, you can apply a record-level security policy (RLS) in the data source, which is defined on the bound column, to the parameter, then at runtime, end users will only see data which the SID allows to view display in the parameter value drop-down list.

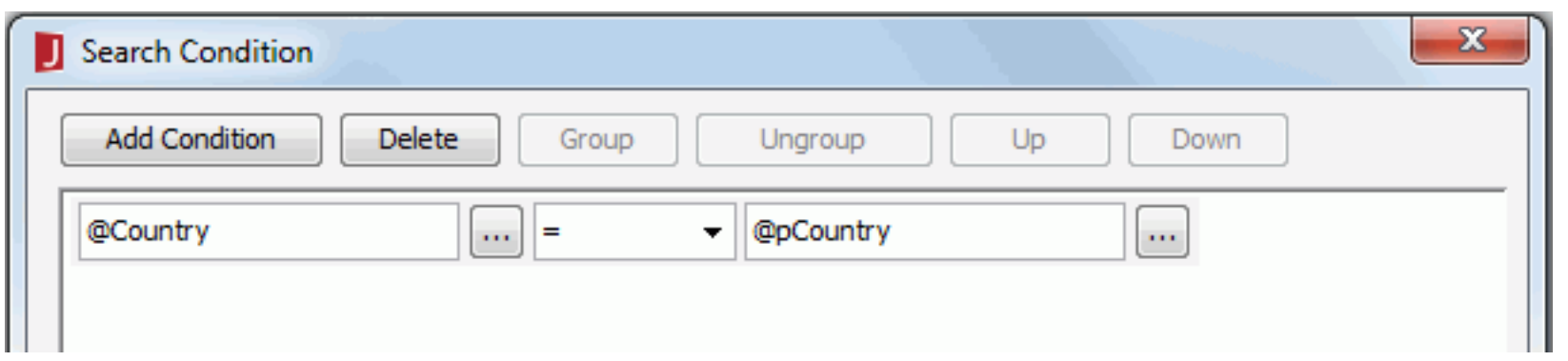
The following example explains how to apply RLS to the value drop-down list of a parameter in detail.

1. Open the demo catalog SampleReports.cat and in the Catalog Browser, expand the data source where to create the parameter.
2. Right-click **Security Entry** in the Security node and then click **Add Security Entry**.
3. Name the security policy **CusCountry** and in the Security dialog check **Valid RLS**.
4. Click and then select **Add User** to add the user John, which has the permission to view the Canada records in the Country column only.



5. Add another user David which has the permission to view records in the United Kingdom only.

   For details about how to set up a RLS policy, refer to Setting up a connection-scope security policy.
6. In the same data source, right-click the **Parameters** node and then click **Add Parameter**.
7. In the Create Parameter dialog, enter **pCountry** in the Name field, select **Bind with Single Column** from the Value Setting drop-down list, and bind the parameter with the **Country** column. In the Options box, select **CusCountry** as value of Record Level Security, and then set the Distinct option to **true**.
8. Create a query CustomersInfo on the table Customers. Select all the fields contained in the table.
9. Create a standard banded report based on the query, and have the fields Customer ID, Customer Name, Country and Phone displayed in the report.
10. In the Resource View panel, right-click the node that represents the query, and then select **Edit Query** from the shortcut menu.
11. In the Query Editor, click **Query** > **Filter**.
12. In the Search Condition dialog, add a condition for the query as follows (for details, see Filtering with the filter format):



13. Click **OK** to apply the changes to the query.
14. View the report, and in the Security Identifier dialog, enter **John** and click **OK**, the Enter Parameter Values dialog will then be displayed.
15. Click the parameter value drop-down list, and you can see that only Canada is displayed for the user SID.
16. View the report again and this time enter the user SID David. You will find that only the value United Kingdom is displayed in the parameter value drop-down list this time.

Notes:

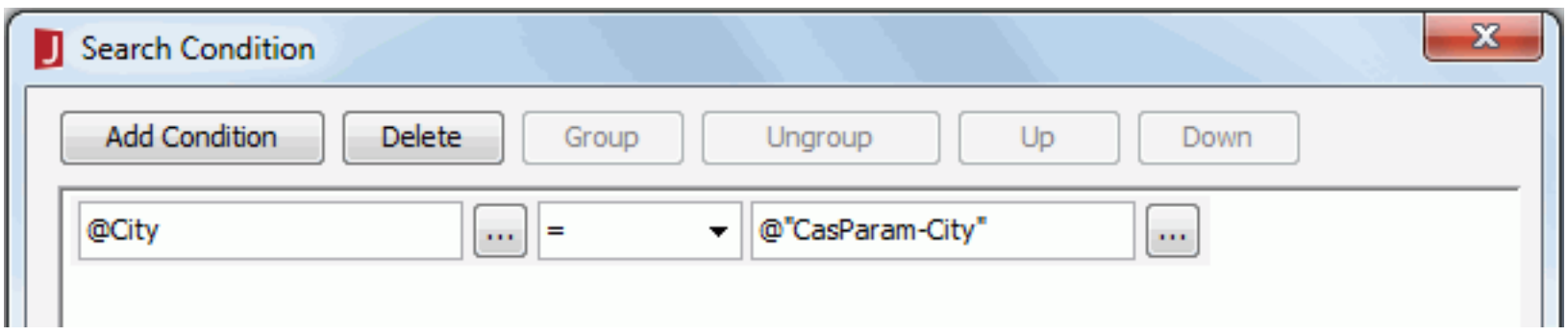
- When applying an RLS policy to a parameter bound with a column, you must make sure the parameter bound column and the column used to set conditions in the RLS policy are in the same table. In addition, if the RLS policy also contains conditions defined on other columns, all the columns must be in the same table, and the RLS policy cannot contain parameters and formulas; otherwise, when you apply the RLS to the parameter, the RLS would not work properly.
- When you have applied an RLS policy to a parameter and then you click the SQL button in the Create Parameter dialog to edit the SQL statement of the parameter, the parameter value drop-down list may be Null because the changed SQL statement may not match the permissions of the RLS.
- To have the RLS applied to a parameter work on JReport Server, the users for the RLS must be imported from the server. If the users are created in JReport Designer, you need to first create these users on JReport Server, and then in JReport Designer, synchronize the security information with JReport Server by means of importing security information from the server with the Merge option checked.
- When previewing a report that contains parameters applied with RLS in the Page Report Result or Web Report Result format in JReport Designer, you need to make sure your RLS policy contains the user Jinfonet created using the way mentioned above. That is because the user Jinfonet is the default and only user of the preview server. Any other users in the RLS policy cannot be recognized by the preview server.

Filtering a parameter with another parameter

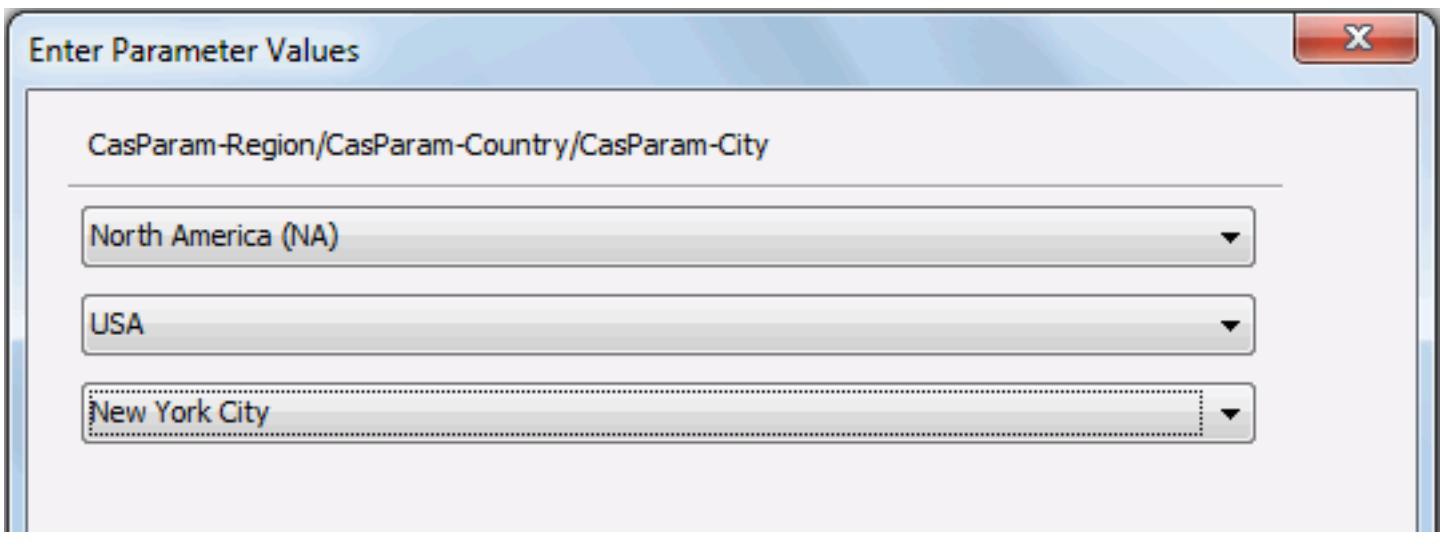
You can use the value of one parameter to filter another one. The following examples show you the two methods that you can use to achieve this purpose.

Method 1

1. Open the demo catalog SampleReports.cat and in the Catalog Browser, expand the desired data source node.
2. Right-click the **Parameters** node and then click **Add Parameter**.
3. In the Create Parameter dialog, enter **CasParam** in the Name field, select **Bind with Cascading Columns** from the Value Setting drop-down list, and then select **Customers** of the Tables type from the Data Source drop-down list.
4. Click to add a parameter row, select **Region** as the Bind Column and Display Column, and then click in the Parameter cell to create the parameter in the cascading group.
5. Click to add another parameter row, select **Country** as the Bind Column and Display Column, and then click in the Parameter cell.
6. Click to add one more parameter row, select **City** as the Bind Column and Display Column, click in the Parameter cell, and then click **OK** in the dialog.
7. In the same data source, create a query CustomersInfo on the table Customers and select all the fields in the table.
8. Filter the records of the query by adding a condition as follows in the Search Condition dialog (for details, see Filtering with the filter format):



9. Create a table report based on the query, and have the fields Customer Name, City, Country and Region displayed in the report.
10. View the report and the Enter Parameter Values dialog appears.
11. From the region drop-down list, select a region, then only the countries in the selected region will be displayed in the country drop-down list. Choose a country from the list, and then only the cities in the selected country will be displayed in the city drop-down list.



12. Click **OK**. You will now find that only the specified records are shown.

Method 2

1. Open the demo catalog SampleReports.cat and in the Catalog Browser, expand the desired data source node.
2. Right-click the **Parameters** node and then click **Add Parameter**.
3. Create a parameter named ParamRegion, select **Bind with Single Column** from the Value Setting drop-down list, select **Tables and Views** from the Source drop-down list, select **Region** as the Bind Column and Display Column. Then in the value cell of the Import SQL option, you can see the following statement:

   ```
   SELECT CUSTOMERS.REGION FROM CUSTOMERS
   ```

4. Create another parameter named ParamCountry. Bind it with the column Country, then click in the value cell of Import SQL and edit its SQL statement as follows:

   ```
   SELECT CUSTOMERS.COUNTRY FROM CUSTOMERS WHERE (Customers.Region=@ParamRegion)
   ```

5. Create one more parameter named ParamCity. Bind it with the column City, and then edit its SQL statement as follows:

   ```
   SELECT CUSTOMERS.City FROM CUSTOMERS WHERE (Customers.Country=@ParamCountry)
   ```

6. In the same data source, create a query CustomersInfo on the table Customers and select all the fields in the table, and filter the records of the query by adding a condition as follows:

   ```
   CUSTOMERS.CITY=@ParamCity
   ```

7. Create a table report based on the query, and have the fields Customer Name, City, Country and Region displayed in the report.
8. View the report. In the Enter Parameter Values dialog, you can use the value of ParamRegion to filter the value of ParamCounty, and use the value of ParamCounty to filter the value of ParamCity.

Dynamically grouping/sorting report data

An important usage of parameters is to use them to dynamically group and sort data in a report, which enables you to specify the grouping and sorting conditions at runtime according to the varied requirements.

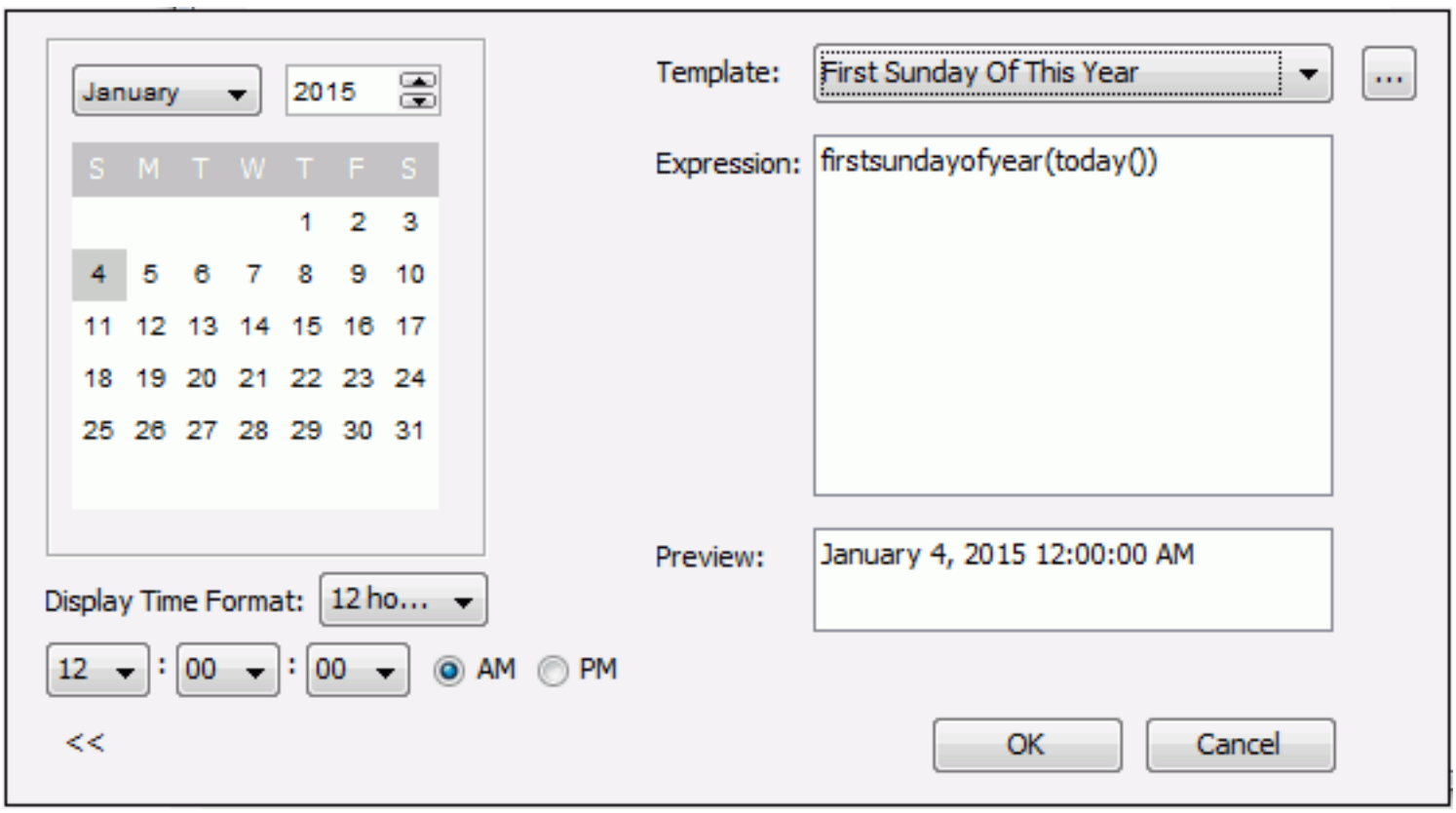
For details, refer to the sections Grouping data dynamically and Sorting data dynamically.

Using built-in functions to set date and time parameter values

For a parameter of the Date, DateTime, or Time type, you can customize a dynamic date and time value by creating an expression using built-in formula functions.

To do this:

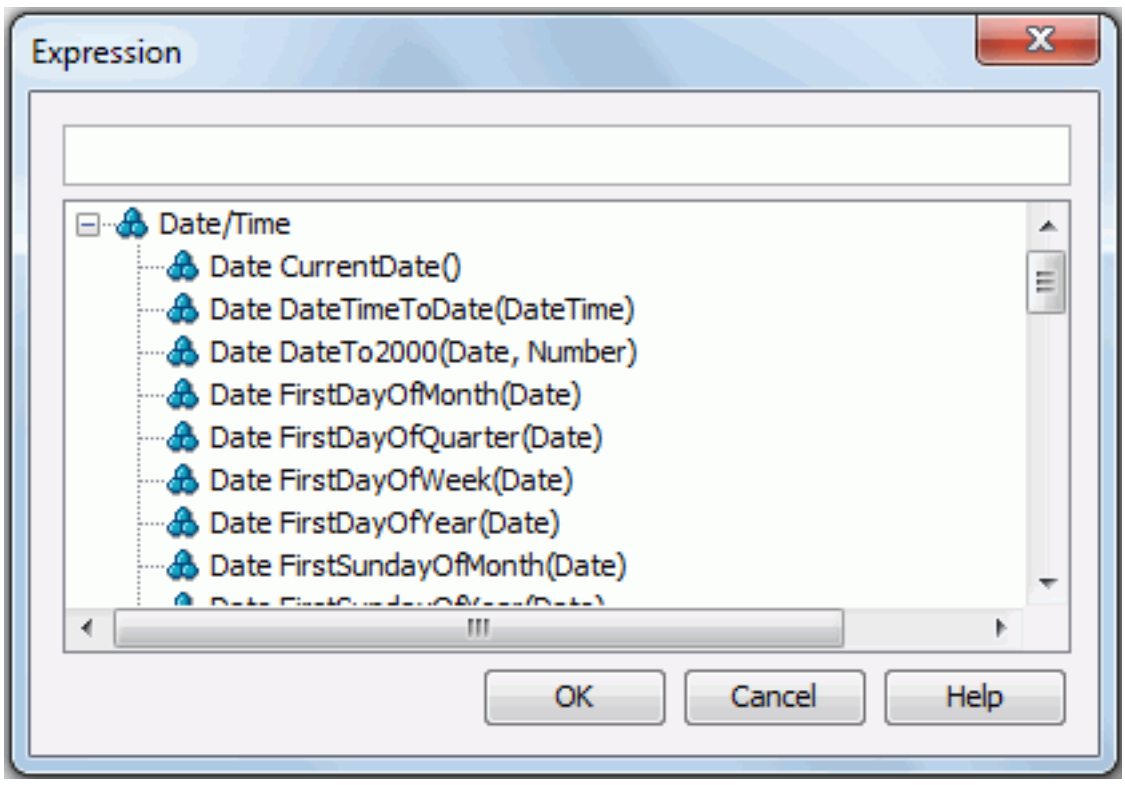
1. In the places where you are able to specify a value for the parameter, you may see the calendar button . Click this button and the Calendar dialog appears.



2. The Template drop-down list on the right provides some predefined expressions for you to use. You can customize your own expression either based on an existing template or by directly editing the contents in the Expression text box.

   If you cannot see the Template label on the right, click **>>** on the bottom left corner.

3. JReport's built-in Date/Time formula functions are available for inserting into your expression. To make use of them, click and the Expression dialog appears.



4. The top text box shows the current expression statement. Expand the **Date/Time** node below, double-click the required function to insert it where the cursor is located in the expression. Upon finishing, click **OK** to return to the Calendar dialog.

5. The Preview box calculates the result of the current expression. Each time you modify the expression, you can click in the box to refresh the result. If it is an invalid expression, an error message will be shown in the box and no change will be made to the calendar and the currently selected date/time will be remained in the calendar.

6. When done, click **OK** to add the value.

Editing the display sequence of parameters

When you run reports with parameters, a parameter page will be displayed for specifying the values of the parameters. The order of the parameters on this parameter page are adjustable.

There are two types of parameter order settings: one is for the whole catalog, the other a specific report. When the order of parameters in a catalog is specified, the order will be applied when running reports created in the catalog. If you just specify the parameter order for a report, only when you run the report will the order be adopted. When both types of orders are configured, the report level order has higher priority than the catalog level order.

Adjusting for a catalog

1. Open the catalog with parameters of which you would like to adjust the order.
2. In the Catalog Browser, expand a data source, select the **Parameters** node or any parameter in the node, then click the **Sort Parameter** button on the Catalog Browser toolbar.
3. In the Parameter Run Time Order dialog, make use of the Move Up and Move Down buttons to adjust the order of the parameters, or simply drag and drop a parameter. Since a catalog may have many parameters, it would be much easier by dragging and dropping.

   Move a parameter closer to the top position if you want it at a better place for specifying its value more easily at runtime.
4. When done, click **OK** to accept the changes.

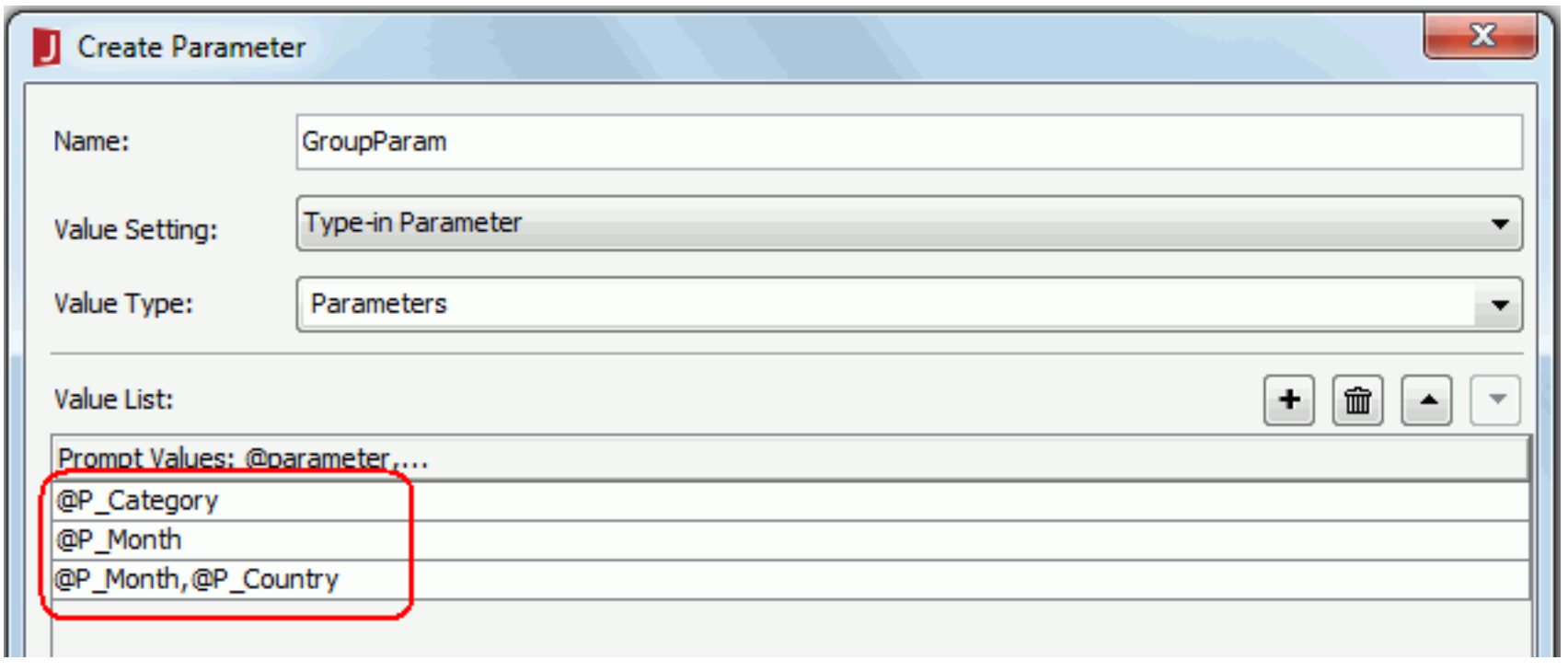
Adjusting for a specific report

1. Open the report with parameters of which you would like to adjust the order, then click **Edit** > **Parameter Order** on the menu bar.
2. In the Parameter Run Time Order dialog, make use of the Move Up and Move Down buttons to adjust the order of the parameters, or simply drag and drop a parameter. Move a parameter closer to the top position if you want it at a better place for specifying its value more easily at runtime.
3. Click **OK**.

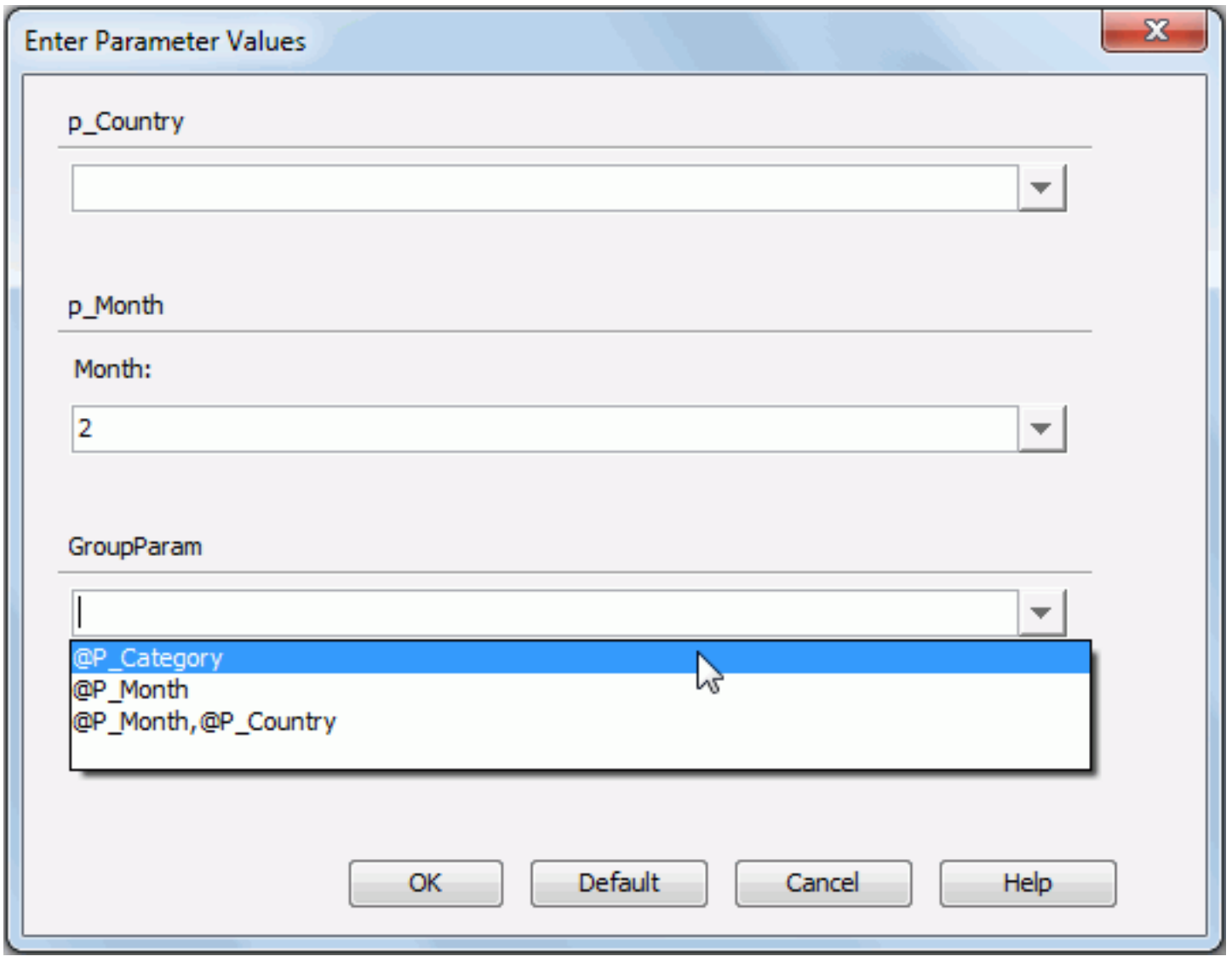
Controlling multiple parameters

When multiple parameters are used in a report, but you only want to show some of them when running it, you can group them and then select the parameter you want for the report. This function can be implemented by the Parameters type parameters. See the example below:

1. Open the demo catalog SampleReports.cat in `<intall_root>\Demo\Reports\SampleReports`.
2. In the Catalog Browser, expand **Data Source 1**, then right-click the **Parameters** node and select **Add Parameter**.
3. In the Create Parameter dialog, enter **GroupParam** in the Name field, select **Parameters** from the Value Type drop-down list, then add three parameters as the default values of GroupParam as shown in the following image:



4. Open an existing report. Insert the parameters P_Category, P_Month, P_Country, and GroupParam into a report tab of the report.
5. View the report, and the Enter Parameter Values dialog appears. You can select the parameter you want from the GroupParam drop-down list.



Select @P_Category, the P_Category parameter is displayed in the dialog and you can select or type in a value for the parameter. If you select @P_Month,@P_Country, then both the parameters P_Month and P_Country will be displayed for you to specify values with which to run the report.

Importing parameter values

When you view a report with parameters, the Enter Parameter Values dialog will be displayed, showing the parameters the report uses for you to input values. The default values shown in the dialog are fixed at the development time. However, showing a default, fixed value which is dated makes no sense to users.

JReport provides an interface for importing different default values from an outside class file so that the default values can be specified flexibly. There is only one method for this interface: public Hashtable promptValues(String paramsName[]);

Reference: JReport Javadoc jet.util.ImportParamValues interface in `<install_root>\help\api`.

The following are the general steps for importing parameter values from a class file:

1. Define a Java class file. For help, see our sample programs ParamTest.java and TestParamList.java in `<install_root>\help\samples\APIParameter`. To import parameter values, implement the interface jet.util.ImportParamValues in your Java file. The class definition may be as follows:

```
package help.;
import java.util.*;
import jet.util.*;
public class ParamTest implements ImportParamValues
{
    public ParamTest()
    {
        //Class body
    }
    public Hashtable promptValues(String paramsName[])
    {
        //Class body
    }
}
```

   The following explains the code:

   - The package name can be defined by yourself. In our sample code, we use help as the package name.
   - Since the interface ImportParamValues is in the package jet.util, you will have to import the class package jet.util.*.
   - A public constructor method without parameters is required.

2. Compile the Java file to generate the class file.
3. Append the class path to the ADDCLASSPATH variable of the file setenv.bat in `<install_root>\bin`.
4. Start JReport Designer with the modified batch file.
5. Open the report with the parameter value that you want to import.
6. In the Report Inspector, set the property Import Parameter Values of the report to be the class name that you just generated with the package name, and set Parameter List Auto to **false**.

The following two examples explain how to import parameter values and what you should be aware of while using this feature.

Example 1: Importing from a database

In the sample program ParamTest.java in `<install_root>\help\samples\APIParameter`, different values were assigned to the parameter p_StartDate by accessing values from an HSQL database. In addition, the sample report CustomerAnalysis.cls uses this parameter in the query condition. So here, we use this demo report as an example to explain how to import parameter values from a database, so that you do not need to type in the values one by one while specifying the parameter default value (note that in this example we assume your JReport Designer is installed in the default path, that is `C:\JReport\Designer`).

Note that in the sample ParamTest.java, we use the method compareToIgnoreCase() to compare the parameter name in the class file with the one in your report. This method is not case sensitive when performing the comparison.

1. Compile ParamTest.java to generate ParamTest.class, and store the class file in `C:\JReport\Designer\help`.

   ```
   javac -classpath c:\JReport\Designer\lib\JREngine.jar; ParamTest.java
   ```

2. Append `C:\JReport\Designer\help` to the ADDCLASSPATH variable of setenv.bat in `C:\JReport\Designer\bin`.
3. Start JReport Designer and open CustomerAnalysis.cls.
4. In the Report Inspector, find the report property **Import Parameter Values**, and input the class name with the full package name. In this example, input **help.ParamTest**, and then set the property Parameter List Auto to **false**.

   Tip: The report node is not shown in the resource tree of the Report Inspector by default. To have it shown, select the report tab node in the tree (in this example, the CustomerAnalysis node), and then click the **Up** button on the Report Inspector toolbar.

5. Save the report and catalog, and then view the report result.
6. In the Enter Parameter Values dialog, click the drop-down list of the parameter. You will find that all the values you specified in ParamTest.java have been imported into the list.
7. Select one of the values with which to view the report.
8. Publish the saved report CustomerAnalysis.cls to JReport Server (for details, see Publishing resources remotely).
9. Append `C:\JReport\Designer\help` to the ADDCLASSPATH variable of setenv.bat in `<server_install_root>\bin`.
10. Start JReport Server with the modified batch file.
11. Load your web browser to access the JReport Server Console Page. You will then have the same parameter values as you did in JReport Designer.

Example 2: Importing from a class file

In addition to Example 1, another example of applying the feature of importing parameter values is to specify your required default values. When you group/sort data dynamically with a parameter, when viewing the report, a dialog will be displayed, in which you will find many default values in the drop-down list. However, since you do not need so many values with which to group/sort, you can specify your required default values in a Java class, and then import them. For help, refer to the sample program TestParamList.java in `<install_root>\help\samples\APIParameter`.

To specify the default parameter values in a class file and then import them, follow the steps below:

1. Compile TestParamList.java to generate TestParamList.class, and store the class file in `<install_root>\help`.
2. Append `<install_root>` to the ADDCLASSPATH variable of setenv.bat in `<install_root>\bin`.
3. Start JReport Designer.
4. Create a report with a dynamic group/sort using the parameter ImportPara (for details, see Grouping data dynamically and Sorting data dynamically).
5. Set the report property Parameter List Auto to **true** in the Report Inspector, and view the report. Many values will be listed in the Enter Parameter Values dialog.
6. Set Parameter List Auto to **false**, Import Parameter Value to **TestParamList**, and view the report again. You will now see that only four values are displayed as defined in TestParamList.java.

Notes:

- You need to ensure that the parameter format in the class file is consistent with that in JReport, specifically, that the parameter format used in the class file is parsed when viewing the report so that the correct data records will be returned.
- When you implement the interface jet.util.ImportParamValues and use its function promptValues(), ensure that:
  - The parameter of this function is a String array which contains the names of the parameters.
  - The return value of this method is a Hashtable, its keys are parameter names used in the report and its values are vector objects which contain the parameter values.
- When there are subreports in a primary report, setting the properties Import Parameter Value and Parameter List Auto will not affect them. That is, you have to set these properties in the primary report and subreports separately.
- If you want to use a parameter as grouping/sorting criteria, you will need to make sure that the parameter values are within the columns used in the report's query. In addition, the name should be consistent with the column names defined in the catalog. For example, the query of the report joins two tables: Orders and Orders Detail. In the Catalog Browser, expand the query used by the report. You will then see all the columns from the two tables. Make sure that the values returned are within them and consistent with the column names.

Applying all values to a parameter in SQL statement

If a parameter allows multiple values and is enabled to use one single value "All" to represent all its values (the way to enable the "All" value is to set Enable the "All" Option to true when creating or editing the parameter), when you use the parameter in an imported SQL, you should embed each IN condition in parenthesis to enable the "All" value to work properly.

For example, there are two parameters: @pCountry and @pID. Suppose that they both can allow multiple values and have enabled the "All" option and you would like to reference them in an imported SQL query, you should embed each IN condition in parenthesis as follows:

```
select... from ... where ... and ... and (country in @pCountry) and (customerid in
@pID)
```

There is another case in which you should also embed each IN condition in parenthesis when using such parameters for the same purpose: when you create or edit a JReport query via the Query Editor, you choose to modify its SQL statement manually. If the customized statement fails to be parsed by JReport and you would like to save the statement anyway, then it is up to you to guarantee the statement correctness. In this case, you should embed each IN condition in parenthesis.

Summaries

Summaries allow you to generate functions for your data, such as count, average, sum, and standard deviation. Summary functions are based on how you initially grouped the data. For example, if you grouped your data according to country, and you have two groups (US and Canada). Then, if you specify to perform a summary break by country group, you will get the summary result for each group.

A summary is a special kind of formula, but it is often used for DBField or an existing formula. Summary functions can also be realized with formula functions. Since summaries are frequently used in reports, JReport provides you with an easy-to-use GUI for this purpose.

The following topics describe summaries in detail:

- Summary functions
- Creating a summary
- Using summaries in reports

See an example: The SampleComponents catalog, included with JReport Designer, contains reports that have examples of how summaries could be used in a report. For the example, open `<install_root>\Demo\Reports\SampleComponents\UsingFormula.cls`.

Summary functions

The following are functions that can be used for creating summaries. They are divided into two types: normal functions and running functions.

A normal function calculates all records and returns only one fixed value.

A running function calculates a value on each record dynamically. It is like the total sum of money for each transaction that is recorded on your checkbook. Each time you deposit or withdraw money from the bank, the record will compute the total of your checking account balance.

- **Count**
  Total count of all the values in a field. The value will be shown in the Group Footer. For example, if you have six fields in a group, the value generated by Count will be 6.
- **Sum**
  This function sums up all the numerical values in a field. The value will be shown in the Group Footer. For example, if a field contains 5, 10, 15, 20 and 25, the Sum function will return 75.
- **Minimum**
  This function finds the smallest numerical value in a field. The value will be shown in the Group Footer. For example, if a field contains 5, 10, 15, 20, and 25, the Minimum function will return 5.
- **Maximum**
  This function finds the largest numerical value in a field. The value will be shown in the Group Footer. For example, if a field contains 5, 10, 15, 20, and 25, the Maximum function will return 25.
- **Average**
  This function takes the average of all the numerical values in a field. The value will be shown in the Group Footer. For example, if a field contains 5, 10, 15, 20, and 25, the Average function will return 15.
- **Distinct Count**
  This function takes a total count of all the distinct fields in a report. This value will be shown in the Report Footer. For example, if you have forty three fields in your report and three fields have identical values, the value generated by Distinct Count will be 41.
- **Population Standard Deviation**
  This function uses the following equation to compute its value. PSD takes the square root of the Population Variance.

$$\sqrt{\frac{\sum_{i=0}^{n}(\bar{x}-x_i)^2}{n}}$$

- **Standard Deviation**
  This function uses the following equation to compute its value. SD takes the square root of the Variance.

$$\sqrt{\frac{\sum_{i=0}^{n}(\bar{x}-x_i)^2}{n-1}}$$

- **Population Variance**
  This function uses the following equation. PV sums up the square of the difference between a value and its average, and then divides the result by the total number of values.

$$\frac{\sum_{i=0}^{n}(\bar{x}-x_i)^2}{n}$$

- **Variance**
  This function uses the following equation. Variance sums up the square of the difference between a value and its average, and then divides the result by the total number of values subtracted by one.

$$\frac{\sum_{i=0}^{n}(\bar{x}-x_i)^2}{n-1}$$

- **Running Count**
  This function takes a total count on running records of the values in a field.
- **Running Distinct Count**
  This function takes a total count of all the distinct fields in a report on running records.
- **Running Sum**
  This function sums up all the numerical values in a field on running records.
- **Running Minimum**
  This function finds the smallest numerical value in a field on running records.
- **Running Maximum**
  This function finds the largest numerical value in a field on running records.
- **Running Average**
  This function takes the average of all the numerical values in a field on running records.

Examples

The following are two examples comparing normal functions and running functions.

Sum on banded object

Summaries can be added in a banded object as columns which makes the comparison quite obvious.

The first page of a banded object:

1 of 33

Order Date	Quantity	RunningSum_Quantity	Sum_Quan
7/6/2014	266	266	997056
9/6/2014	105	371	997056
5/16/2013	105	476	997056
12/5/2013	343	819	997056
8/5/2014	266	1085	997056
11/20/2014	266	1351	997056
11/2/2013	210	1561	997056
3/2/2014	532	2093	997056
6/4/2014	161	2254	997056
10/10/2014	343	2597	997056
7/21/2013	196	2793	997056
10/6/2014	511	3304	997056
12/7/2013	49	3353	997056
2/7/2014	77	3430	997056
7/22/2013	1365	4795	997056
11/22/2013	259	5054	997056
7/22/2014	420	5474	997056
9/21/2013	161	5635	997056

The last page of the banded object:

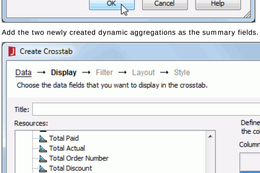
33 of 33

Order Date	Quantity	RunningSum_Quantity	Sum_Quantity
4/8/2013	154	995093	997056
4/7/2013	91	995184	997056
4/7/2013	357	995541	997056
4/7/2013	200	995741	997056
4/7/2013	216	995957	997056
4/7/2013	231	996188	997056
4/8/2013	9	996197	997056
4/8/2013	13	996210	997056
4/8/2013	16	996226	997056
4/8/2013	22	996248	997056
4/13/2013	150	996398	997056
4/13/2013	190	996588	997056
4/13/2013	157	996745	997056
4/13/2013	125	996870	997056
4/13/2013	86	996956	997056
4/13/2013	100	997056	997056

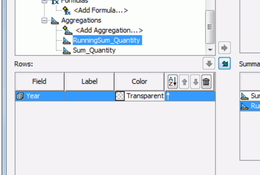
As revealed in the first and last pages of the banded object, the sum on quantity is always the same value which is a total quantity of all the records, while running sum on quantity is the current accumulated value on each row.

Sum on crosstab

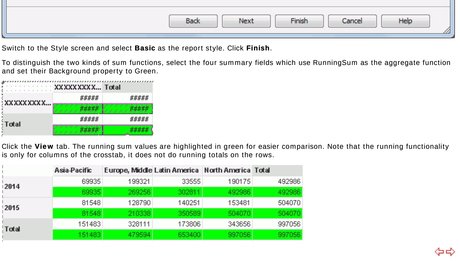
1. Open the catalog file SampleReports.cat.
2. Click **File** > **New** > **Web Report** on the menu bar.
3. Drag **Crosstab** from the Toolbox panel to the web report. The Create Crosstab wizard appears.
4. In the Data screen of the wizard, select **SaleStat** from Data Source 1. Then click **Next**.
5. In the Display screen, add the DBField **Assigned Region** to the Columns box and **Year** to the Rows box.
8. In the Dynamic Resources > Aggregations node, click **<Add Aggregation...>**.
9. In the Add Aggregation dialog, input **Sum_Quantity** in the Aggregation Name text field, click beside the Mapping Name text field and select **Quantity** as the mapping name, then select **Sum** as the aggregate function and click **OK**.



10. Repeat the step above to create another aggregation named RunningSum_Quantity, also use Quantity as the mapping name, and specify the aggregate function as RunningSum.



11. Add the two newly created dynamic aggregations as the summary fields.



12. Switch to the Style screen and select **Basic** as the report style. Click **Finish**.
13. To distinguish the two kinds of sum functions, select the four summary fields which use RunningSum as the aggregate function and set their Background property to Green.

	XXXXXXXXX...	Total
	#####	#####
XXXXXXXXX...	#####	#####
	#####	#####
Total	#####	#####

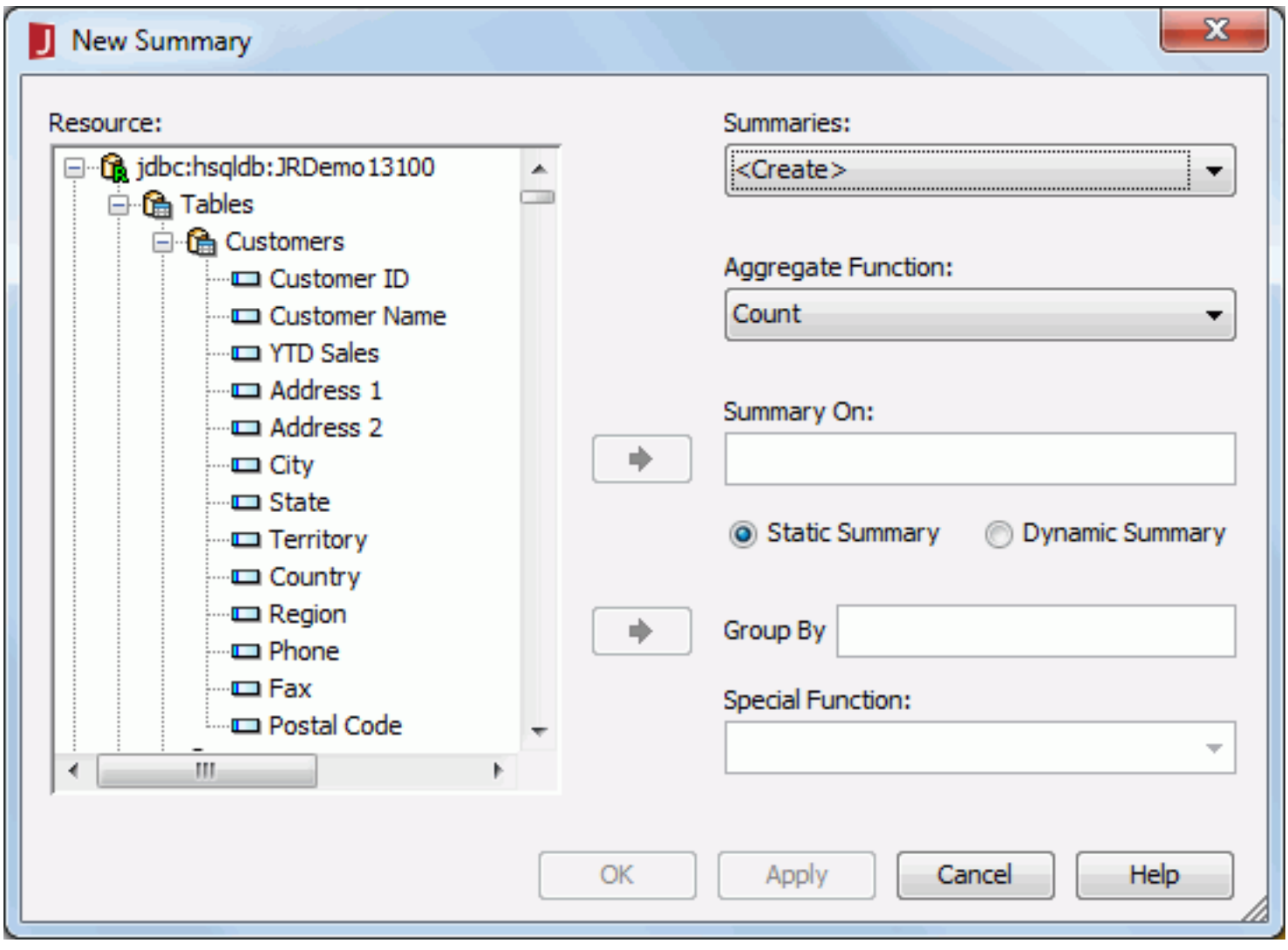
14. Click the **View** tab. The running sum values are highlighted in green for easier comparison. Note that the running functionality is only for columns of the crosstab, it does not do running totals on the rows.

	Asia-Pacific	Europe, Middle	Latin America	North America	Total
2014	69935	199321	33555	190175	492986
	69935	269256	302811	492986	492986
2015	81548	128790	140251	153481	504070
	81548	210338	350589	504070	504070
Total	151483	328111	173806	343656	997056
	151483	479594	653403	997056	997056

Creating a summary

To create a summary in a catalog, follow the steps below:

1. In JReport Designer, open the required catalog, then in the Catalog Browser, expand the data source in which you want to create the summary.
2. Right-click the **Summaries** node and click **Add Summary** on the shortcut menu. The New Summary dialog appears.



3. Select **<Create>** in the Summaries drop-down list.
4. Select a function from the Function drop-down list for your field to be summed by.
5. Select a field for the summary to compute from the Resource box, then click next to Summary On.
6. If required, specify to which group the summary will be applied.
   - If Static Summary is checked, select the field from the Resource box and then click next to Group By. And if the field you select is of Numeric/String/Date/Time type, you can further specify to which level the data will be grouped by by selecting a special function from the Special Function drop-down list. If Customize is selected, you can set the function by your own in the Customized Function dialog.
   - If Dynamic Summary is checked, first select Up or Down from the drop-down list, then in the combo box, select a value or input an integer, which should be between 0 to 127, to specify the group on which the summary will take effect.

     For example, if you select Up or Down and input 0 in the combo box and insert this summary into any group, the summary will take effect on this group; if you select Up and input 1 and insert this summary to a group, the summary will take effect on the group higher than the group where the summary is inserted, and if you specify Down and 1, it will take effect on the lower group; and the rest may be deduced by analogy.

     Notes:

     - If the dynamic summary is to be used in a chart, when defining the summary, you must choose Down from the Group By drop-down list; if it is to be inserted to other data containers, the summary must be defined on Up level.
     - When you insert a dynamic summary into a report and there is no corresponding group level to match settings of the dynamic summary, an error message will be generated.
     - If you want to insert a summary defined with a special function into a table or banded object, there must be a group with the same special function in the table or banded object.
7. Click the **Apply** button to add the new summary to the catalog. The Summary Name dialog will then be displayed, asking you to name the summary. Enter a name as required, and then click **OK**.
8. Click the **OK** button in the New Summary dialog to close the dialog.

Using summaries in reports

When you have created some summaries in a catalog, you can then use them as components in reports. For details, refer to the topic Summary fields.

Formulas

Formulas are objects that are computed at runtime, which allow you to manipulate field data by performing calculations on it. They control the data that will be displayed, and can even create new data not directly available from the database.

A formula is a symbolic statement of the manipulations that are to be performed on certain data before it is printed onto your report. For example, if your report contains an @Sales field and an @Cost field, and you want to create an @GrossProfit formula, you can designate its value as @Sales - @Cost. @GrossProfit is a simple formula that tells JReport to subtract the value of the @Cost field from the value of the @Sales field, and then to print the result. Formulas can be used to calculate numeric values, compare one value to another and select alternative actions based on the comparison. Also, they can be used to join multiple text strings into a single string. In general, formulas can be used to:

- calculate information you cannot directly obtain from database data fields.
- compare data.
- join text with data.
- convert data from one form to another.
- enhance formatting options with text strings.
- do other things that customize your reports.

The following are some specific topics about JReport formulas:

- Formula levels
- Creating a formula
- Formula examples
- User defined formula functions
- DBArray in formulas
- Using formulas in reports
- Using formulas to control object properties

Notes:

- You can only use JDK (not JRE) to compile formulas created in JReport.
- JReport formula searches for javac based on the java.home option, but this way may fail under some user environments. To ensure the formulas can work, you are recommended to use the javac_path option to specify the full path of javac yourself, for example, set `-Djavac_path=c:\jdk1.6.0_17\bin\javac.exe` in the startup file of JReport.

See an example: The SampleComponents catalog, included with JReport Designer, contains reports that have examples of how formulas could be used in a report. For the example, open `<install_root>\Demo\Reports\SampleComponents\UsingFormula.cls`.

Formula levels

Formulas can be categorized into 6 levels. These levels determine when the actual statements in your formula will be executed. JReport Engine will make several passes over the data to determine when formulas are executed. Even though an individual formula appears to be a certain level, if it references anything in the higher level then it also becomes the higher level.

- **Constant level formula**
  A constant level formula is a formula that does not refer to any DBFields, summaries, global variables, page level special fields (the value of the special field is ready at the time when the report result is generated) or other non-constant formulas. For example, `return CurrentDate()` is a constant formula. JReport will execute these formulas before fetching data from the dataset, so if you place it in the report header, it will be executed immediately as it is encountered.

- **Record level pass one formula**
  A formula is a record level pass one formula if and only if the following are true:
  - Refers to a DBField or any other formula which references a DBField.
  - Does not refer to any page level special fields or formulas which refer to page level special fields.
  - Does not refer to any global variable or any record level pass two formulas
  - Does not refer to a summary field.

  Record level pass one formulas are executed on each record immediately after the record is fetched from the dataset. They are often used in the detail panel of banded reports and tables such as calculating a value based on other DBFields such as `return @PRICE - @COST`.

- **Record level pass two formula**
  A formula is a record level pass two formula if and only if the following are true:
  - Refers to a DBField and a summary.
  - Refers to any other record level pass two formula.

  Record level pass two formulas are executed on each record of the dataset but only after the data has been grouped. For example, if a record level pass two formula is placed in the banded header, the group is considered to be the entire report. If it is placed in a group header such as Category, it will be the value after all of the detail rows have been processed for the group. For example, you might want to print in a group footer the group name and total such as `return "Total for " + @CATEGORY + ": " + @Sum_Total;`.

- **Group level formula**
  A formula is a group level formula if it refers to only summaries or other group level formulas. For example, if you have a summary called Sum_Total and a summary called Count_Detail, you could write a group level formula as `return @Sum_Total / @Count_Detail` to print the average total sales on each line of the order.

  These formulas are executed when breaking the group on which the summaries are defined which could be a group or the entire report for a report level summary in the banded header or banded footer.

- **Report global and global formulas**
  If a formula refers to a global variable or report global variable and does not refer to any page level special fields, there are several rules for when the formula will be calculated.
  - Report Header and Group Header: Formula will be calculated when the group starts.
  - Page Header, Page Footer and Detail: Formula will be calculated on each record after the data has been grouped the same as record level pass two formulas.
  - Report Footer and Group Footer: Formula will be calculated when ending the group or report.

  These formulas will be calculated based on all of the records of the group or report. For example, if you are calculating an accumulated total that you display in a report footer or group footer, then it will work as expected. For example, you may want to calculate a total margin such as:

  ```
  global number GrossProfit = 0;
  GrossProfit = GrossProfit + (@Price - @Cost);
  return GrossProfit;
  ```

  However, if you are showing a running total in the detail, you will see the group or report total on every line rather than the expected running total. The solution is to convert the formula to a page level formula by adding a reference to any page level special fields (see below).

- **Page level formula**
  All of the other formula types are converted to a page level formula if the formula refers to next(), prev() or any of the page level special fields. Page level special fields include FetchDate, FetchTime, RecordNumber, PageNumber, PrintDate, and PrintTime.

  The page level formulas are calculated in the exact sequence of the panels as they are laid out in JReport Engine. Thus if a page level formula includes any of the page level special fields, it will be calculated in the detail or group header and footer exactly in the order they are laid out. In the above example, to do a running total on each detail the code is as follows:

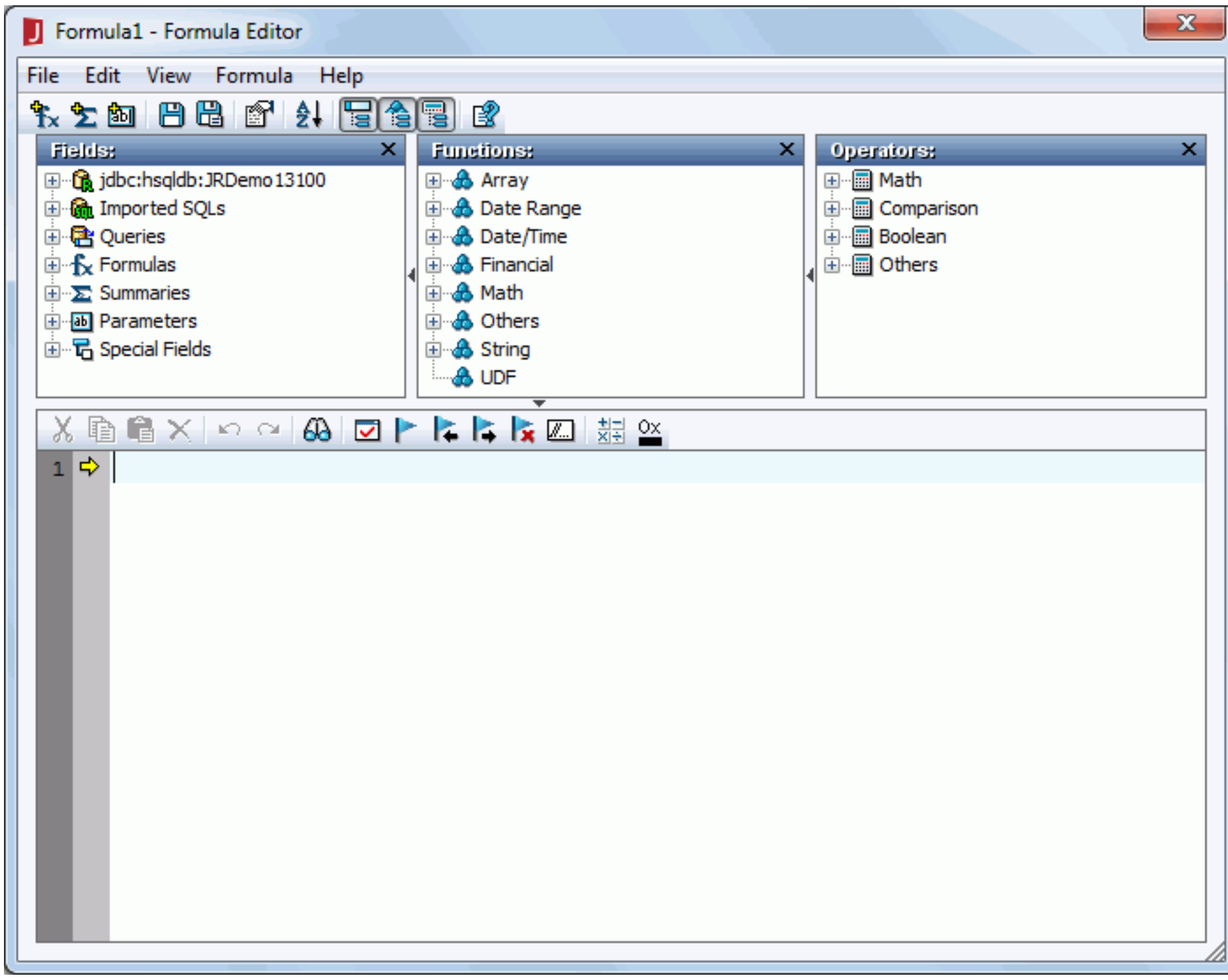
  ```
  PageNumber;
  global number GrossProfit = 0;
  PageNumber;
  GrossProfit = GrossProfit + (@Price - @Cost);
  return GrossProfit;
  ```

  PageNumber also controls the sequence of calculations around page breaks. If PageNumber is referenced, the calculation will be done before the page break. If PageNumber is not referenced, the calculation will be done after the page break. This is important when using formulas for setting properties.

Creating a formula

To create a formula in catalog, follow the steps below:

1. In JReport Designer, open the catalog to which you want to add the formula.
2. In the Data tab of the Catalog Browser, expand the data source in which to create the formula, then right-click the **Formulas** node, and then select **Add Formula** from the shortcut menu.
3. Enter a name for the formula in the Input Formula Name dialog and click **OK**. The Formula Editor window appears.



4. Compose the formula by selecting the required fields, functions and operators from the Fields, Functions and Operators panels. You can also write the formula by yourself in the editing panel.
5. To comment a line, click the **Comment/Uncomment** button on the toolbar.
6. If you want to bookmark a line so that it can be searched easily later, click the **Bookmark** button.
7. Click the **Check** button to check whether or not the syntax of your formula is correct.
8. When done, click the **OK** button to add the formula.

Notes:

- You should have some knowledge of the formula syntax before you can successfully compose a formula with no errors. To learn the formula syntax, refer to topics in Formula syntax.
- If you refer to any field in the formula, the reference name for that field will be prefixed with an @ sign. If the field name contains spaces, the reference name in formula will be quoted with double-quotation marks (""). For example, if the field name is Customer Name, then the reference name will be @"Customer Name".
- When formulas refer display names or mapping names, the names should not contain any of below characters if the names are not quoted by double-quotation marks "":

  "~", "`", "!", "@", "#", "$", "%", "^", "&", "*", "(", ")", "-", "+", "=", "{", "}", "[", "]", "|", "\\", ":", ";", "\", " ' ", "<", ",", ">", ".", "?", "/"

  Examples:

  - Expression @Customer#; will cause a syntax error. But @"Customer#" is ok.
  - If a field has the display name Category.Measure, when adding it to a formula, quote it as "Category.Measure" or "Category"."Measure".
- The number of the "if-else" statements in a formula is limited to 190. When this number is reached, you should use the "select-case" statement instead.
- In the Fields panel, if the summaries or parameters you want to use have not been predefined in the catalog, you can create them by your own by clicking the **New Summary** button or the **New Parameter** button on the toolbar (for details, see Creating a summary and Creating a parameter).

Formula examples

The following are some formula examples.

Example 1: TerritoryUSA

```
if (@State in ["CT", "ME", "MA", "NH", "NY", "RI", "VT"]) {
  "NorthEast, USA";}
else if ( @State in ["DC", "DE", "MD", "NJ", "PA", "VA"]) {
  "MidAtlantic, USA";}
else if ( @State in ["CO", "ID", "KS", "MT", "NE", "NV", "NM", "ND", "SD", "UT"]) {
  "Rockies, USA";}
else if ( @State in ["AZ", "AR", "LA", "MO", "OK", "TX"]) {
  "South, USA";}
else if ( @State in ["AL", "FL", "GA", "MS", "NC", "SC"]) {
  "SouthEast, USA";}
else if ( @State in ["AK", "CA", "HI", "OR", "WA"]) {
  "West, USA";}
else if ( @State in ["IL", "IN", "LA", "KY", "MI", "MN", "OH", "TN", "WI", "WV", "WY"]) {
  "MidWest, USA";}
```

Purpose: In your sales report, you want JReport to print out which territory the customer belongs to according to the abbreviated state names stored in the State field.

Explanation: In the six else-if statements, JReport will retrieve the value of the State field, and compare it with the values in brackets []. If a value matches the value in brackets, the formula will return the territory name.

Example 2: DateToMonth

```
Number m = Month ( @"Order Date" ) ;
String str = "";
if ( m == 1 ) {
  str = " January Sales";}
else if ( m == 2 ) {
  str = " February Sales";}
else if ( m == 3 ) {
  str = " March Sales";}
else if( m == 4 ) {
  str = " April Sales";}
else if( m == 5 ) {
  str = " May Sales";}
else if ( m == 6 ) {
  str = " June Sales";}
else if ( m == 7 ) {
  str = " July Sales";}
else if( m == 8 ) {
  str = " August Sales";}
else if ( m == 9 ) {
  str = " September Sales";}
else if( m == 10 ) {
  str = " October Sales";}
else if ( m == 11 ) {
  str = " November Sales";}
else if ( m == 12 ) {
  str = " December Sales";}
```

Purpose: In your sales report, you want JReport to print out the month name of the orders, so that you can compare the orders of each month in a year.

Explanation: Since the database field Order Date is of the TimeStamp data type, the built-in function Month (DateTime) is first used to extract the month portion of Order Date. Then, the else-if statement is used to decide the month according to the return value of the function, and assigns a String value which contains a relative month name to the pre-declared string variable str.

Example 3: SectionInvisible

```
boolean s;
if (pagenumber==1){
  s=true;}
else {
  s=false;}
```

Purpose: This formula is used to control the Suppressed property of a certain section such as Page Header, so that the section will be suppressed on the first page of the report.

Explanation: Since it is only required to suppress the section on the first page, if only the Suppressed property of the section is set to true in the Report Inspector, then it will be suppressed on every page. In this formula, a Boolean type variable is first declared. Then, an else-if statement is used to decide whether or not the page is the first page. If the statement is true, true is assigned to "s". If the statement is false, false is assigned false to "s". In this formula, pagenumber is a special field.

Example 4: Running total of page

You can use the running total function to calculate the sum of one page. And to use the running total function, you have to create a set of formulas.

1. Create a formula named Formula1 as follows to initialize a global variable as the sum variable. And insert it into the BandedPageHeader panel. The reference to PageNumber forces the calculation to run after the page break. For more information, see Formula levels.

   ```
   pagenumber;
   global number pagetotal;
   pagetotal=0;
   ```

2. Create a formula named Formula2 as follows to accumulate the value. And insert it into the Detail panel.

   ```
   pagetotal=pagetotal+@"YTD Sales";
   ```

3. Create a formula named Formula3 as follows to display the sum of one page. And insert it into the BandedPageFooter panel.

   ```
   return pagetotal;
   ```

Example 5: Running total of group

This example will accumulate the total of the successful and failed shipments for each group.

1. Create a formula named Formula1 as follows to initialize the value. And insert it into the group header.

   ```
   global integer iFail;
   global integer iSuccess;
   iFail=0;
   iSuccess=0;
   ```

2. Create a formula named Formula2 as follows to accumulate the value of the successful and failed shipments, and insert it into the Detail panel.

   ```
   if(@Shipped) {
     iSuccess=iSuccess+1;
     iFail=iFail;
   }
   else {
     iSuccess=iSuccess;
     iFail=iFail+1;
   }
   ```

3. Create a formula named Formula3 as follows to return the successful shipments and the failed shipments, and insert it into the group footer.

   ```
   return "Total successful shipments is "+ iSuccess +" and total failed shipments is "+
   iFail;
   ```

User defined formula functions

When the built-in functions do not satisfy your requirements, you can design your own formula functions. JReport provides you with user defined formula (UDF) functions. You can first write the code to implement a function, and then import it as a class when you build a formula.

However, when you want to develop a Java file to implement a function, only the following data types can be used for interacting with JReport products (pass parameters and return parameters): DbBigInt, DbDouble, fCurrency, DbBit (using for boolean type), DbChar (using for String), DbDate, DbTime, DbTimestamp, DbBinary, fText (using for Long VarChar), fImage (using for long VarBinary), fIntArray, fNumArray, fCurArray, fBoolArray, fStrArray, fDateArray, fTimeArray, fDateTimeArray, fBinaryArray, fTextArray, fImageArray, fIntRange, fNumRange, fCurRange, fBoolRange, fStrRange, fDateRange, fTimeRange, fDateTimeRange. All the data types that start with 'f' belong to the package jet.formula.*. The other data types belong to the package jet.connect.*. For details, refer to JReport Javadoc jet.formula and jet.connect packages in `<install_root>\help\api`.

Importing UDF classes

To import a user defined formula class, follow the steps below:

1. Develop your own Java program that implements the functions you need.
2. Compile the Java program and add it to the ADDCLASSPATH variable of the batch file setenv.bat in `<install_root>\bin`.
3. Start JReport Designer and open the Formula Editor.
4. Import your class file in either of the following two ways:
   - **Using the "import" statement:**

     `import` **`instName`** `from "`**`ClassName`**`";`

     Where,

     - instName is the instance name that you give when you load the class. It is a global variable, you can use it in your formulas in the current data source.
     - ClassName is the class name defined in your class file. If the class file is put in the package jet.formula.javaformula, only the class name is required. However, if it is in your own package, you must specify the qualify name of the class.

       For example, the following declaration assumes the class MyFunctions is in package jet.formula.javaformula:

       ```
       import myfunc from "MyFunctions";
       ```

       And the following declaration allows user defined UDF class in any package, supposing the class is com.mycom.MyFunctions:

       ```
       import myfunc from "com.mycom.MyFunctions";
       ```

       Notes:

       - You should write the import statement only without anything else in a separate formula so as to avoid importing unwanted things.
       - You should avoid giving the same instance name when importing the same UDF.
5. **Using dialog:**
   a. On the File menu of the Formula Editor, click **Import UDF Classes**.
   b. In the UDF Classes dialog, click the button ✚.
   c. In the Import UDF Class dialog, input the class name with full package path and then click **OK**.
   d. The class with package path is displayed. Click **OK**.

Using UDF functions

When your class is successfully imported, the functions defined in the class will be displayed in the UDF node in the Functions panel of the Formula Editor. You can then call the functions in a formula by double-clicking them or using the following statement:

`instName.`**`MethodName`**`(parameters);`

Where, MethodName is the method name in your class. You can call methods in your class as many times as you want by loading this class only once.

If you want to view the formula references of any UDF function, select the function and click **Formula References** on the Formula menu, or right-click the function and click **Formula References** on the shortcut menu, then the UDF References dialog will be displayed, showing all the formulas that reference the UDF function if there are.

Example of using UDF functions

In this example, a demo Java program MyFunctions.java which is put in the package jet.formula.javaformula will be used to illustrate how to write record data to a file using UDF functions. You can get the Java source file from `<install_root>\help\samples\APIUDFormula\jet\formula\javaformula`.

Take the following steps:

1. Compile this Java file to generate the class file.

   ```
   javac -classpath "<install_root>\lib\JREngine.jar;<install_root>\lib\report.jar;"
   MyFunctions.java
   ```

2. Edit the batch file setenv.bat in `<install_root>\bin`.

   When you add class path, just add the root path. For example, suppose that the class file is located in `C:\JReport\Designer\help\samples\APIUDFormula\jet\formula\javaformula`, you can append `C:\JReport\Designer\help\samples\APIUDFormula` into the ADDCLASSPATH variable in the batch file.

3. Start JReport Designer with the modified batch file.
4. Create a new formula to load this class and open a file on disk by calling the methods in MyFunctions.java. Here we define the formula (fmlA) to load the class and call the method to open a temp file:

   ```
   pagenumber;
   import myfunc from "MyFunctions";
   global integer filehandle = myfunc.openfile("e:\\test\\data.txt", false);
   ```

   Note: If the class file is in your own package com.mycom other than jet.formula.javaformula, import the class in a formula as follows:

   ```
   import myfunc from "com.mycom.MyFunctions";
   ```

5. Compose a formula (fmlB) to call the method to write data into the temp file:

   ```
   string contents ="";
   contents = @"Customer Name"+"\t"+ @Country+"\t"+ @Phone + "\t"+ PageNumber + "\n";
   myfunc.write(filehandle, contents);
   ```

   The return value of the last statement is the formula result.

6. Compose another formula (fmlC) to close the temp file:

   ```
   PageNumber;
   myfunc.closefile(filehandle);
   ```

   If the statement PageNumber is added to the formula, the formula will be calculated after the page break, which means that the calculation point is controlled. For more information, see Formula levels.

7. We have now defined three formulas (fmlA, fmlB, fmlC) in the above three steps. You can use it in a report as with a normal one.

Note: Now in JReport, the '8859-1' encoding is not required any more and the UDF functions you define should be able to return the correct unicode strings. Thus, if you have UDF fuctions created in previous JReport versions which still return '8859-1' encoded strings, you may need to modify them to make them return unicode strings.

DBArray in formulas

JReport Designer supports a multiple-value field, which contains a set of values in a single field, in a formula. Using the multiple value field is the same as using an array in a formula. You can refer to a field with multiple values as with a normal field in a formula, such as @FIELDNAME, and then use the array functions that JReport Designer provides in the Formula Editor to manipulate the values in it. JReport Designer first converts this field into a normal array, and then manipulates it in the usual way.

For example, normally an array in a formula is manually defined as follows:

```
Integer myArray[4]=[1, 2, 3, 4]; //Datatype ArrayName[ArraySize]=[ArrayContent];
```

The array may be referred to in this way:

```
integer myInt=3;
myInt in myArray;
```

If a field, named myField, for example, contains a similar array value, it can be referred to similarly:

```
integer myInt=3;
myInt in @myField;
```

Using formulas in reports

When you have created some formulas in a catalog, you can then use them as component in reports. For details, refer to the topic Formula fields.

Using formulas to control object properties

One of the biggest features that formulas have is that you can use formulas to control the property values of objects at runtime. This provides you much more control, allowing you to customize your report and have flexibility to control the way objects are displayed based on the data values. For web reports and library components, only the Invisible property can be controlled by formulas.

When you edit the property values of an object in the Report Inspector, you may notice that if you click some properties, there will be a button appearing on the right of the drop-down arrow, and when you click this button, it will change to be another button and also the value cell will become empty. Then when you click the drop-down arrow, a drop-down menu will be listed. If you select one item in the list, the property will be controlled according to the return value of the formula. Meanwhile, you will also find the button next to the value text boxes of some options in certain dialogs (like the Display Type dialog). You can use formulas to control the options in the same way you do to properties.

Notes:

- Usually when you open the drop-down list of this kind of property, there is no formula to be selected directly unless:
  - For page report
    You have created some formulas in the same catalog before building the report, and the formulas are created in the same data source connection as the query on which your report is going to built. You can also click the <Create Formula> item in the drop-down list to create one as required. For a crosstab in a page report, if you have created some crosstab formulas in the crosstab, they are also available in the drop-down list. You can click the <Create Crosstab Formula> item in the drop-down list to create one too.
  - For web report and library component
    You have created some dynamic formulas in the corresponding business view. For details, see Using formulas to control showing or hiding components.
- When you are composing a new formula that directly binds to a property using the Report Inspector, make sure the data type of the return value is the same as the property type. Otherwise, the new formula will not be bound to this property, although it will be created in the catalog data source.

Example: Changing the banded panel background color at runtime

This example uses the formula MColor to control the Background property of the detail panel of a banded report at runtime. You can find this formula in Data Source 1 of the catalog SampleReports. The following is the content of the formula:

```
String sc = "" + (@MColor_RecordNumber *10 );
Number idx = InStr(".",sc);
if(idx != -1)
          sc = Left(sc, idx);
if(Length(sc) < 2)
          sc = "0" + sc;
String s = "0x00" + sc + "7f";
return s
```

In this example we will vary the middle hexadecimal digits of the RGB color values. The Red value is fixed at 0x00, the Blue value is fixed at 0x7f. The Green value will change according to the number of records displayed. Record 1 will use color 0x00107f, Record 2 will use 0x00207f, and so on.

Take the following steps:

1. Make sure the currently open catalog is SampleReports. If not, open the catalog.
2. Create a standard banded report with a query stored in Data Source 1 of the catalog.
3. In the design area, select the detail panel (DP) of the banded object.
4. In the Properties panel of the Report Inspector, click the value cell of the property Background, and click . From the drop-down list, select **MColor**.
5. Click the **View** tab to view the report. The report will be shown somewhat as follows, depending on the report you use.

Product Name	Discount	Quantity	Unit Price
Macadamia Nut	.00	40	9.00
Macadamia Nut	.00	10	9.00
rish Mocha Crosse	.00	20	9.00
Jamaican Rum	.00	15	9.00
Cappuccino	.00	20	9.00
rish Mocha Crosse	.00	30	9.00
Irish Crosse	.00	15	9.00
rish Mocha Crosse	.00	10	9.00
French Vanilla	.00	10	9.00
Hazelnut Crosse	.00	10	9.00
Butternut Rum	.00	15	9.00
Chocolate Hazelnu	.00	10	9.00
Almond Tosted	.00	10	9.00
rish Mocha Crosse	.00	30	9.00
French Vanilla	.00	20	9.00
Hazelnut Crosse	.00	15	9.00
Vanilla Hazelnut	.00	30	9.00
Almond Tosted	.00	25	9.00
Cappuccino	.00	25	9.00

Business Views

Business views provide report users with business-oriented views of their data and formulas in query based data sources. They contain database connections and relationships between view elements that are required for creating reports and library components.

Business views shield report end users from having to understand physical structure, and enable the easy building of complex reports containing multiple components. Business views enable end users to build reports and analyze data based on a set of view elements that they can easily understand. For example, they do not need to worry about where the data source is from (SQL, stored procedure, view), or what the data source is (Oracle, Sybase, XML). Business views also enable IT professionals to maintain control of business data and to ensure its integrity, while presenting end users with an intuitive view of the underlying data structures.

The following detail information is available:

- Business view elements
- Adding a business view
- Editing a business view
- Converting query-based components to business view-based

Note: A JReport Live license for JReport Designer is required in order to use this feature. If you do not have the license, contact your Jinfonet Software account manager to obtain it.

Business view elements

The following are elements contained in business view.

Categories

Categories are the main tools for organizing data. They are components which correspond to particular collections of data in the data source. They are grouped in folders that are named to reflect information collections. A category can contain several subcategories.

Organizing data required by a user into multiple categories or multiple levels of categories is meaningful, and makes it easy to use. End users can use categories to find data elements they need to use without knowing the underlying table names although sometimes the table names are used as categories.

Group objects

Group objects are view elements that are the basis for organizing data in a report. They present the availability and key performance of data, and characteristically return text data or dates, and answer the following question: who, when, what, where and which. A group object can be inserted wherever a field can be inserted, as a column or row in a crosstab, or as a group field or detail field in a table or as a category or series in a chart. They may be based on DBFields such as region and country or they could be based on formulas such as the year, quarter, or month portion of a date field.

Aggregation objects

Aggregation objects are numeric view elements that are calculated at runtime based on the group objects in the report. Aggregations can be used alone or calculated with other fields. An aggregation object can be inserted wherever summaries can be inserted. For example, aggregations can be inserted in the group header or footer panel of a table, as an aggregate field in a crosstab, or as the value in a chart. JReport calculates values based on the group level at which the aggregation object is inserted such as total sales, average order size and number of orders from a customer.

Detail objects

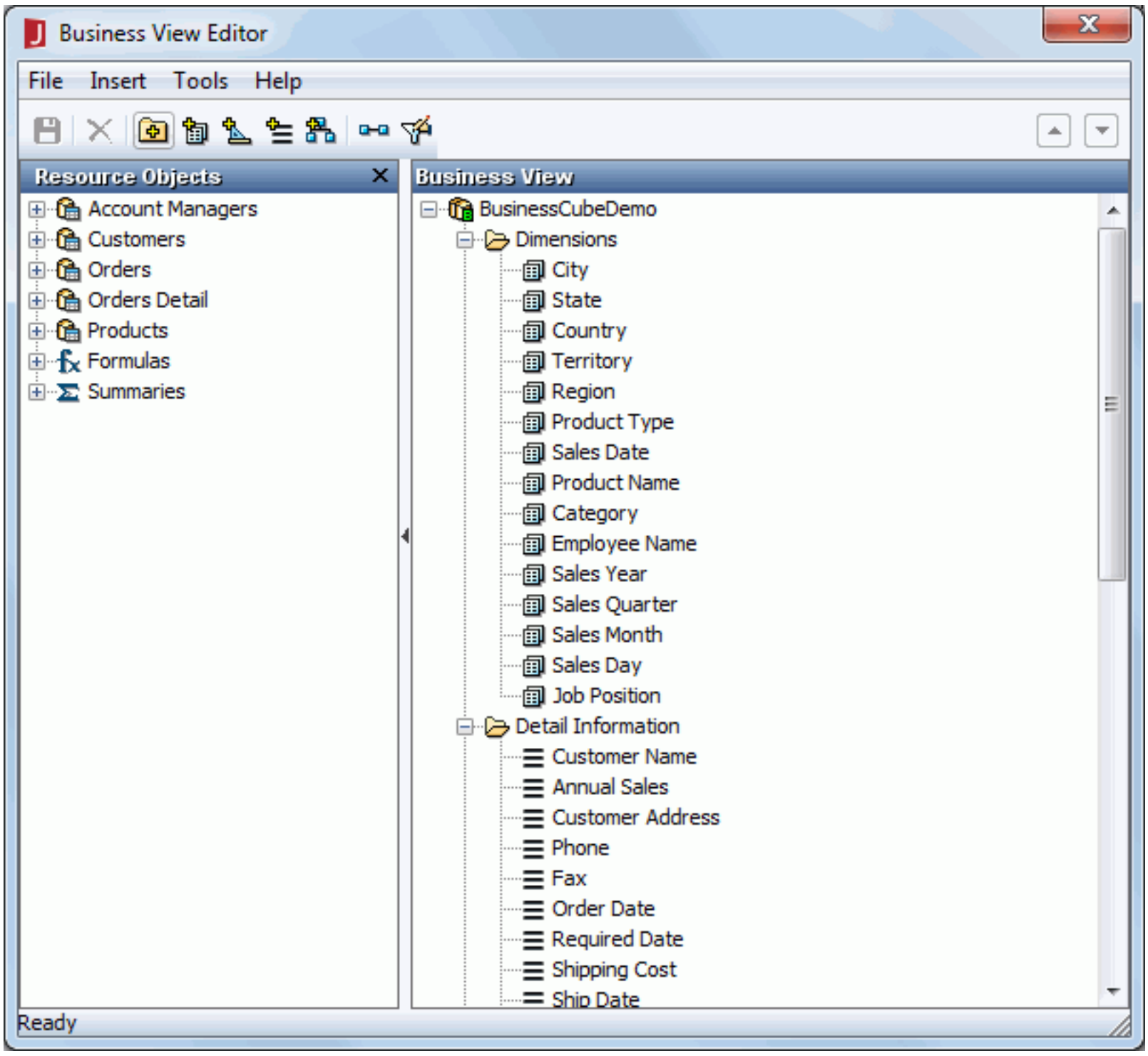
Detail objects provide additional information. It can be bound to a database field, group object, aggregation object or category/subcategory. Detail objects are used for displaying detailed field values such as address and phone number for a customer in a table. Detail objects cannot be used in crosstabs and charts and cannot be used for grouping in tables.

Adding a business view

To create a business view, follow the steps below:

1. In JReport Designer, click **File** > **Catalog Management** > **Open Catalog** to open the catalog to which you want to add the business view.
2. In the Data tab of the Catalog Browser, expand the desired data source of the catalog, right-click the **Business View** node and then click **Add Business View**.
3. In the Input Business View Name dialog, enter a name for the business view and click **OK**.
4. The Add Table/View/Query dialog is displayed. Add the resources based on which to create the business view. If the catalog data source is connected with multiple connections, you can mash-up multiple data resources in the business view that come from all these connections, including tables, views, synonyms, queries, imported SQLs, stored procedures and user defined data sources. When a query, imported SQL, stored procedure or user defined data source is added, it will be added as a single table with all of its columns. When two resources (for example, a table and a view) use the same name, they cannot be added to the right box at the same time, and when a table is already contained in a query, you cannot add the table and the query at the same time.

   If the current catalog data source contains JDBC connections, the Include Added Tables/Views option is available. Check it if you want to show the tables, views and synonyms that have already been added to the query in the All Tables/Views/Queries box. You can then add the tables, views and synonyms of the JDBC connections to the query as many times as you want by providing different names for the tables, views and synonyms each time you add them in the Input Table Name dialog.
5. If more than one resource is added for the business view, the Query Editor window appears, in which you can select the columns in each table that you want to use for the business view, add or delete the tables used for creating the business view, set up joins between the tables, create formula fields to use in the business view or add filters to narrow data used for the business view in the same way you edit a query. When done, click **OK** in the Query Editor to apply the changes.
6. The Business View Editor window appears.



7. Add elements to the business view.
8. Define hierarchies on the business view to allow end users to drill report data to particular groups.
9. Create some predefined filters on the business view to narrow down records.
10. Configure member-level security for the business view if necessary.
11. Click the **Save** button on the toolbar to save the business view.
12. Click **File** > **Close** to close the window. The business view will now have been added to the catalog.

Editing a business view

You can edit a business view in a catalog at any time. For example, rename it, add more elements to it, redefine the filters and so on. You can also delete the business view if it is no longer needed.

To rename a business view:

1. In the Catalog Browser resource tree, select the business view, right-click it and select **Rename** from the shortcut menu.
2. The name text field of the business view will become editable. Input the new name.
3. Click **Enter** on your keyboard or anywhere outside the name text field.
4. A warning message is displayed. Click **OK** to confirm the change.

To delete a business view:

1. In the Catalog Browser resource tree, select the business view, right-click it and select **Delete** from the shortcut menu.
2. A warning message is displayed. Click **OK** to confirm the removal.

The following topics show more about editing a business view:

- Adding/Editing elements
- Creating predefined filters
- Defining hierarchies
- Setting up and editing joins

Adding/Editing elements

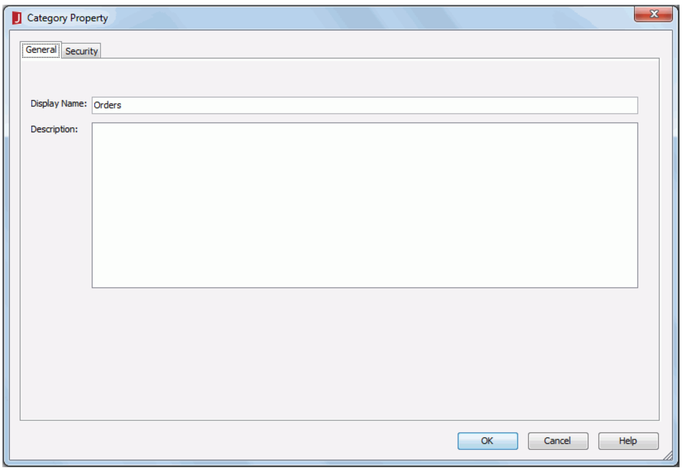
You can add more elements to a business view, or edit any of the existing elements at any time to suit your requirements.

Adding a category

To add a category, follow the steps below:

1. Do one of the following:
   - In the Catalog Browser resource tree, right-click an existing category in a business view and select **Add Category** on the shortcut menu.
   - In the Business View Editor window, select the business view or an existing category in the Business View panel, then,
     - Right-click and select **Add Category** from the shortcut menu.
     - Click **Insert** > **Category** on the menu bar.
     - Click the **Insert Category** button on the toolbar.

   The Category Property dialog appears.



2. In the General tab, set the display name of the category in the Display Name text field. It must be unique at the current resource level.
3. Type a description for the category in the Description text box if required, which will be shown when you hover the mouse pointer over the category in the resources panel of Web Report Studio and Page Report Studio.
4. Click the **Security** tab and set permission to group elements as required (for details, see Configuring member-level security).
5. Click **OK**. A new category will be added to the business view or the selected category. You can then add view elements to it.

Editing a category

To edit an existing category:

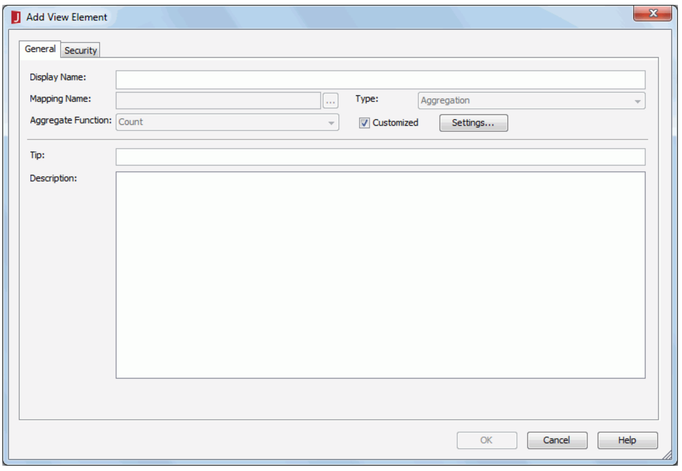
1. Do either of the following to open the Category Property dialog.
   - In the Catalog Browser resource tree, select the category you want to edit, right-click it and click **Edit Category** on the shortcut menu.
   - In the Catalog Browser resource tree, select the business view, right-click it and click **Edit** on the shortcut menu, then in the Business View Editor window, select the category you want to edit in the Business View panel, right-click it and select **Edit Category** from the shortcut menu.
2. In the General tab, edit the general information of the category.
3. In the Security tab, edit the security information.
4. When done, click OK to accept the changes.

Adding a view element

To add a view element, take either of the following two ways:

- **Adding via dialog**
  1. Do one of the following:
     - In the Catalog Browser resource tree, select an existing category, right-click it and click **Add View Element** on the shortcut menu.
     - In the Catalog Browser resource tree, select the business view, right-click it and click **Edit** on the shortcut menu, then in the Business View Editor window, select the business view or any existing category into which you want to add the view element in the Business View panel, then,
       - Right-click and select **Add View Element** from the shortcut menu.
       - Click **Insert** > **Group**/**Aggregation**/**Detail** on the menu bar.
       - Click the **Insert Group** button, **Insert Aggregation** button, or **Insert Detail** button on the toolbar.

     The Add View Element dialog appears.



  2. In the General tab, decide the type of the view element by selecting the corresponding item from the Type drop-down list.
  3. Click the button ... next to the Mapping Name text field to specify the mapping field of the view element.
  4. Specify the display name of the view element in the Display Name text field.

     By default, the display name is the same as the mapping name. You can change it according to your requirements. The display name must be unique at the current rsource level.
  5. If you choose to create an aggregation object, click the **Aggregate Function** drop-down list to specify the aggregate function with which to summarize the mapping field. Check the **Customized** option if you want to create a custom aggregation, then click the **Settings** button to open the Formula Editor to compose a formula using resources in the current business view, which can return an aggregation.
  6. Specify the tooltip and description of the view element in the Tip and Description text boxes respectively if required, which will be shown when you hover the mouse pointer over the view element in the Resources panel of Web Report Studio.
  7. Click the **Security** tab and set permission to group elements as required (for details, see Configuring member-level security).
  8. When done, click **OK** to close the dialog.
- **Adding by dragging**

  In the Business View Editor window, drag one or more tables, DBFields, formulas, or summaries from the Resource Objects panel to the business view or a category. Then, all DBFields in the tables, the DBFields, formulas, or summaries will be added as view elements into the business view or the category you drop them into, and JReport will determine their types automatically.

Editing a view element

To edit an existing view element:

1. Do either of the following to open the Edit View element dialog.
   - In the Catalog Browser resource tree, select the view element you want to edit, right-click it and click **Edit** on the shortcut menu.
   - In the Catalog Browser resource tree, select the business view, right-click it and click **Edit** on the shortcut menu, then in the Business View Editor window, select the view element you want to edit in the right panel, right-click it and select **Edit** from the shortcut menu.
2. In the General tab, edit the general information of the view element.
3. In the Security tab, edit the security information.
4. When done, click **OK** to accept the changes.

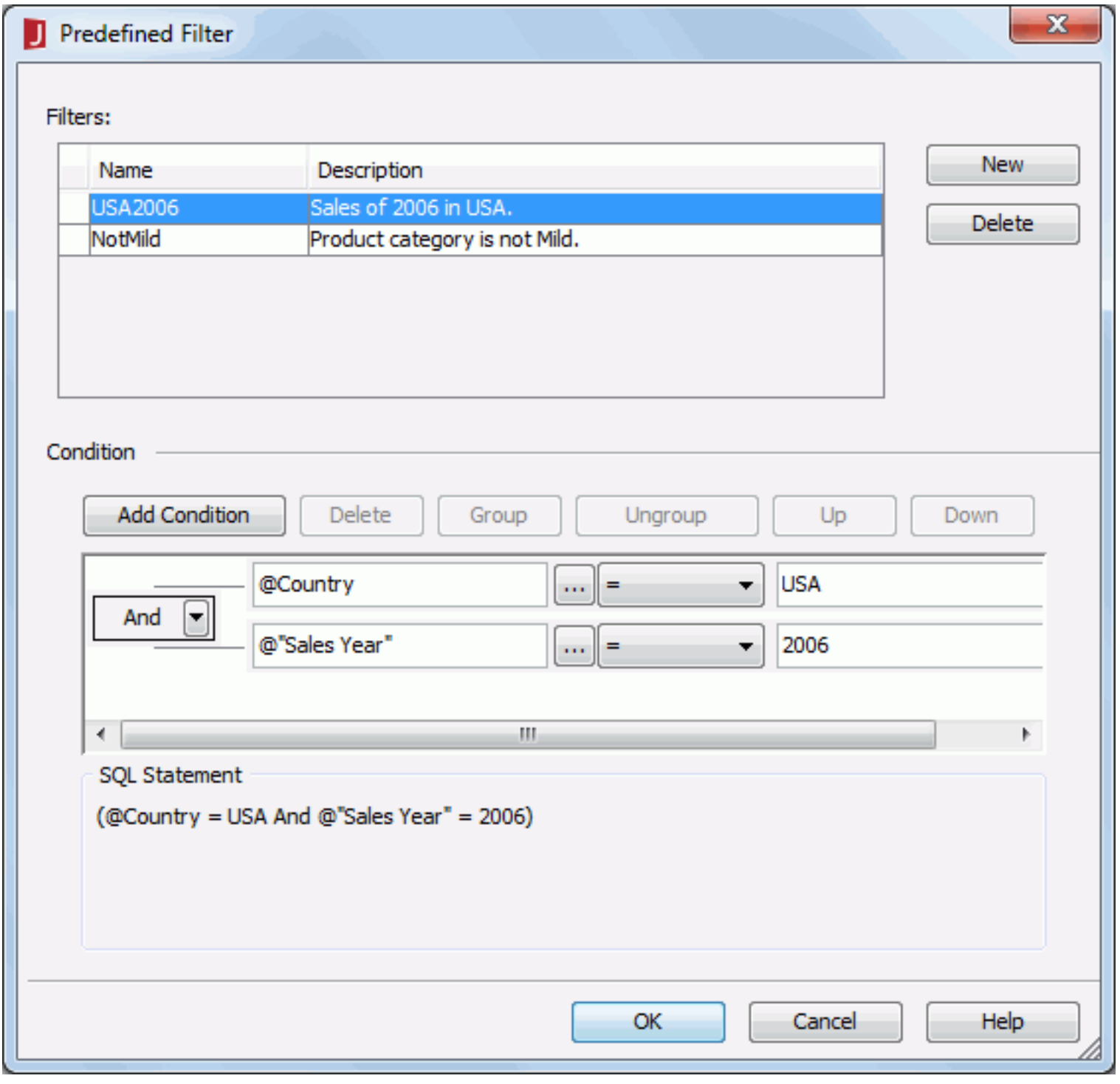
Notes:

- The display names of the view elements in a category must be unique, and the category display names also should be unique in a business view.
- When using a custom aggregation in a data component, you need to make sure the group objects referenced in the custom aggregation are also be added as group by fields, column/row headers or chart axis fields in the data component.

Creating predefined filters

You can predefine filters in business views for end users to choose when they design or modify reports and library components. The following example shows the basic usage.

1. Open the catalog file SampleReports.cat in `<install_root>\Demo\Reports\SampleReports`.
2. In the Catalog Browser, go to Data Source 1 > Business View, right-click WorldWideSalesBV, and select Edit from the shortcut menu.
3. In the Business View Editor, click **Tools** > **Predefined Filter** on the menu bar or click the **Predefined Filter** button on the toolbar. The Predefined Filter dialog appears.



4. Click the **New** button, double-click in the Name cell of the filter and change its name to **QuantityFilter**, then in the Condition panel, click the **Add Condition** button and define the filter as **Quantity>=3000**.
5. Click the **New** button to add another filter and name it **RegionFilter**, then define the filter as **Region=North America**.
6. Click **OK** in the Predefined Filter dialog to save the filters. You can then apply them when creating or editing reports and library components based on the business view in JReport Designer or in report studio after you publish the corresponding catalog to JReport Server.

Notes:

- When the resources that are used in some predefined filters are removed from the business view, the next time when you open the Predefined Filter dialog, you will be prompted with warning messages showing the details and the filters will be removed automatically from the dialog as well.
- The following SQL type of data cannot be sorted: Db.SQL_BINARY, Db.SQL_BLOB, Db.SQL_CLOB, Db.SQL_LONGVARCHAR, Db.SQL_LONGVARBINARY, Db.SQL_VARBINARY and Db.SQL_OTHER.

Defining hierarchies

Hierarchies can be defined in a business view to allow users to drill report data up and down to particular groups on JReport Server. A hierarchy is defined as a hierarchy category holding one group of group objects sharing a hierarchical relationship, following the order from the highest level to the lowest. For example, these group objects can be added into one hierarchy group called Dates: Sales Year, Sales Quarter, Sales Month, and Sales Date. When any of these group fields are used for grouping data in a table, crosstab or chart, end users can use the context menu to drill up or down to the next level. You can create any number of hierarchies such as dates, times, geography, product types, and so on.

The group objects added in a hierarchy are just references to the real group objects and therefore cannot be edited.

To define hierarchies in a business view, you can take either of the following two ways:

Defining a data hierarchy in the Business View Editor

1. In the Catalog Browser resource tree, select the business view, right-click it and click **Edit** on the shortcut menu.
2. In the Business View Editor, select the business view or any category in the Business View panel, then,
   - Right-click and select **Add Hierarchy** from the shortcut menu.
   - Click **Insert** > **Hierarchy** on the menu bar.
   - Click the **Insert Hierarchy** button on the toolbar.
3. In the Add Hierarchy dialog, provide a name and click **OK**. The name should be unique among the hierarchies in the business view.
4. A blank hierarchy will be added to the business view resource tree. You can then add group objects into the hierarchy as a hierarchical group by dragging and dropping.
5. Drag a group object either from the resource tree in the Resource Objects panel or from the business view resource tree in the Business View panel and then drop it into the hierarchy. The inserted position depends on the dropped position:
   - When you drop on the hierarchy root node, the group object will be placed to the bottom level of the hierarchy.
   - When on an existing level, the group object will be placed below it.
   - When between two levels, the group object will be placed between them.
6. To adjust the order of the group objects in the hierarchy, make use of the or button. The higher the position is, the higher the level is.
7. To remove a group object from the hierarchy, select it, then right-click and select **Remove from Hierarchy** on the shortcut menu.

Adding a data hierarchy in the Data tab of the Catalog Browser

1. In the Catalog Browser resource tree, right-click a category in a business view and select **Add Hierarchy** from the shortcut menu.
2. Provide a name in the Add Hierarchy dialog and then click **OK**. The name should be unique among the hierarchies in the business view.
3. A hierarchy will be added in the business view.
4. Follow the steps 5 to 7 in the above section to define the hierarchy in the Business View Editor.

Tip: Hierarchies can be added into any category in a business view and it does not make any difference where it is placed. For easy look-over, you can put them under the root category.

Setting up and editing joins

Sometimes you may need to modify the joins of a business view. To do this:

1. In the Catalog Browser resource tree, select the business view, right-click it and select **Edit** from the shortcut menu.
2. In the Business View Editor window, click **Tools** > **Join Editor** on the menu bar or the **Join Editor** button on the toolbar. The Query Editor window appears.
3. Add more tables to use in the business view if needed and set up the joins between the tables, then click **OK** to save the settings.

   For more information, refer to Adding and deleting tables in a query and Joining tables in a query.

Converting query-based components to business view-based

In order for a query-based report component created in JReport Designer, such as table, crosstab, chart, and banded object, to be able to perform analytic reporting actions on when you work with them in Page Report Studio, JReport needs to convert the component from query-based to business view-based.

This section describes how JReport performs the conversion work and what a report designer should pay attention to in order to increase the opportunity of success in the conversion.

When the conversion is required

The conversion is required when the Page Report Studio end user is trying to perform an analytic reporting action such as drilling through data on a table, crosstab, chart, or banded object which is created based on a query.

There is a report tab level property available in the Report Inspector of JReport Designer: Automatic Cube Initialization, which controls whether a dialog will be displayed to the user to convert the dataset to a business view when the user requests an analytic action. By default, this property is set to false which means the business view will not be automatically initialized until the user does an analysis action such as dragging a new field onto the report. After that action, the business view is initialized and normal drilling and other actions will work properly. If you expect users to use the actions such as drilling, you should set this to true so that the conversion will be done automatically when opening the report in Page Report Studio without the need for an explicit analysis action.

What makes the conversion successful

The success of the conversion lies in two aspects: there should be at least one business view in the query on which the report component is created, and the fields used in the component have corresponding view elements which have been defined in the business view.

To ensure success of the conversion, the report designer should avoid the cases illustrated in What cannot be converted.

Tip: The report designer can create a complete and full-function business view for the required query, in this way, all fields in the query are able to be converted since there are always corresponding view elements found in the complete business view. To create such a complete business view, create a group and a detail object for each DBField and formula, and create an aggregation for each summary.

How to convert

The following lists what types of fields in the query will be converted to what types of elements in the business view:

When Query Fields Are Used As	View Elements
Group by field | crosstab group | chart X axis | chart Z axis	Group
Crosstab aggregation	Aggregation
Chart Y axis & summary based on fields and function	Aggregation
Chart Y axis & a Number data type element	Detail object
Others & there are corresponding detail objects in the business view	Detail object
Others & there are not corresponding detail objects in the business view	Group

What cannot be converted

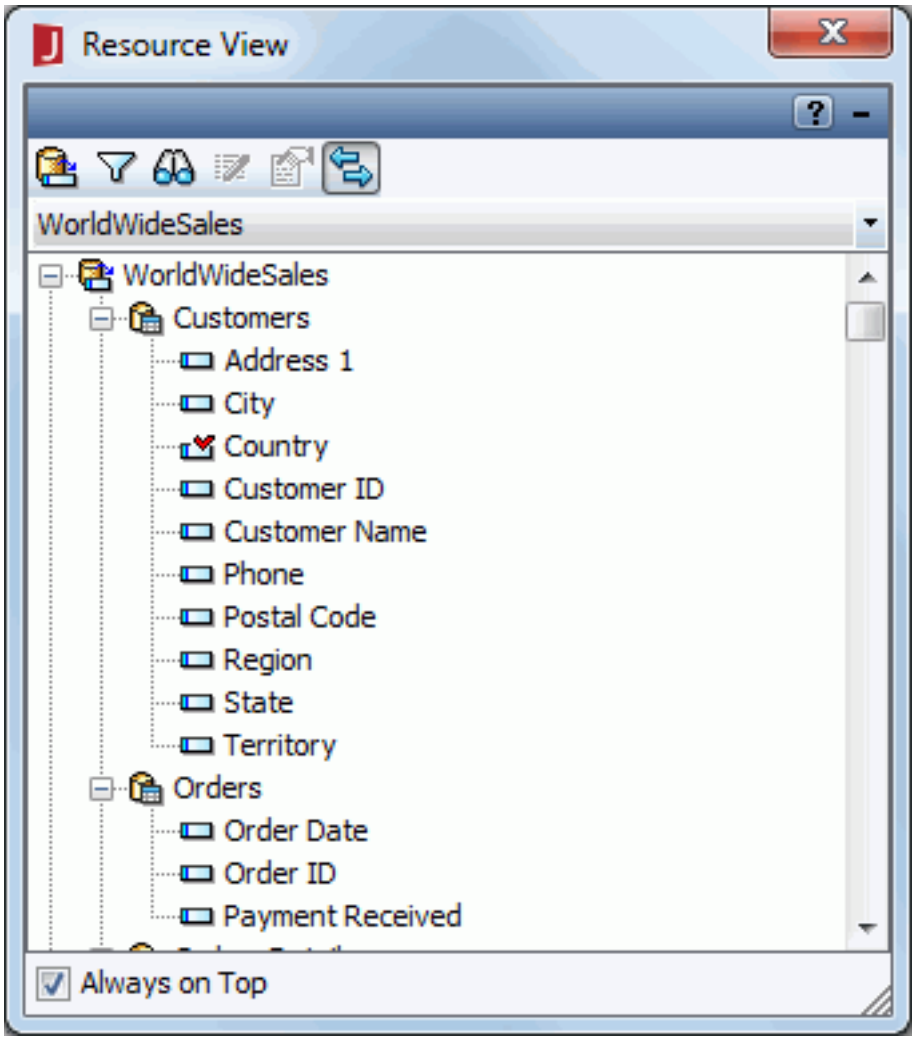
The following list shows the cases in a query-based component that do not support conversion. The list supposes that all the involved query fields have their corresponding view elements in the business view based on the query:

- Static summary (except that used as chart Y axis or current group by level).
- Dynamic summary (except when group by level = 0).
- Formula that references summaries.
- Page level formula, that is, the formula calculated after the whole report finishes rendering.
- Group by DBField or formula while the group has a special function.
- Group by parameter.
- Parameter on chart X axis | chart Y axis | chart Z axis.
- Chart in a banded object while the data of the chart is inherited.
- Table, crosstab, or banded object whose data is inherited.
- Table, crosstab, chart, or banded object when one of its children components cannot be converted (the children components exclude blank components and those having already been converted).

Datasets

A dataset is the set of data built from the result of a query, and can optionally have filters applied to it. A dataset contains not only DBFields, but also any formulas, summaries, and parameters that use the DBFields.

A dataset has a name, which by default is the name of the resource on which it is created. You can view items in a dataset by choosing the Dataset View mode from the toolbar of the Resource View panel.



When you create a data component, you specify whether a new dataset is created for it or an existing one is used. Having data components use an existing dataset whenever possible will have a dramatic effect on the performance of your reports in the runtime environment. This is because each dataset is created by running a query against the database, and that is the most expensive part of running a report in terms of execution time.

This chapter describes the following topics related to datasets:

- Creating a dataset
- Managing datasets

Creating a dataset

The dataset is created automatically when a query is created and mapped to a report. You can also create a dataset independently, and then use it on a report later. This section describes the ways to create a dataset as follows:

Creating a dataset while creating report

When you create a data component in a page report tab or a library component, a dataset is created at the same time, either empty or containing some data fields, depending on whether data fields are added to the report.

Creating a dataset from the Report menu

To create a dataset by using the Report menu, follow the steps below:

1. In JReport Designer, click **Report** > **Dataset Management** on the menu bar.
2. In the **Dataset Management** dialog, click the **Add** button.
3. In the New Dataset dialog, specify the data source on which the dataset will be built.
4. Enter the name of the dataset in the New Dataset Name field.
5. Click **OK**, and an empty dataset will be created.

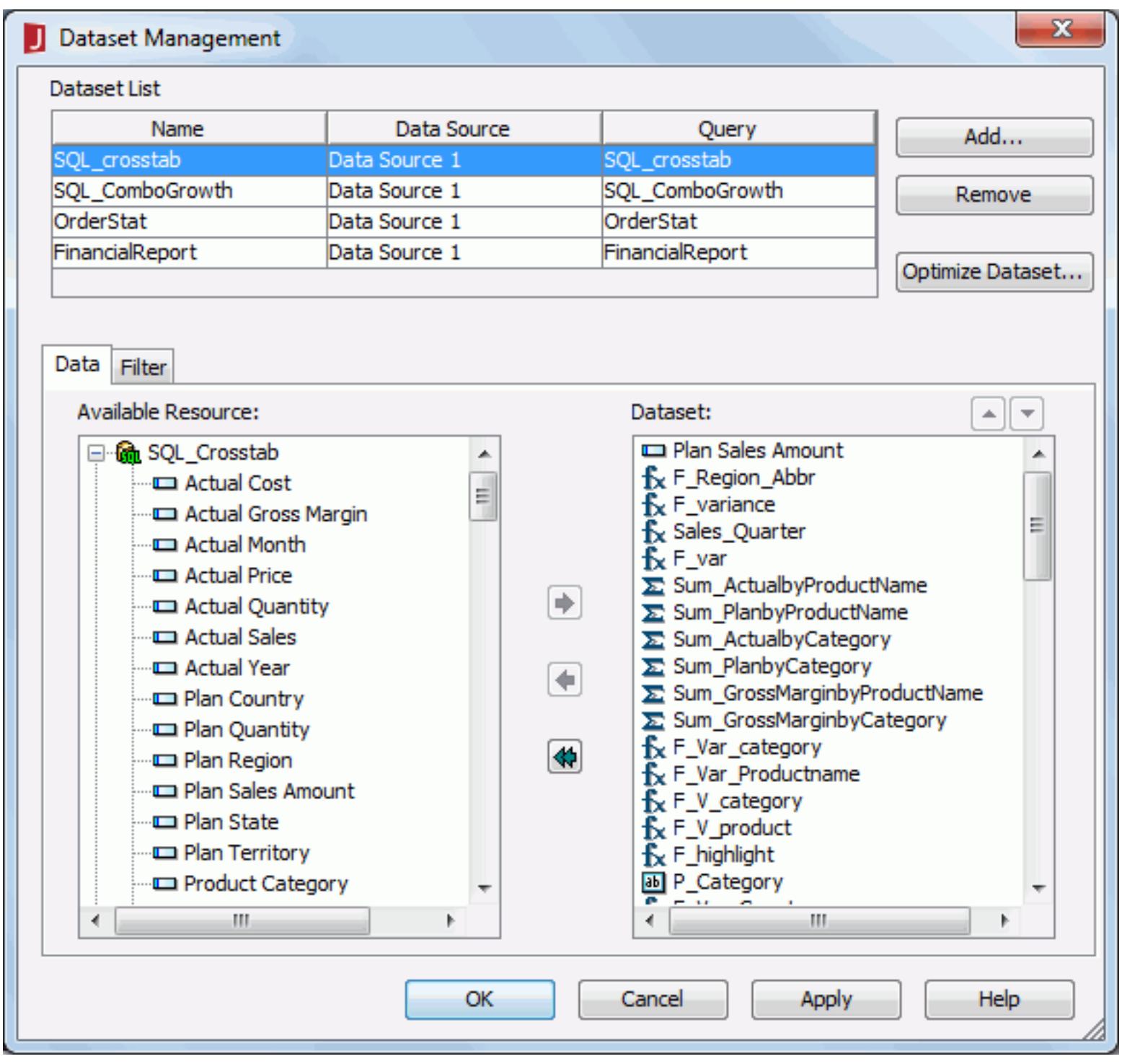
Creating a dataset via the Resource View panel

Another way to create a dataset is by using the Resource View panel. To do this:

1. In the Resource View panel, choose **<Choose Data From...>** from the dataset drop-down list.
2. In the New Dataset dialog, specify the source on which the dataset will be built.
3. Enter the name of the dataset in the New Dataset Name field.
4. Click **OK**, and an empty dataset will be created.

Managing datasets

After you have created some datasets in a report, you can manage them according to your requirements, for example, you can add data fields to a dataset or filter the dataset. To do this, click **Report** > **Dataset Management** on the menu bar, and the Dataset Management dialog appears, in which you can manage your datasets as required.



Adding/Removing data fields in a dataset

You can add additional data fields to a dataset as long as they come from the source on which the dataset is created, or remove some unnecessary data fields from a dataset. To do this:

1. In the Dataset List box of the Dataset Management dialog, select the dataset you want to edit.
2. In the Data tab of the dialog:
   - To add more data fields to the dataset, from the Available Resource box, select the fields and then click to add them to the dataset.
   - To remove an unnecessary data field from the dataset, select it in the Dataset box and click . Only the fields which are not used by any data component created on the dataset, either directly or indirectly, can be removed. To remove all the data fields in a dataset, click , however, when you click the button, only the unused data fields will be removed actually. When you enter the dialog the next time, you will find that the data fields used by data components created on the dataset will still display in the Dataset box.
3. When done, click **OK** to apply the changes and close the dialog box.

Filtering data fields in a dataset

1. In the Dataset List box of the Dataset Management dialog, select the dataset with the data fields you want to filter, then click the **Filter** tab.
2. Click the **Add Condition** button to add a condition line.
3. In the field text field, specify the field on which the filter will be based.
4. From the operator drop-down list, set the operator with which to compose the filter expression.
5. In the value text field, specify the value of how to filter the field.
6. To add another line of filter expression, click the **Add Condition** button and define the expression as required. Then from the logic drop-down list, specify the relationship between the two filter expressions.
7. Repeat the above steps to add more filter expressions if necessary.

   To make some conditions grouped, select them and click the **Group** button, then the selected conditions will be added in one group and work as one line of filter expression. Conditions and groups together can be further grouped. To take any condition or group in a group out, select it and click **Ungroup.**

   To adjust the priority of the conditions, select it and click the **Up** or **Down** button.

   To delete a condition line, select it and click the **Delete** button.
8. When done, click **OK** to apply the changes and close the dialog.

Tips:

- You can also add/remove the data fields in a dataset or filter a dataset while creating a data component with wizard. To do this, check the **Existing** radio button in the Data screen of the wizard, then select an existing dataset and click the **Modify** button. In the Dataset Editor, edit the dataset as required.
- A dataset can also be filtered using the Dataset Filter dialog, which is displayed by clicking the Dataset Filter button on the toolbar of the Resource View panel.
- You can use parameters to dynamically filter datasets as you do with queries. For details about parameter's usage on this, refer to Dynamically filtering queries.

Note: The following SQL type of data cannot be sorted: Db.SQL_BINARY, Db.SQL_BLOB, Db.SQL_CLOB, Db.SQL_LONGVARCHAR, Db.SQL_LONGVARBINARY, Db.SQL_VARBINARY and Db.SQL_OTHER.

Optimizing a dataset

You can enlarge or decrease the scope of retrieved data for a dataset, and therefore make a balanceable decision between better performance and special usage cases/demands. To do this:

1. In the Dataset List box of the Dataset Management dialog, select the dataset with the name you want to optimize.
2. Click the **Optimize Dataset** button.
3. In the Optimize Dataset dialog, choose a retrieved data scope for the dataset.
   - **Only Columns Used in Report**
     Only data columns used in the current report are retrieved at runtime. This way ensures the best performance since the least data is retrieved.
   - **All Columns in Dataset**
     All data columns defined in the dataset are retrieved at runtime.
   - **All Columns in Query**
     All data columns in the query that the dataset is based on are retrieved at runtime. The performance of this way is not as good as the other two.
4. Click **OK** to apply the settings and close the dialog.

Renaming a dataset

1. In the Dataset List box of the Dataset Management dialog, select the dataset with the name you want to modify.
2. Double-click the **Name** cell of the dataset.
3. Input a new name in the cell and press Enter to apply the changes.

   However, you are not recommended to rename a dataset that has been used by other data components randomly. Renaming a referenced dataset may cause data components based on the dataset not to work.

Removing a dataset

After you have created one or more datasets in a report, you can choose to remove them if they are no longer required for the report. To do this:

1. In the Dataset List box of the Dataset Management dialog, specify the dataset you want to remove.
2. Click the **Remove** button to remove the specified dataset.
3. When done, click **OK** to apply the changes and close the dialog.

   However, if you remove a dataset directly in this way, any data components created based on the dataset will not work. So, when you want to remove a dataset, it is recommended that you first remove the data components which reference this dataset.

Components

Components are the objects that you can place in a report. JReport provides a full set of components that allow you to present and control the report data and presentation in a wide variety of ways.

Component categories

Components in JReport can be classified as follows:

- **JReport components**: labels, text boxes, images, barcodes, ranks, tables, horizontal table, crosstabs, charts, tabulars, banded objects, subreports, UDOs, maps
- **Fields**: DBFields, formula fields, summary fields, parameter fields
- **Web controls**: form, text field, password, text area, checkbox, radio button, image button, button, drop-down list, list, expand/collapse group, parameter control, parameter form control, filter control, navigation control, slider
- **Multimedia objects**: applets, flash files, real media files, windows media files
- **Drawing objects**: lines, arcs, boxes, ovals, round boxes
- **Special fields:** Print Date, Print Time, Fetch Date, Fetch Time, Record Number and many more.

Components can be bound to a dataset

Some components can be bound with a dataset or inherit data from the dataset of their parents. These components are table, chart, crosstab, banded object and geographic map, and they are also referred to as data containers or data components.

You can set up link conditions between two data containers (except for geographic map) which have a parent-child relationship. For more information, see Setting up a data container link.

Component placement

Components can be placed within banded objects or some other components, as well as onto an empty area of a report. JReport Designer indicates whether or not the target location is valid for the component.

The following tables lists the report areas that are valid targets for the various components, listed on the left.

Report layout areas and the components they can contain

	Report Layout Area										
Component	**Banded Object BH panel**	**Banded Object BPH panel**	**Banded Object DT panel**	**Banded Object BPF panel**	**Banded Object BF panel**	**Banded Object GH panel**	**Banded Object GF panel**	**Text Box**	**Table Cell**	**Tabular Cell**	**Page Header / Footer Panel**
Banded object	Y	Y	Y	Y	Y	Y	Y	N	N	Y	Y
Chart	Y	N	N	N	Y	Y	Y	N	Y	Y	Y
Crosstab	Y	Y	Y	Y	Y	Y	Y	N	N	Y	Y
Table	Y	Y	Y	Y	Y	Y	Y	N	N	Y	Y
DBField	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y
Parameter field	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y
Summary field	Y	N	N	N	Y	Y	Y	N	N	N	N
Formula field	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y
Special field	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y
Label	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y
Tabular	Y	Y	Y	Y	Y	Y	Y	N	N	Y	Y
Image	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y
Text box	Y	Y	Y	Y	Y	Y	Y	N	N	Y	Y
Multimedia object	Y	Y	Y	Y	Y	Y	Y	Y	N	N	Y
Drawing object	Y	Y	Y	Y	Y	Y	Y	N	N	N	N
Subreport	Y	Y	Y	Y	Y	Y	Y	N	N	Y	Y
Map	Y	N	N	N	Y	Y	Y	N	N	Y	Y
UDO	Y	Y	Y	Y	Y	Y	Y	N	N	N	N
Form	Y	Y	Y	Y	Y	Y	Y	Y	N	Y	Y
Web control	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y
Barcode Rank	Barcode and rank components can be inserted in a report layout area that is valid for the component on which the barcode or rank is based. For example, a rank component that represents a DBField is allowed in the report layout areas in which a DBField component is allowed.										

This chapter describes each component in more detail:

- Labels
- DBFields
- Formula fields
- Summary fields
- Parameter fields
- Special fields
- Text boxes
- Tables
- Banded objects
- Crosstabs
- Charts
- Tabulars
- Subreports
- Images
- Maps
- Web controls
- Multimedia objects
- Barcodes
- Ranks
- Drawing objects
- UDOs

Labels

A label is an object that contains a string. It is typically a brief description used to identify a field or other value nearby. In JReport, the display type of a label can be changed, that is you can map the string to another value or image to be displayed instead.

The following tasks related to labels are described in this section:

- Inserting a label
- Editing and formatting a label
- Binding a link to a label
- Changing the display type of a label
- Applying web actions to a label

See an example: The SampleComponents catalog, included with JReport Designer, contains reports that have examples of how each component type could be used in a report. For the label component example, open `<install_root>\Demo\Reports\SampleComponents\ForLabel.cls`.

Inserting a label

Labels can be inserted in the report areas listed in Component placement.

You can use either of the following methods to insert a label:

- **Using menu/toolbar command**

  Click **Insert** > **Label** on the menu bar, or the **Insert Label** button on the Insert toolbar, then click the mouse button in the location where you want the label to be displayed.

- **By dragging and dropping**

  Drag **Label** from the Toolbox panel to the destination.

A label with the default text "Label" is inserted. Now, you can customize its appearance and behavior by setting properties in the Report Inspector, which can be saved as a CSS style for future use (for details, see Creating a CSS style).

Editing and formatting a label

The text of a label can be edited or formatted.

To edit a label:

- To edit the full text of a label, double-click the label to select the text, then type the replacement text.
- To edit a portion of a label's text, double-click the label, then click in the label and drag to select the portion, after that, type the replacement text.

You can also modify a label's text by using the Text property value in the Report Inspector.

To format a label:

You can format the label text with the Format toolbar, for example, you can change the font size, color, and alignment of the text.

To format a label, first select the label, and then perform any of the following tasks:

- To change the alignment of a label, click the required alignment button: **Justify**, **Left**, **Center** or **Right**. The text in the label will then be aligned according to your selection.
- To change the font face, click the **Font Face Box** Arial and then select the required font from the drop-down list. The fonts introduced with * are True Type Fonts. Refer to the chapter True Type Fonts for more information.
- To change the font size, click the **Font Size Box** 9 and then select the required size from the drop-down list (or enter the required value in the text box and then press **Enter**).
- To change the font style, click the corresponding font style buttons on the Format toolbar: **Bold** B, **Italic** I and **Underline** U.
- To change the background color of the label, click the **Background Color** button; to change the foreground color of the label text, click the **Foreground Color** button T.

Binding a link to a label

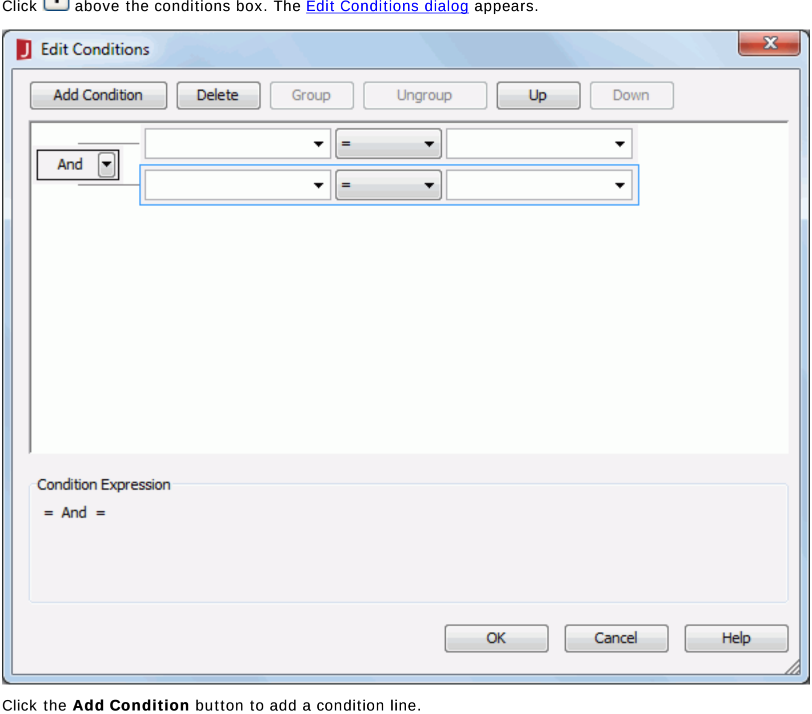
You can make the labels in a report linked with other reports, locations specified by URLs, e-mail addresses, or Blob data type fields. The link can either be a simple link or a conditional link. With conditional link, different targets can be loaded based on different conditions. However, conditional link is not supported on labels that are not bound with datasets.

To bind a simple link to a label:

1. Right-click the label and click **Link** on the shortcut menu to display the Insert Link dialog.
2. From the Link Type drop-down, select the target to which the label will be linked: Report, Master/Detail Report (available to page report only), URL, E-mail or Content.
3. Specify the options for the link target.
4. When done, click **OK** to create the link.

To bind a conditional link to a label:

1. Right-click the label and click **Link** on the shortcut menu to display the Insert Link dialog.
2. Check the **Conditional Link** checkbox.
3. Click [+] above the conditions box. The Edit Conditions dialog appears.



4. Click the **Add Condition** button to add a condition line.
5. From the field drop-down list, select the field on which the condition will be based.
6. Choose the operator with which to compose the condition expression from the operator drop-down list.
7. From the value drop-down list, specify the value of how to build the condition. You can also type in the value manually.
8. Click **Add Condition** to add more condition lines and define the relationship between the condition lines.
9. When done, click **OK** to save the condition. The newly added condition will then be displayed and highlighted in the conditions box in the Insert Link dialog.
10. From the Link Type drop-down, select the target to which the label will be linked under this condition: Report, Master/Detail Report (available to page report only), URL, E-mail or Content, then specify the options for the link target.
11. Repeat the above steps to add more conditions and define the target for each condition.

    To edit a condition, select the condition, click [edit], then edit it in the Edit Conditions dialog.

    To adjust the priority of the conditions, select a condition and click the [up] or [down] button.

    To remove a condition, select the condition and then click [delete].
12. Check the **Others** checkbox and define a target to which the label will be linked when none of the condition you have specified is met.
13. When done, click **OK** to create the conditional link.

For an illustrated example, see Example of binding a conditional link.

When a link is created, you can further edit it or remove it if you want.

- To edit an existing link on a label, right-click the label and select **Edit Link** from the shortcut menu. Then in the Edit Link dialog, redefine the link.
- To remove a link from a label, right-click the label and select **Remove Link** from the shortcut menu.

See also the Insert Link dialog for page report or web report/library component and the Edit Link dialog for page report or web report/library component for additional information about options in the dialogs.

Tip: By default, the link action has the highest click priority at runtime. You can change the priority by setting the report's Click Priority property.

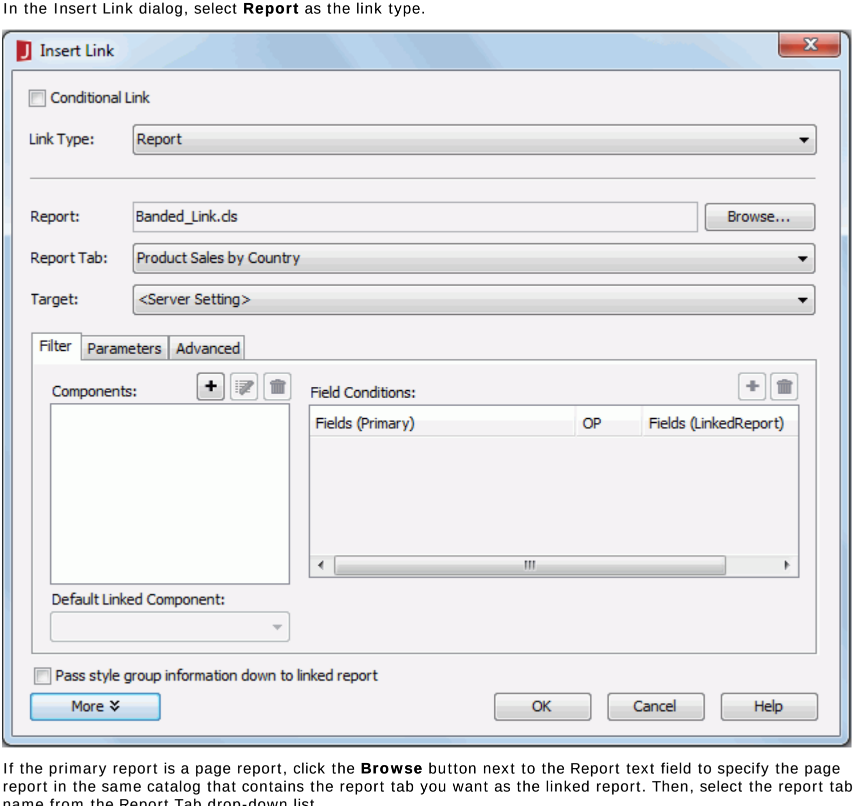
Linking to another report

In most cases, your reports are affiliated with one another, with each report emphasizing one or more aspects of a bigger picture. Relationships can be set up for users to browse from one report to another through the relationship "channels" created. You can achieve an inter-report relationship network by setting up the link conditions between two reports in the same catalog. Then, when you view the primary report at runtime, you can gain access to the linked report by clicking a trigger, which is an object defined in the primary report to load a linked report.

Usually, you can use a label, a field (DBField, summary, formula, parameter, or special field), an image, the category (X) axis, the series (Z) axis, the legend, or the data markers in a chart as a trigger object.

To make a label linked to another report:

1. In the Insert Link dialog, select **Report** as the link type.



2. If the primary report is a page report, click the **Browse** button next to the Report text field to specify the page report in the same catalog that contains the report tab you want as the linked report. Then, select the report tab name from the Report Tab drop-down list.

   If the primary report is a web report or a library component, click the **Browse** button to specify the web report in the same catalog that you want as the linked report.
3. From the Target drop-down list, select where the linked report will be loaded.

   The Target property works when the link is triggered in Page Report Studio or Web Report Studio. When the link is triggered in JDashboard, except for Same Frame, all the other targets are treated as New Window.
4. Click the **More** button to display more options for setting the link.
5. In the Filter tab, click the button [+] above the Components box. In the Choose Component dialog, add the components in the linked report you want to be interlinked with the primary report. If the required components use different datasets, you need to add them respectively, however, as long as the components are based on the same dataset, you can add them all at a time.
6. Specify the filter conditions based on which the link relationship between the primary report and the selected components in the linked report is set up.
   a. Select a component from the Components box.
   b. Click [+] above the Field Conditions box and a condition line is added.
   c. Select the desired field of the primary report from the drop-down list in the Fields(Primary) column.
   d. Choose an operator from the drop-down list in the OP column.
   e. Specify the field/formula of the linked report from the drop-down list in the Fields(LinkedReport) column. All fields in the linked report of the same value type as the selected primary report field are available in the drop-down list.
   f. If necessary, you can specify more conditions by clicking the button [+] and then specifying the primary report field, the operator, and the corresponding field in the linked report. Note that the relationship between these conditions is AND, which means that JReport will fetch linked report data which meets all of the conditions.
   g. Repeat the above steps to set link conditions between the primary report and other components in the linked report.
7. From the Default Linked Component drop-down list, select which component in the linked report will be linked with the primary report by default.
8. If the linked report contains one or more parameters, the Parameters tab will be enabled. In this tab, you can assign field of the primary report to parameter of the linked report. Then, when running the linked report from the link, the field value of the primary report will be assigned to the parameter automatically.
9. If you would like the values of the filter objects such as filter controls and sliders in the primary report to be passed to filter objects in the linked report, go to the **Advanced** tab to set up the relationship between the filter objects as follows:
   a. Click [+] to add a line.
   b. In the Primary Report Property/Object column, click [...] to select a filter object in the primary report in the Select Value dialog and click **OK**.
   c. In the Linked Report Property/Object column, click [...] to select a filter object in the linked report in the Select Value dialog and click **OK**. You need to make sure that values of the filter object in the primary report can be applied to the filter object in the linked report in the same line.
   d. Follow the above steps to set up the relationship between more filter objects.
10. Check **Pass style group information down to the linked report** to transfer style information from the primary report to the linked report, so that the style of the primary report will apply to the linked report.
11. Click **OK** to apply the settings.

Then, when you run the primary report at server runtime and click the label, you will find that the linked report is displayed according to the specified link conditions. You can also trigger the link in the HTML, PDF or Excel result if you check the Run Linked Report option in the corresponding format's export/option dialog.

Notes:

- A page report can only be linked with another page report, and a web report with another web report. For a library component, it can only be linked with a web report.
- When linking reports, you need to avoid link loops. For example, if you have linked report A to report B, then you cannot link report B back to report A again.
- The conditions specified in the Filter tab are used for filtering when the link is triggered at server runtime, that is, only the data that meet the conditions in the linked report will be displayed. However, when the link is triggered in HTML, PDF or Excel results, the conditions are used for setting up search criteria between the two linked reports, which means after you click the trigger object in the primary report, the pages containing the data that meet the conditions in the linked report will be displayed.

Linking to a detail report

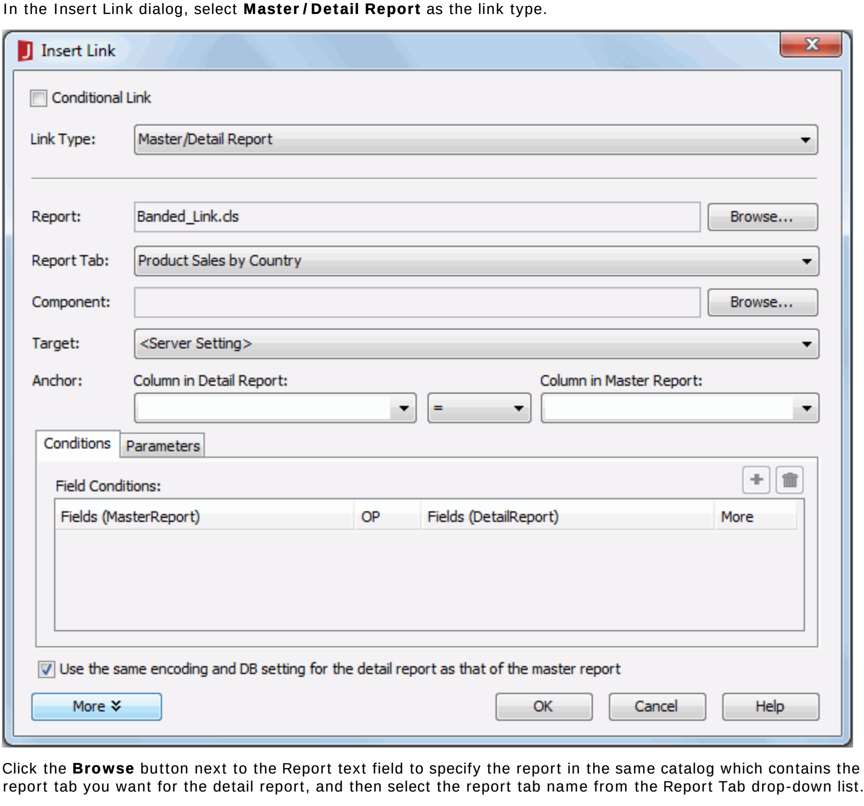
Reports in the same catalog can be joined together in order to create a master/detail report group. Usually, the master report holds comprehensive data, while the detail reports hold related detailed information. A detail report can also be the master report of another report. In this way, you can set up more pairs of master/detail reports, where many reports are joined together, eventually leading to the formation of a report chain. Then, when you view the master report at runtime, you can gain access to the detail report by clicking a trigger, which is an object defined in the master report to load a detail report. However, in a master/detail report group, crosstab reports cannot be used as the master.

Usually, you can use a label, a field (DBField, summary, formula, parameter, or special field), an image, the category (X) axis, the series (Z) axis, the legend, or the data markers in a chart as a trigger object.

Master/detail report is supported on page report only at present.

To make a label linked to a detail report:

1. In the Insert Link dialog, select **Master/Detail Report** as the link type.



2. Click the **Browse** button next to the Report text field to specify the report in the same catalog which contains the report tab you want for the detail report, and then select the report tab name from the Report Tab drop-down list.
3. Click the **Browse** button beside the Component field. In the Choose Component dialog, specify which component in the detail report you want to be interlinked with the master report with conditions. You can select more than one component as long as the components are based on the same dataset.
4. From the Target drop-down list, select where the detail report will be loaded.
5. Specify the join relationship between the master report and detail report in the Anchor section. The join can only have one condition. For example, the join can be: (Detail Report) Product ID = (Master Report) Product ID. The join condition should be as:

   `(Detail RPT) expression1 operator (Master RPT) expresion2`

   To make the join more meaningful, expression1 should be a column or record level formula in the GROUP section of a component in the detail report; and expression2 should be a column or record level formula in the DETAIL section of a component in the master report. This is because generally the master report is used to present comprehensive data, and the detail report to present more detailed information. To specify the join condition,
   a. Select a field/formula from the Column in Detail Report drop-down list. If this list does not offer the required field/formula, then you will need to specify the component with the data source that contains the specific field/formula. To do this, specify the component by clicking the Browse button besides the Component field.
   b. Choose the operator from the Operator drop-down list.
   c. Specify the field/formula of the master report from the Column in Master Report drop-down list. This field/formula should be of the same data type as the selected field/formula of the detail report.
6. Click the **More** button to display more options for setting the link.
7. In the Conditions tab, specify the filter conditions between the master report and the detail report.
   a. Click the add button [+] above the Field Conditions box and a condition line is added.
   b. Select the desired field of the master report from the Fields(MasterReport) drop-down list.
   c. Choose an operator from the drop-down list in the OP column.
   d. Specify the field/formula of the detail report from the drop-down list in the Fields(DetailReport) column. All fields in the detail report of the same value type as the selected primary report field are available in the drop-down list.
   e. If necessary, you can specify more conditions by clicking the button [+] and then specifying the master report field, the operator, and the corresponding field in the detail report. You can use the More column to specify the relationship between conditions to be AND or OR.
8. If there is at least one parameter applied in the filters of the queries that the detail report uses, the Parameters tab will be enabled. In this tab, you can assign field of the master report to parameter of the detail report. Then, when running the detail report from the link, the field value of the master report will be assigned to the parameter automatically.
9. If both reports use the same encoding and DB settings, you should check the option **Use the same encoding and DB settings for the detail report as that of the master report**. If this option is not checked, when the detail report is triggered at runtime, you will be prompted to specify the encoding and DB settings for the detail report.
10. Click **OK** to confirm.

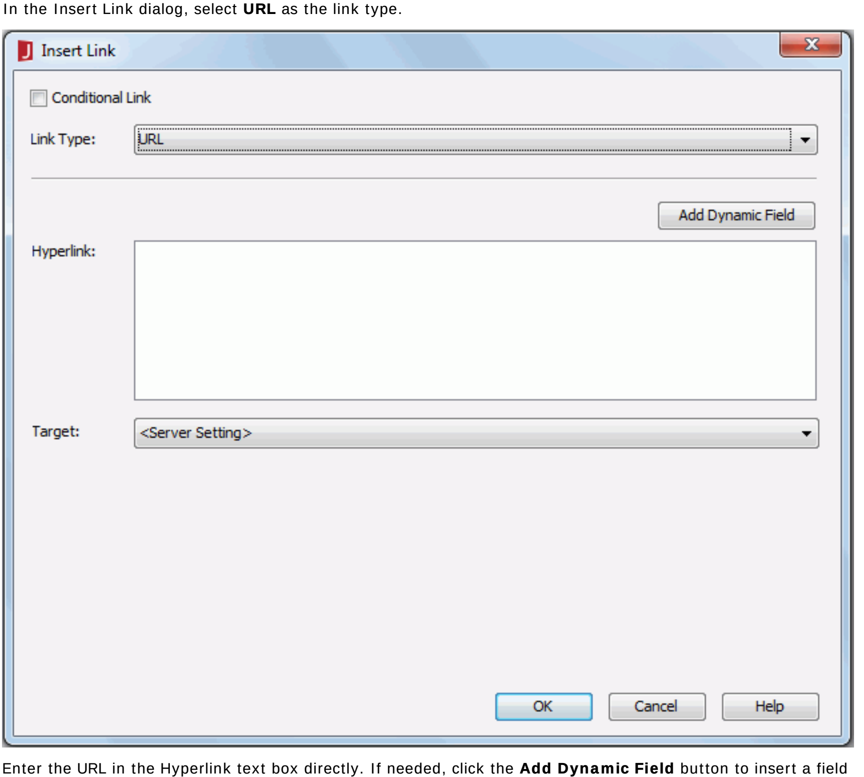
After a set of master/detail reports have been defined in JReport Designer and published to JReport Server, they can be viewed in Page Report Studio. The detail report supports all the page report operations, such as sort, filter, drill, and search.

Notes:

- The join and filter conditions work together. If you want to set a single condition, you can use join (anchor). If you want to set a group of conditions, you can use the filter conditions together with a join condition. The relationship between the join and filter conditions is "Join condition AND (filter conditions)".
- If you specify to load the detail report in a new browser window, the detail report will be opened independently when you click the trigger in Page Report Studio. That is to say, the join and filter conditions you have set between the master report and detail report will not be applied.

Linking to a URL

1. In the Insert Link dialog, select **URL** as the link type.



2. Enter the URL in the Hyperlink text box directly. If needed, click the **Add Dynamic Field** button to insert a field into the URL to compose a dynamic URL.

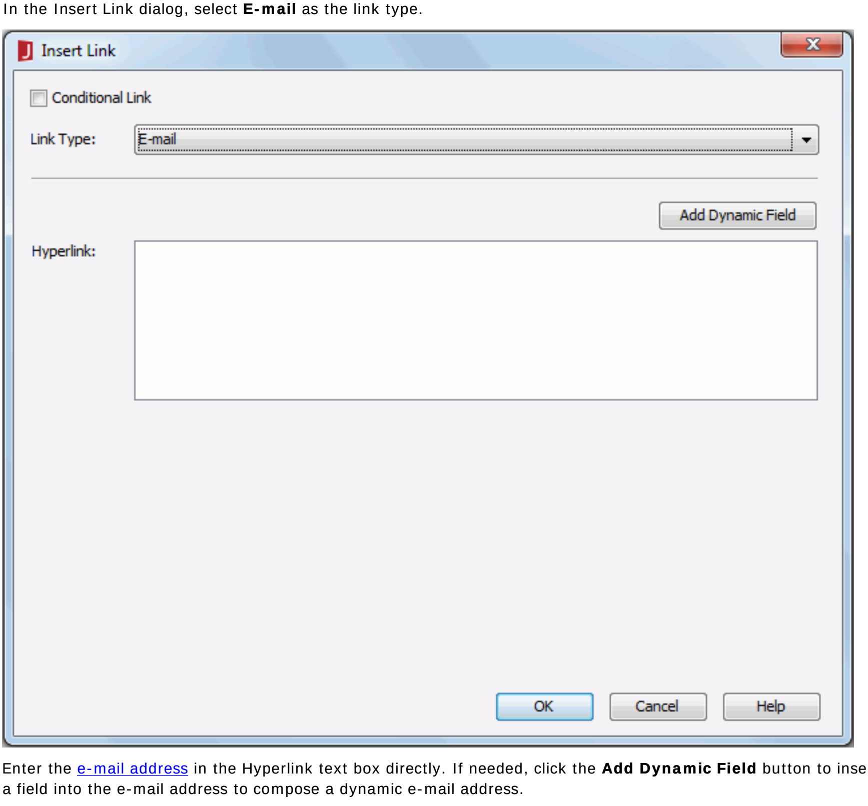
   For example, you can compose a URL as follows: type in **http://www.google.com/search?q=** into the Hyperlink text box, then click the **Add Dynamic Field** button to insert the field Country at the end of the URL. Then at runtime, you can click the label to open a specific URL that directs to a specific country.
3. Specify the frame from the Target drop-down list.

   The Target property works when the link is triggered in Page Report Studio or Web Report Studio. When the link is triggered in JDashboard, except for Same Frame, all the other targets are treated as New Window.
4. When you link a label in a web report table to a URL, you can click the **IE Navigate2 Flags** button to use the IE Navigate2 method and its flags to control the behavior of IE when loading the URL in Internet Explorer. Choose one or more flags as needed.
5. When done, click **OK** to set up the link.

Then, when you run the report at server runtime, or view it in HTML, PDF, or Excel format, you can click the label to open the URL.

Linking to an e-mail

1. In the Insert Link dialog, select **E-mail** as the link type.



2. Enter the e-mail address in the Hyperlink text box directly. If needed, click the **Add Dynamic Field** button to insert a field into the e-mail address to compose a dynamic e-mail address.
3. When done, click **OK** to set up the link.

Then, when you run the report at server runtime, or view it in HTML, PDF, or Excel format, you can click the label to open the e-mail with the information specified in the Hyperlink box. You can further customize the e-mail and send it.

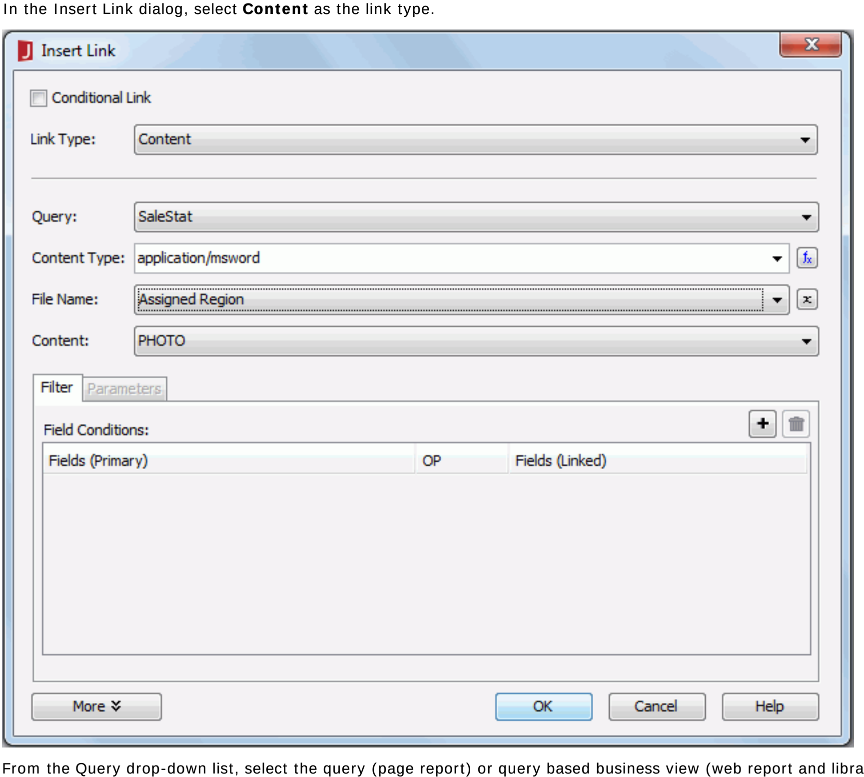
Linking to a Blob data type field

You can make an object linked with a Blob data type field. The Blob data type field will then be bound to the object and displayed as a hyperlink. You can download the Blob content by clicking this hyperlink.

In JReport, Blob could be image, Binary, Blob, Clob, LongBlob, LongClob, Varbinary, Longvarbinary, and so on.

To make a label linked to a Blob data type field:

1. In the Insert Link dialog, select **Content** as the link type.



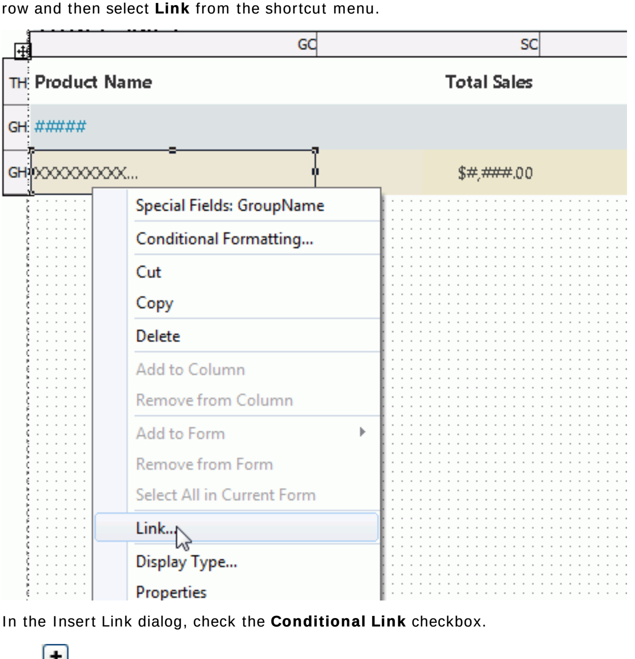
2. From the Query drop-down list, select the query (page report) or query based business view (web report and library component) which contains the required Blob data type field. Only queries/business views that come from the same data source connection as the query/business view used by the current report are available in the drop-down list.
3. From the Content Type drop-down list, specify the content type for the Blob type data. You can also click the button [fx] to bind the content type with a field or a formula in the selected query/business view.
4. In the File Name text field, specify a file name for the linked Blob type data. You can also click the button [fx] to specify a field or a formula in the selected query/business view to control the file name.
5. From the Content drop-down list, choose a Blob data type field in the selected query/business view.
6. Click the **More** button to display more link options.
7. In the Filter tab, specify the filter conditions between the query/business view used by the current report and the query/business view that contains the linked Blob content as follows:
   a. Click [+] above the Field Conditions box and a condition line is added.
   b. Select the desired field of the query/business view used by the current report from the Fields(Primary) drop-down list.
   c. Choose an operator from the drop-down list in the OP column.
   d. Specify the field/formula of the query/business view that contains the linked Blob content from the drop-down list in the Fields(Linked) column. All fields that are of the same data type as the selected field in the Fields (Primary) column are available in the drop-down list.
   e. If necessary, you can specify more conditions by clicking the button [+] and then specifying the fields and operators accordingly. Note that the relationship between these link conditions is AND, which means that JReport will fetch the linked Blob type data which meets all of the conditions.
8. If the query/business view that contains the linked Blob content uses parameters, the Parameters tab will be enabled. In this tab, you can assign fields of the query/business view used by the current report to the parameters. Then, when accessing the Blob type data from the link, the field values will be assigned to the parameters automatically.
9. Click **OK** to apply the settings.

Then, when you run the report at server runtime, or view it in HTML, PDF, or Excel format, you can click the label to download the Blob content according to the specified conditions.

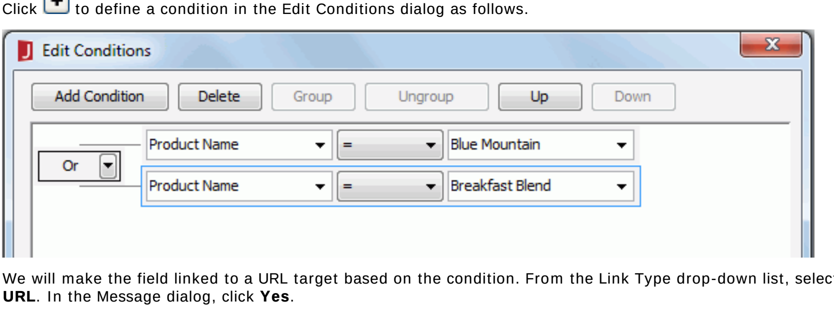
Example of binding a conditional link

This example shows the usage of conditional link and dynamic field.

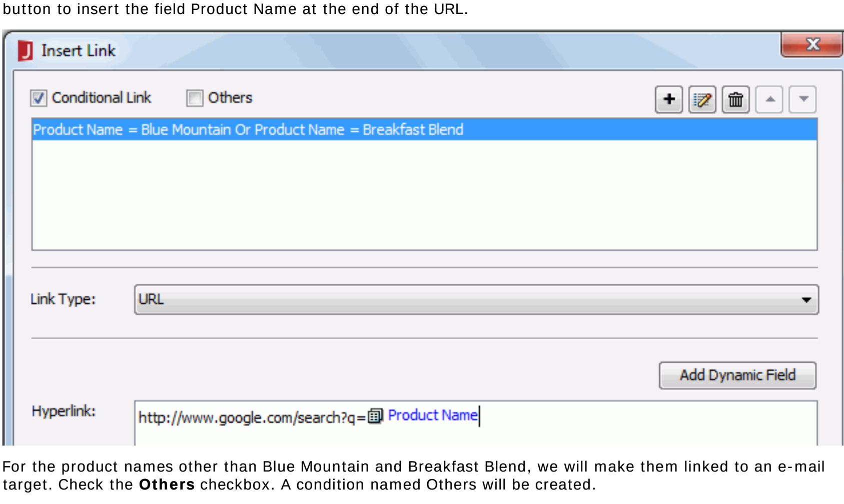
1. Open the report Link.wls located in `<install_root>\Demo\Reports\SampleReports\SampleReports.cat`.
2. We will define a link on the table of the report. Right-click on the **Product Name** field in the second group header row and then select **Link** from the shortcut menu.



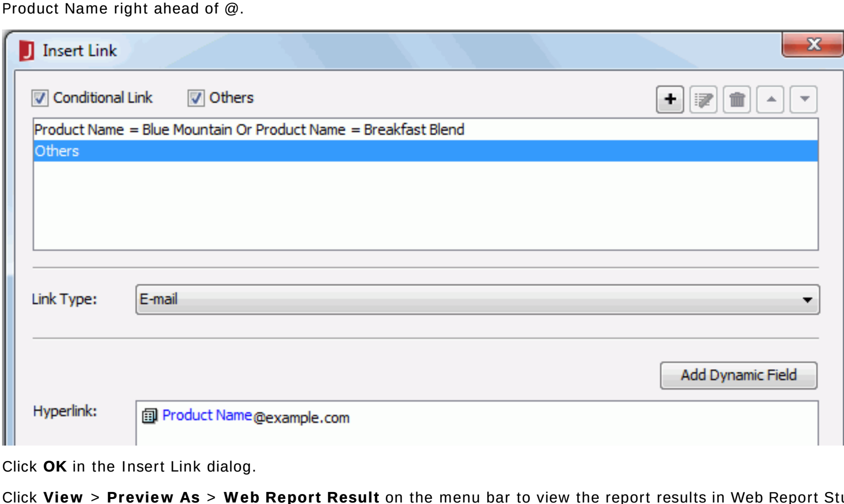
3. In the Insert Link dialog, check the **Conditional Link** checkbox.
4. Click [+] to define a condition in the Edit Conditions dialog as follows.



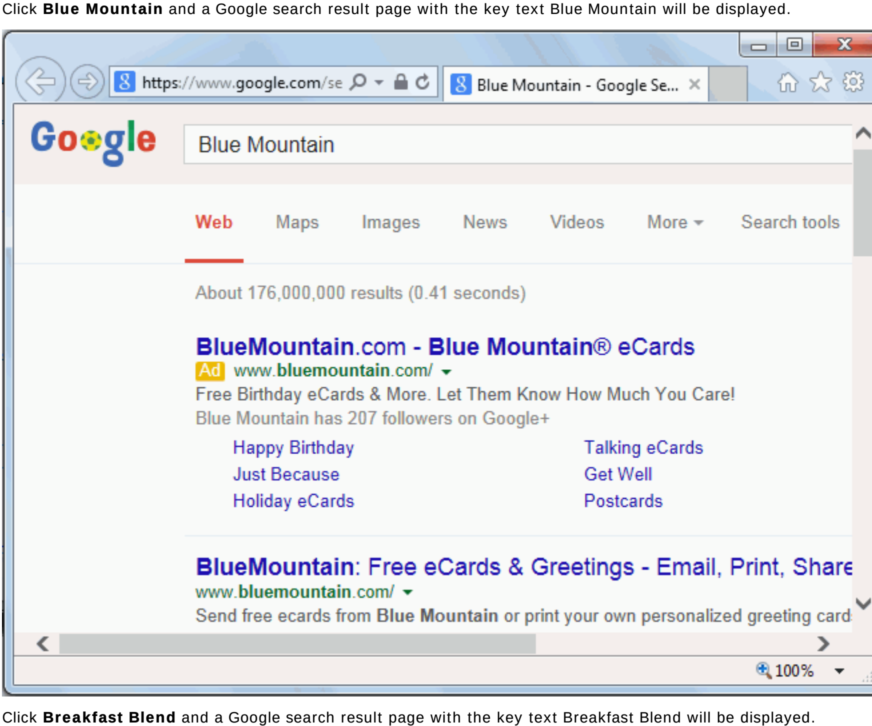
5. We will make the field linked to a URL target based on the condition. From the Link Type drop-down list, select **URL**. In the Message dialog, click **Yes**.
6. In the Hyperlink text box, type in **http://www.google.com/search?q=** and then click the **Add Dynamic Field** button to insert the field Product Name at the end of the URL.



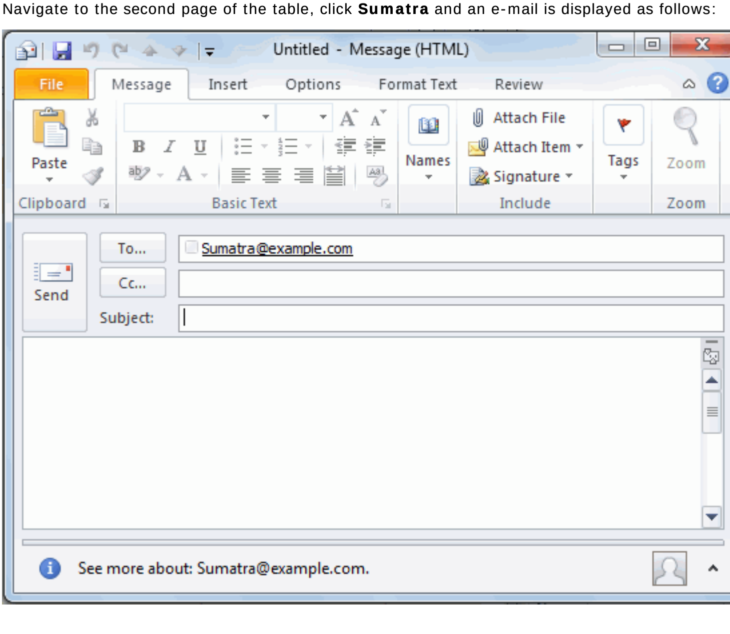
7. For the product names other than Blue Mountain and Breakfast Blend, we will make them linked to an e-mail target. Check the **Others** checkbox. A condition named Others will be created.
8. From the Link Type drop-down list, select **E-mail**. In the Message dialog, click **Yes**.
9. In the Hyperlink text box, type in **@example.com** and then click the **Add Dynamic Field** button to insert the field Product Name right ahead of @.



10. Click **OK** in the Insert Link dialog.
11. Click **View** > **Preview As** > **Web Report Result** on the menu bar to view the report results in Web Report Studio. We can click the product names in the table to trigger the link.
12. Click **Blue Mountain** and a Google search result page with the key text Blue Mountain will be displayed.



13. Click **Breakfast Blend** and a Google search result page with the key text Breakfast Blend will be displayed.
14. Navigate to the second page of the table, click **Sumatra** and an e-mail is displayed as follows:



Changing the display type of a label

The display type of a label can be changed, that is to say, you can map the label to another value or image to be displayed instead.

JReport provides 13 kinds of display types: text, image, barcode, rank, text field, hidden field, text area, checkbox, radio button, image button, button, list, and drop-down list.

To change the display type of a label:

1. Right-click the label and select **Display Type** to open the Display Type dialog.
2. In the Type box, select the required type.
3. Set the options for the selected type in the Web Options panel.
4. Click the **OK** button to close the dialog.

A label can be displayed as one of the following general types:

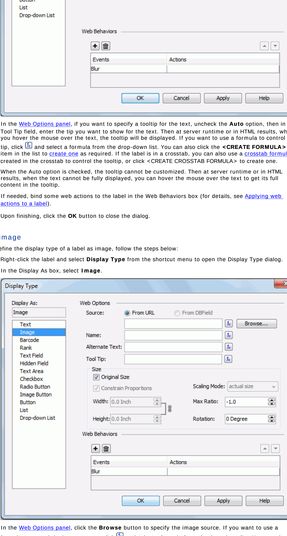
- As text
- As image
- As barcode
- As rank

The display type of a label can also be changed to a web control, including text field, hidden field, text area, checkbox, radio button, image button, and button. A label cannot be displayed as list or drop-down list. For details about how to make settings of these display types, see Using basic web controls.

As text

To define the display type of a label as text, follow the steps below:

1. Right-click the label and select **Display Type** from the shortcut menu to open the Display Type dialog.
2. In the Display As box, select **Text**.



3. In the Web Options panel, if you want to specify a tooltip for the text, uncheck the **Auto** option, then in the Tool Tip field, enter the tip you want to show for the text. Then at server runtime or in HTML results, when you hover the mouse over the text, the tooltip will be displayed. If you want to use a formula to control the tip, click and select a formula from the drop-down list. You can also click the **<CREATE FORMULA>** item in the list to create one as required. If the label is in a crosstab, you can also use a crosstab formula created in the crosstab to control the tooltip, or click <CREATE CROSSTAB FORMULA> to create one.

   When the Auto option is checked, the tooltip cannot be customized. Then at server runtime or in HTML results, when the text cannot be fully displayed, you can hover the mouse over the text to get its full content in the tooltip.
4. If needed, bind some web actions to the label in the Web Behaviors box (for details, see Applying web actions to a label).
5. Upon finishing, click the **OK** button to close the dialog.

As image

To define the display type of a label as image, follow the steps below:

1. Right-click the label and select **Display Type** from the shortcut menu to open the Display Type dialog.
2. In the Display As box, select **Image**.

3. In the Web Options panel, click the **Browse** button to specify the image source. If you want to use a formula to control the image source, click and select a formula from the drop-down list. You can also click the **<CREATE FORMULA>** item in the list to create one as required. If the label is in a crosstab, you can also use a crosstab formula created in the crosstab to control the image source, or click <CREATE CROSSTAB FORMULA> to create one.

   For a DBField, formula, or a summary, you can also check the From DBField radio button to make the value of the DBField/formula/summary the image source. If you choose this option, the Decode Type drop-down list is enabled, from which you can specify the type for decoding the image.
4. Type the name of the image in the Name field, or use a formula to control the name.
5. Type a string in the Alternate Text field to serve as content when the image cannot be displayed, or use a formula to control the alternate text.
6. In the Tool Tip field, enter the tip you want to show for the image or use a formula to control the tooltip. Then at server runtime or in HTML results, when you hover the mouse over the image, the tooltip will be displayed.
7. Specify the scaling mode of the image from the Scaling Mode drop-down list, and the maximum scaling ratio in the Max Ratio field. By default the scaling ratio of the image is not limited.

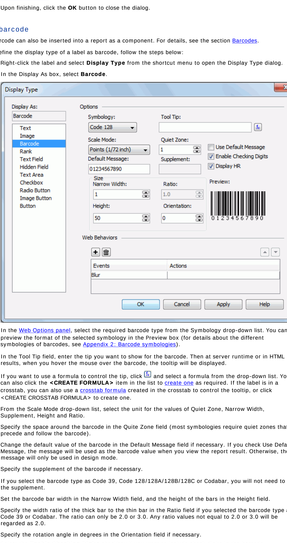
   The following are the available scaling modes:
   - **actual size** - If selected, the image will be shown in its actual size.
   - **fit image** - If selected, the image will be scaled to be wholly shown.
   - **fit width** - If selected, the image will be scaled to fit the width of the image viewer.
   - **fit height** - If selected, the image will be scaled to fit the height of the image viewer.
   - **customize** - If selected, the image will be scaled according to the width and height that you specify in the Width and Height box.
8. Specify the size of the image.
   - If you choose to use the original size of the image, check the **Original Size** option.
   - If you want to use customized image size, uncheck **Original Size**, and then set the display width and height of the image in the Width and Height fields. Check the **Constrain Proportion** option if you want to constrain aspect ratio of the image when setting the width and height.
9. Specify the rotation degree of the image in the Rotation field.
   - 0 - No rotation.
   - Positive value - Rotates the image clockwise.
   - Negative value - Rotates the image anticlockwise.
10. If needed, bind some web actions to the label in the Web Behaviors box (for details, see Applying web actions to a label).
11. Upon finishing, click the **OK** button to close the dialog.

As barcode

A barcode can also be inserted into a report as a component. For details, see the section Barcodes.

To define the display type of a label as barcode, follow the steps below:

1. Right-click the label and select **Display Type** from the shortcut menu to open the Display Type dialog.
2. In the Display As box, select **Barcode**.



3. In the Web Options panel, select the required barcode type from the Symbology drop-down list. You can preview the format of the selected symbology in the Preview box (for details about the different symbologies of barcodes, see Appendix 2: Barcode symbologies).
4. In the Tool Tip field, enter the tip you want to show for the barcode. Then at server runtime or in HTML results, when you hover the mouse over the barcode, the tooltip will be displayed.

   If you want to use a formula to control the tip, click and select a formula from the drop-down list. You can also click the **<CREATE FORMULA>** item in the list to create one as required. If the label is in a crosstab, you can also use a crosstab formula created in the crosstab to control the tooltip, or click <CREATE CROSSTAB FORMULA> to create one.
5. From the Scale Mode drop-down list, select the unit for the values of Quiet Zone, Narrow Width, Supplement, Height and Ratio.
6. Specify the space around the barcode in the Quite Zone field (most symbologies require quiet zones that precede and follow the barcode).
7. Change the default value of the barcode in the Default Message field if necessary. If you check Use Default Message, the message will be used as the barcode value when you view the report result. Otherwise, the message will only be used in design mode.
8. Specify the supplement of the barcode if necessary.

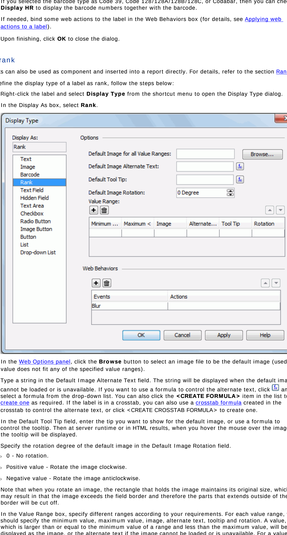
   If you select the barcode type as Code 39, Code 128/128A/128B/128C or Codabar, you will not need to set the supplement.
9. Set the barcode bar width in the Narrow Width field, and the height of the bars in the Height field.
10. Specify the width ratio of the thick bar to the thin bar in the Ratio field if you selected the barcode type as Code 39 or Codabar. The ratio can only be 2.0 or 3.0. Any ratio values not equal to 2.0 or 3.0 will be regarded as 2.0.
11. Specify the rotation angle in degrees in the Orientation field if necessary.
12. Select the **Enable Checking Digits** checkbox if you want to include check digits in the barcode.
13. If you selected the barcode type as Code 39, Code 128/128A/128B/128C, or Codabar, then you can check **Display HR** to display the barcode numbers together with the barcode.
14. If needed, bind some web actions to the label in the Web Behaviors box (for details, see Applying web actions to a label).
15. Upon finishing, click **OK** to close the dialog.

As rank

Ranks can also be used as component and inserted into a report directly. For details, refer to the section Ranks.

To define the display type of a label as rank, follow the steps below:

1. Right-click the label and select **Display Type** from the shortcut menu to open the Display Type dialog.
2. In the Display As box, select **Rank**.



3. In the Web Options panel, click the **Browse** button to select an image file to be the default image (used if a value does not fit any of the specified value ranges).
4. Type a string in the Default Image Alternate Text field. The string will be displayed when the default image cannot be loaded or is unavailable. If you want to use a formula to control the alternate text, click and select a formula from the drop-down list. You can also click the **<CREATE FORMULA>** item in the list to create one as required. If the label is in a crosstab, you can also use a crosstab formula created in the crosstab to control the alternate text, or click <CREATE CROSSTAB FORMULA> to create one.
5. In the Default Tool Tip field, enter the tip you want to show for the default image, or use a formula to control the tooltip. Then at server runtime or in HTML results, when you hover the mouse over the image, the tooltip will be displayed.
6. Specify the rotation degree of the default image in the Default Image Rotation field.
   - 0 - No rotation.
   - Positive value - Rotate the image clockwise.
   - Negative value - Rotate the image anticlockwise.

   Note that when you rotate an image, the rectangle that holds the image maintains its original size, which may result in that the image exceeds the field border and therefore the parts that extends outside of the border will be cut off.
7. In the Value Range box, specify different ranges according to your requirements. For each value range, you should specify the minimum value, maximum value, image, alternate text, tooltip and rotation. A value, which is larger than or equal to the minimum value of a range and less than the maximum value, will be displayed as the image, or the alternate text if the image cannot be loaded or is unavailable. For a value belonging to two or more overlapping ranges, the highest range in the Value Range box will apply.

   Click or or to add or delete a range, and or or to adjust the order of the ranges.
8. If needed, bind some web actions to the label in the Web Behaviors box (for details, see Applying web actions to a label).
9. Upon finishing, click the **OK** button to close this dialog.

Notes:

- If you have specified a tooltip, the tooltip is displayed differently on different browsers:
  - On Internet Explorer, the tip text is automatically wrapped if its length exceeds the maximal tip width the browser allows. In addition, for Internet Explorer, you can manually make some text displayed in a new line by adding &#10 or &#13 ahead of the text when editing it in the Tool Tip field.
  - On Firefox, the tip text is displayed in one line with the maximal tip width the browser allows, and the text that cannot be displayed within the width will be cut off and replaced by ellipsis.
- When you rotate an image, the rectangle that holds the image maintains its original size, which may result in that the image exceeds the field border and therefore the parts that extends outside of the border will be cut off.

Applying web actions to a label

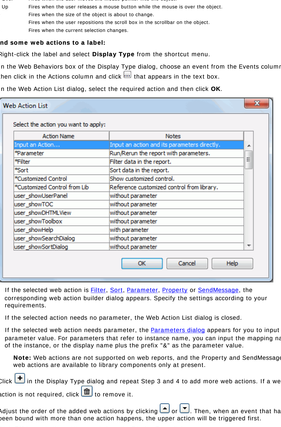
A label can be bound with some web actions. This allows you to customize a label to make it respond to some events, and execute corresponding actions, such as sorting and filtering. For example, you can insert a label in JReport Designer and associate an action with the label's onclick event. Then, when you view the report at runtime and click the label, the defined operation will be executed.

The following table lists the events that can be used to trigger web actions:

Events	Description
Blur	Fires when the object loses the input focus.
Data Change	Fires when the contents of the object or selection have changed.
Click	Fires when the user clicks the left mouse button on the object.
onContextMenu	Fires when the user clicks the right mouse button on the object, opening the shortcut menu.
Double_Click	Fires when the user double-clicks the object.
Focus	Fires when the object receives focus.
Key Down	Fires when the user presses a key.
Key Press	Fires when the user presses an alphanumeric key.
Key Up	Fires when the user releases a key.
Mouse Down	Fires when the user clicks the object with either mouse button.
Mouse Move	Fires when the user moves the mouse over the object.
Mouse Out	Fires when the user moves the mouse pointer outside the boundaries of the object.
Mouse Over	Fires when the user moves the mouse pointer into the object.
Mouse Up	Fires when the user releases a mouse button while the mouse is over the object.
Resize	Fires when the size of the object is about to change.
Scroll	Fires when the user repositions the scroll box in the scrollbar on the object.
Select	Fires when the current selection changes.

To bind some web actions to a label:

1. Right-click the label and select **Display Type** from the shortcut menu.
2. In the Web Behaviors box of the Display Type dialog, choose an event from the Events column, then click in the Actions column and click [...] that appears in the text box.
3. In the Web Action List dialog, select the required action and then click **OK**.



- If the selected web action is Filter, Sort, Parameter, Property or SendMessage, the corresponding web action builder dialog appears. Specify the settings according to your requirements.
- If the selected action needs no parameter, the Web Action List dialog is closed.
- If the selected web action needs parameter, the Parameters dialog appears for you to input the parameter value. For parameters that refer to instance name, you can input the mapping name of the instance, or the display name plus the prefix "&" as the parameter value.

  Note: Web actions are not supported on web reports, and the Property and SendMessage web actions are available to library components only at present.

4. Click [+] in the Display Type dialog and repeat Step 3 and 4 to add more web actions. If a web action is not required, click [trash] to remove it.
5. Adjust the order of the added web actions by clicking [^] or [v]. Then, when an event that has been bound with more than one action happens, the upper action will be triggered first.
6. Click **OK** to apply the settings and close the Display Type dialog.

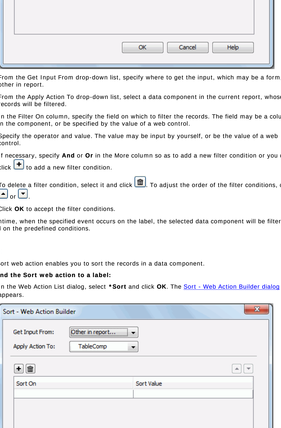
Then, at runtime, when any of the specified events occurs on the label, the web action defined on the event will be triggered.

Filter

The Filter web action enables you to filter the records in a data component.

To bind the Filter web action to a label:

1. In the Web Action List dialog, select ***Filter** and click **OK**. The Filter - Web Action Builder dialog appears.



2. From the Get Input From drop-down list, specify where to get the input, which may be a form, or other in report.
3. From the Apply Action To drop-down list, select a data component in the current report, whose records will be filtered.
4. In the Filter On column, specify the field on which to filter the records. The field may be a column in the component, or be specified by the value of a web control.
5. Specify the operator and value. The value may be input by yourself, or be the value of a web control.
6. If necessary, specify **And** or **Or** in the More column so as to add a new filter condition or you can click [+] to add a new filter condition.

   To delete a filter condition, select it and click [trash]. To adjust the order of the filter conditions, click [^] or [v].
7. Click **OK** to accept the filter conditions.

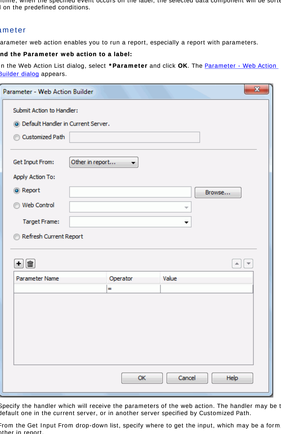
At runtime, when the specified event occurs on the label, the selected data component will be filtered based on the predefined conditions.

Sort

The Sort web action enables you to sort the records in a data component.

To bind the Sort web action to a label:

1. In the Web Action List dialog, select ***Sort** and click **OK**. The Sort - Web Action Builder dialog appears.



2. From the Get Input From drop-down list, specify where to get the input, which may be a form, or other in report.
3. From the Apply Action To drop-down list, select a data component in the current report, whose records will be sorted.
4. In the Sort On column, specify the field on which to sort the records. The field may be a column in the component, or be specified by the value of a web control.
5. Specify the sort value. The value may be Ascending or Descending, or be the value of a web control.
6. If necessary, click [+] to add a new sort condition.

   To delete a sort condition, select it and click [trash]. To adjust the order of the sort conditions, click [^] or [v].
7. Click **OK** to accept the sort conditions.

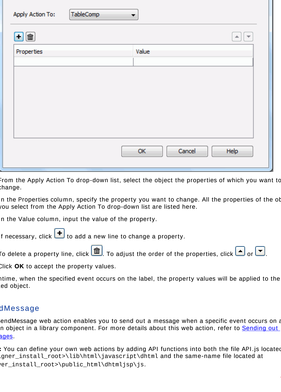
At runtime, when the specified event occurs on the label, the selected data component will be sorted based on the predefined conditions.

Parameter

The Parameter web action enables you to run a report, especially a report with parameters.

To bind the Parameter web action to a label:

1. In the Web Action List dialog, select ***Parameter** and click **OK**. The Parameter - Web Action Builder dialog appears.



2. Specify the handler which will receive the parameters of the web action. The handler may be the default one in the current server, or in another server specified by Customized Path.
3. From the Get Input From drop-down list, specify where to get the input, which may be a form, or other in report.
4. Specifies whether to apply the action to run a report or refresh the current. If you choose to run a report, click the **Browse** button to specify the report or use a web control to retrieve the report name at runtime, then select the window or frame in which the report will be opened from the Target Frame drop-down list.
5. The parameters used by the specified report are automatically listed in the parameters box. Specify the value for each parameter.
6. Click **OK** to accept the parameter values.

At runtime, when the specified event occurs on the label, the selected report will run using the specified parameter values.

Property

The Property web action enables you to make the properties of an object changed at runtime, however, it is available to the following objects in library components only: fields and labels in tables/crosstabs, some chart elements such as legend, chart axis and so on, markers and areas in geographic maps, as well as labels, parameter controls and the Submit buttons of parameter form controls which are not in the library components' configuration panels.

To bind the Property web action to a label:

1. In the Web Action List dialog, select ***Property** and click **OK**. The Change Property - Web Action Builder dialog appears.

2. From the Apply Action To drop-down list, select the object the properties of which you want to change.
3. In the Properties column, specify the property you want to change. All the properties of the object you select from the Apply Action To drop-down list are listed here.
4. In the Value column, input the value of the property.
5. If necessary, click [+] to add a new line to change a property.

   To delete a property line, click [trash]. To adjust the order of the properties, click [^] or [v].
6. Click **OK** to accept the property values.

At runtime, when the specified event occurs on the label, the property values will be applied to the selected object.

SendMessage

The SendMessage web action enables you to send out a message when a specific event occurs on a certain object in a library component. For more details about this web action, refer to Sending out messages.

Note: You can define your own web actions by adding API functions into both the file API.js located at <designer_install_root>\lib\html\javascript\dhtml and the same-name file located at <server_install_root>\public_html\dhtmljsp\js.

DBFields

DBFields, or database fields, are fields directly from databases, user data sources, hierarchical data sources, or other data sources.

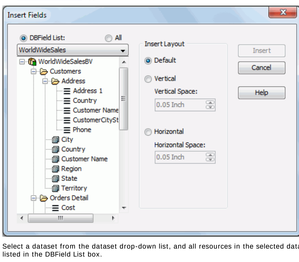
Inserting a DBField in a report

DBFields can be inserted in the report areas listed in Component placement.

To insert a DBField into a report, you can use either of the following two ways:

- **Using dialog**
  1. Position the mouse pointer at the destination where you want to insert the DBField and do one of the following:
     - Click **Insert** > **DBField** on the menu bar.
     - Click the **Insert DBField** button on the Insert toolbar.

     The Insert Fields dialog appears.



  2. Select a dataset from the dataset drop-down list, and all resources in the selected dataset will then be listed in the DBField List box.
  3. Select the required DBField and specify its layout in the Insert Layout box.
  4. Click the **Insert** button to insert the DBField into the destination.
- **By dragging and dropping**

  In the Resource View panel, select a dataset from the dataset drop-down list, then select the required DBField and drag it to the destination.

When a DBField has been inserted into a report, you can customize its appearance and behavior by setting properties in the Report Inspector, which can be saved as a CSS style for future use (for details, see Creating a CSS style).

Notes:

- To insert more than one DBField with the Insert Field dialog, you can select them one by one with the Ctrl key, or you can make a continuous selection with the Shift key.
- You cannot use the Insert Fields dialog to insert a DBField into a crosstab.
- If you insert a DBField into the detail panel of a banded object or table, the name of the DBField will automatically be inserted as a label into the corresponding header panel. Otherwise the field and its name label will be placed in the same panel. If you do not want to insert the field name automatically, you can uncheck Insert field name label with field in the Options dialog (**File** > **Options** > **Component** > **Insert field name label with field**).

Adding a summary for a DBField

To add a summary for a DBField:

1. Right-click the DBField and select **Summary Function** from the shortcut menu.
2. In the Insert Summary dialog, specify the summary settings as required.
3. When done, click **Insert**.

For more information, see Inserting a summary in a report.

Changing the display type of a DBField

The display type of a DBField can be changed, that is to say, you can map the DBField to another value or image to be displayed instead.

To change the display type of a DBField:

1. Right-click the DBField and select **Display Type** from the shortcut menu.
2. In the Display Type dialog, specify the required display type in the Type box, and set the options for the selected type in the Web Options panel.
3. Upon finishing, click the **OK** button to close the dialog.

For details about display types, see Changing the display type of a label.

Applying web actions to a DBField

A DBField can be bound with some web actions. This allows you to customize a DBField to make it respond to some events, and execute corresponding actions, such as sorting and filtering.

To apply web actions to a DBField:

1. Right-click the DBField and select **Display Type** from the shortcut menu.
2. In the Web Behaviors box of the Display Type dialog, choose an event from the Events column, then click in the Actions column and click that appears in the text box.
3. In the Web Action List dialog, select a web action and define the action as required.
4. Click in the Display Type dialog to add more web behavior lines and specify the events and actions accordingly. If a web behavior is not required, click to remove it.
5. Adjust the order of the added web actions by clicking or . Then, when an event that has been bound with more than one action happens, the upper action will be triggered first.
6. When done , click **OK** to accept the settings.

For more information about web actions, see Applying web actions to a label.

Binding a link to a DBField

You can make a DBField linked with another report, a location specified by a URL, an e-mail address, or a Blob data type field. The link can either be a simple link or a conditional link. With conditional link, different targets can be loaded based on different conditions.

To bind a link to a DBField:

1. Right-click the DBField and click **Link** on the shortcut menu.
2. In the Insert Link dialog, specify whether it is a conditional link or a simple link, then select the link target and set the options for the link target as required.
3. When done, click **OK**.

For details about different link targets, see Binding a link to a label.

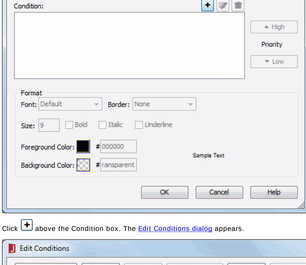
Adding conditional formats to a DBField

You can add some conditional formats to a DBField, then when a specified condition is fulfilled, the format bound with the condition will be applied to the field values automatically for distinguishing.

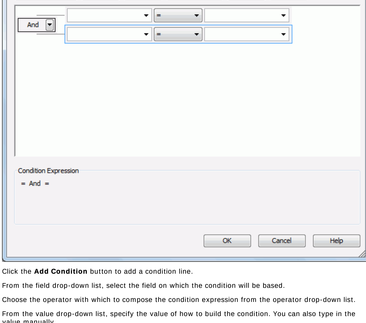
To add conditional formats to a DBField:

1. Select the DBField and then do one of the following:
   - Click **Format** > **Conditional Formatting** on the menu bar.
   - Click the **Conditional Formatting** button on the Format toolbar.
   - Right-click the DBField and select **Conditional Formatting** from the shortcut menu.

   The Conditional Formatting dialog appears.



2. Click above the Condition box. The Edit Conditions dialog appears.



3. Click the **Add Condition** button to add a condition line.
4. From the field drop-down list, select the field on which the condition will be based.
5. Choose the operator with which to compose the condition expression from the operator drop-down list.
6. From the value drop-down list, specify the value of how to build the condition. You can also type in the value manually.
7. Click **Add Condition** to add more condition lines and define the relationship between the condition lines.
8. When done, click **OK** to save the condition. The newly added condition will then be displayed and highlighted in the Condition box in the Conditional Formatting dialog.
9. In the Format box, set the format which will be applied to values of the DBField when the specified condition is fulfilled, such as the font face, font size, font color and so on.
10. Repeat the above steps to add more conditions and define the format for each condition.

    To edit a condition, select the condition from the Condition box, click then edit the condition in the Edit Conditions dialog.

    To adjust the priority of the conditions, select a condition and click the **High** or **Low** button.

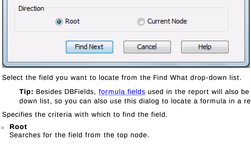
    To remove a condition and the corresponding format, select the condition from the Condition box and then click .
11. When done, click **OK** to save the settings.

Searching for a specific DBField in a report

When a report has many DBFields, you may find it difficult to locate where a certain field is. To easily find out the field you have lost track of in your report, you can make use of the Find DBField feature.

To find a DBField in a report:

1. Click **Edit** > **Find DBField** on the menu bar, or the **Find DBField** button on the Standard toolbar. The Find dialog appears.



2. Select the field you want to locate from the Find What drop-down list.

   Tip: Besides DBFields, formula fields used in the report will also be listed in the Find What drop-down list, so you can also use this dialog to locate a formula in a report.
3. Specifies the criteria with which to find the field.
   - **Root**
     Searches for the field from the top node.
   - **Current Node**
     Searches for the field from the current node.
4. Click the **Find Next** button and the specified field will be selected in the report.

See an example: The SampleComponents catalog, included with JReport Designer, contains reports that have examples of how each component type could be used in a report. For the DBField component example, open `<install_root>\Demo\Reports\SampleComponents\ForDBField.cls`.

Formula fields

Formula fields are calculated from DBFields, other formulas, summaries, and parameters, so they can present information that DBFields cannot do.

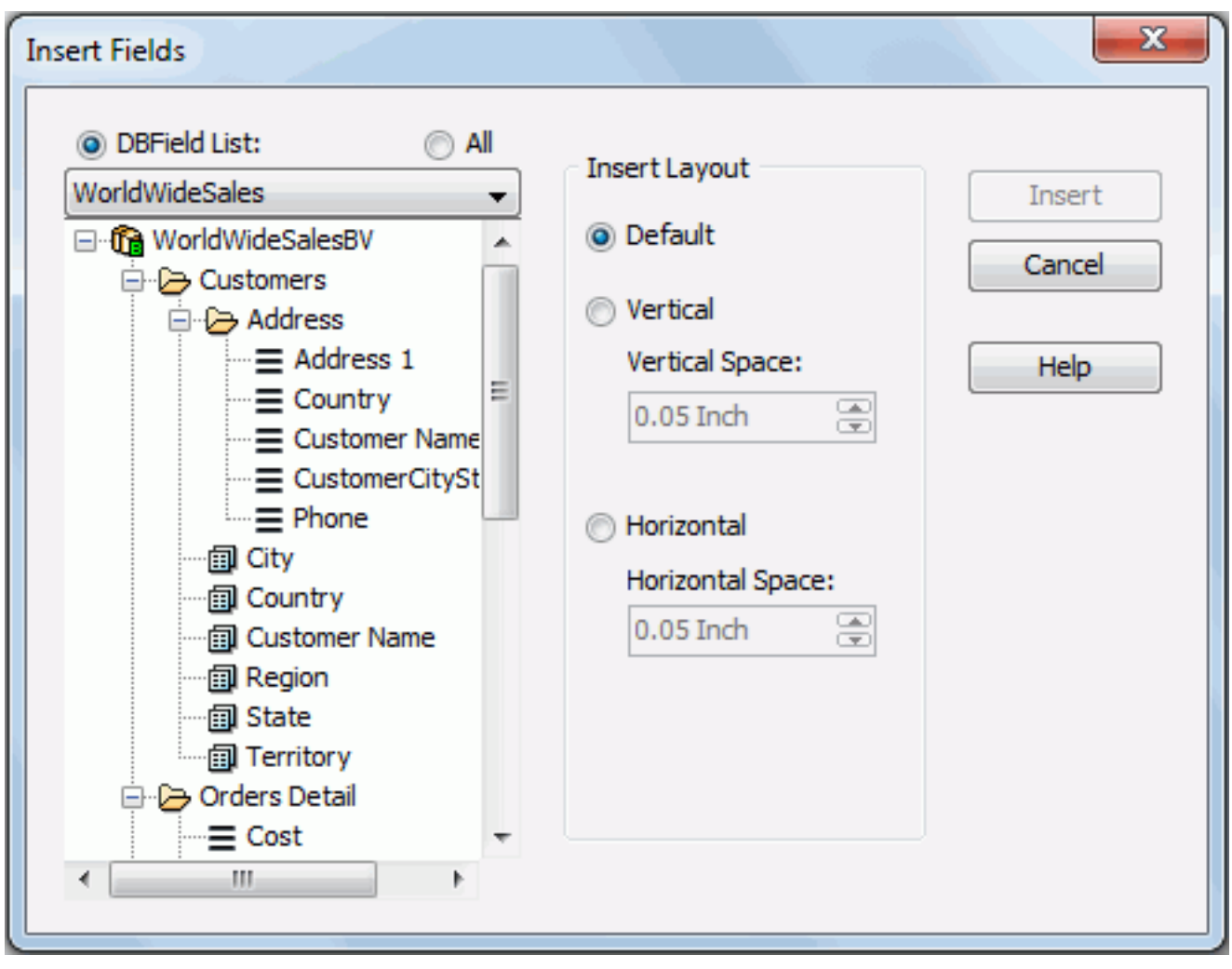
Inserting a formula field in a report

A formula field can be inserted in the report areas listed in Component placement. Before you can insert a formula field into a report, you need to define it. For details about how to define formulas, see Creating a formula.

To insert a formula field into a report, you can use either of the following two ways:

- **Using dialog**
  1. Position the mouse pointer at the destination where you want to insert the formula and do one of the following:
     - Click **Insert** > **Formula** on the menu bar.
     - Click the **Insert Formula** button on the Insert toolbar.

     The Insert Fields dialog appears.



  2. Select the required formula from the Formulas node, then specify its layout in the Insert Layout box.
  3. Click the **Insert** button to insert the formula into the destination.
- **By dragging and dropping**

  Select the required formula from the Formulas node of the Resource View panel, then drag it to the destination.

When a formula field has been inserted into a report, you can customize its appearance and behavior by setting properties in the Report Inspector, which can be saved as a CSS style for future use (for details, see Creating a CSS style).

Notes:

- No matter which way you take, if the given formulas are not what your want, you can click the <Add Formula...> item to create one as required.
- If you insert a formula into the detail panel of a component, the name of the formula will automatically be inserted as a label into the corresponding header panel. Otherwise the formula and its name label will be placed in the same panel. If you do not want to insert the name label automatically, you can uncheck Insert field name label with field in the Options dialog (**File** > **Options** > **Component** > **Insert field name label with field**).

Changing the display type of a formula field

The display type of a formula field can be changed, that is to say, you can map the formula field to another value or image to be displayed instead.

To change the display type a formula field:

1. Right-click the formula field and select **Display Type** from the shortcut menu.
2. In the Display Type dialog, specify the required display type in the Type box, and set the options for the selected type in the Web Options panel.
3. Upon finishing, click the **OK** button to close the dialog.

For details about display types, see Changing the display type of a label.

Applying web actions to a formula field

A formula field can be bound with some web actions. This allows you to customize a formula field to make it respond to some events, and execute corresponding actions, such as sorting and filtering.

To apply web actions to a formula field:

1. Right-click the formula field and select **Display Type** from the shortcut menu.
2. In the Web Behaviors box of the Display Type dialog, choose an event from the Events column, then click in the Actions column and click that appears in the text box.
3. In the Web Action List dialog, select a web action and define the action as required.
4. Click in the Display Type dialog to add more web behavior lines and specify the events and actions accordingly. If a web behavior is not required, click to remove it.
5. Adjust the order of the added web actions by clicking or . Then, when an event that has been bound with more than one action happens, the upper action will be triggered first.
6. When done , click **OK** to accept the settings.

For more information about web actions, see Applying web actions to a label.

Binding a link to a formula field

You can make a formula field linked with another report, a location specified by a URL, an e-mail address, or a Blob data type field. The link can either be a simple link or a conditional link. With conditional link, different targets can be loaded based on different conditions.

To bind a link to a formula field:

1. Right-click the formula field and click **Link** on the shortcut menu.
2. In the Insert Link dialog, specify whether it is a conditional link or a simple link, then select the link target and set the options for the link target as required.
3. When done, click **OK**.

For details about different link targets, see Binding a link to a label.

Adding conditional formats to a formula field

You can add some conditional formats to a formula field, then when a specified condition is fulfilled, the format bound with the condition will be applied to the field values automatically for distinguishing.

To add conditional formats to a formula field:

1. Right-click the formula field and select **Conditional Formatting** from the shortcut menu.
2. In the Conditional Formatting dialog, click the button to add a condition.
3. Set the format that will be applied to the field values when the specified condition is fulfilled.
4. Repeat the above two steps to add more conditions and define the format for each condition.
5. When done, click **OK** to save the settings.

For more details about conditional formatting, see Adding conditional formats to a DBField.

See an example: The SampleComponents catalog, included with JReport Designer, contains reports that have examples of how each component type could be used in a report. For the formula field example, open `<install_root>\Demo\Reports\SampleComponents\UsingFormula.cls`.

Summary fields

A summary is a special kind of formula. A summary generates a count, average, sum, standard deviation or other transformation of a set of data values. A summary applies to a defined group of data.

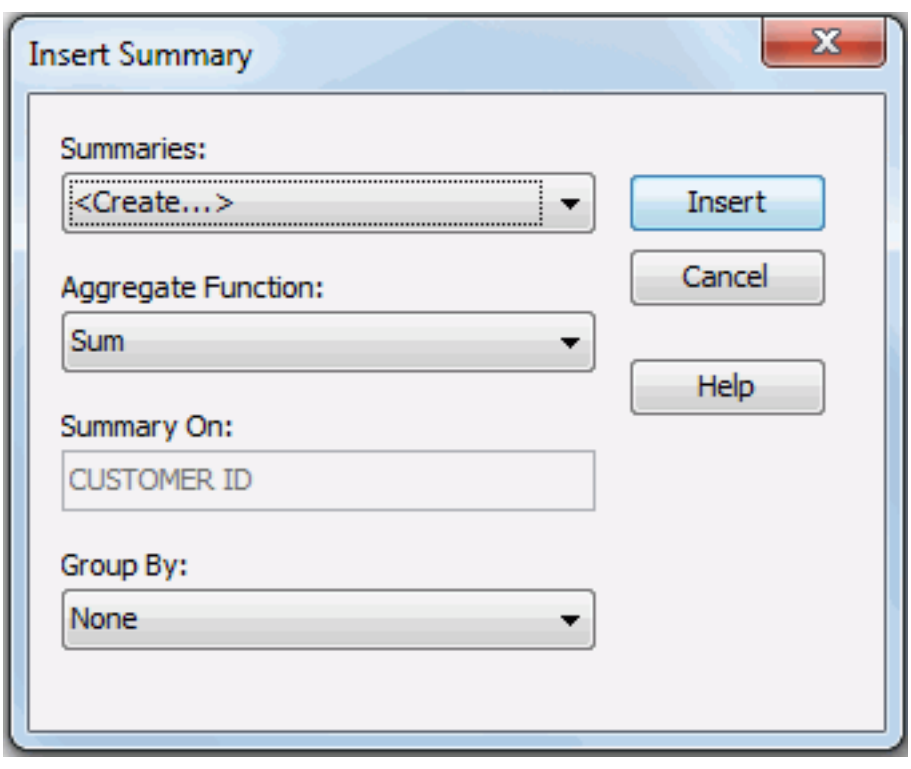
Inserting a summary field in a report

Summaries can be inserted in the report areas listed in Component placement. Before you can insert a summary field into a report, you need to define it. For details about how to define summaries, see Creating a summary.

To insert a summary field into a report, you can use either of the following two ways:

- **Using dialog**
  1. Select the DBField in the report to which you want the summary to be applied.
  2. Do one of the following:
     - Click **Insert** > **Summary** on the menu bar.
     - Click the **Insert Summary** button on the Insert toolbar
     - Right-click the DBField and select **Summary Function** from the shortcut menu.

     The Insert Summary dialog appears.



  3. From the Function drop-down list, specify the function of the summary.
  4. If the report is grouped, select to which group level the summary will be applied from the Group By drop-down list. If None is select, the summary will be added based on the whole report.
  5. Click the **Insert** button to insert the summary into the destination.
- **By dragging and dropping**

  Select the required summary from the Resource View panel, then drag it to the destination. If the given summaries are not what you want, click the **<Add Summary...>** item to create one as required.

When a summary field has been inserted into a report, you can customize its appearance and behavior by setting properties in the Report Inspector, which can be saved as a CSS style for future use (for details, see Creating a CSS style).

Notes:

- Summary fields can only be inserted into the header/footer/detail panel of a banded object or cells in the header/footer/detail panel of a table. However, if you have specified a group by field when creating the summary, it can be inserted into the corresponding group panels.
- If you insert a summary into the detail panel of a component, the name of the summary will automatically be inserted as a label into the corresponding header panel. Otherwise the summary and its name label will be placed in the same panel. If you do not want to insert the name label automatically, you can uncheck Insert field name label with field in the Options dialog (**File** > **Options** > **Component** > **Insert field name label with field**).

Changing the display type of a summary field

The display type of a summary field can be changed, that is to say, you can map the summary field to another value or image to be displayed instead.

To change the display type a summary field:

1. Right-click the summary and select **Display Type** from the shortcut menu.
2. In the Display Type dialog, specify the required display type in the Type box, and set the options for the selected type in the Web Options panel.
3. Upon finishing, click the **OK** button to close the dialog.

For details about display types, see Changing the display type of a label.

Applying web actions to a summary field

A summary field can be bound with some web actions. This allows you to customize a summary field to make it respond to some events, and execute corresponding actions, such as sorting and filtering.

To apply web actions to a summary field:

1. Right-click the summary field and select **Display Type** from the shortcut menu.
2. In the Web Behaviors box of the Display Type dialog, choose an event from the Events column, then click in the Actions column and click that appears in the text box.
3. In the Web Action List dialog, select a web action and define the action as required.
4. Click in the Display Type dialog to add more web behavior lines and specify the events and actions accordingly. If a web behavior is not required, click to remove it.
5. Adjust the order of the added web actions by clicking or . Then, when an event that has been bound with more than one action happens, the upper action will be triggered first.
6. When done , click **OK** to accept the settings.

For more information about web actions, see Applying web actions to a label.

Binding a link to a summary field

You can make a summary field linked with another report, a location specified by a URL, an e-mail address, or a Blob data type field. The link can either be a simple link or a conditional link. With conditional link, different targets can be loaded based on different conditions.

To bind a link to a summary field:

1. Right-click the summary field and click **Link** on the shortcut menu.
2. In the Insert Link dialog, specify whether it is a conditional link or a simple link, then select the link target and set the options for the link target as required.
3. When done, click **OK**.

For details about different link targets, see Binding a link to a label.

Adding conditional formats to a summary field

You can add some conditional formats to a summary field, then when a specified condition is fulfilled, the format bound with the condition will be applied to the field values automatically for distinguishing.

To add conditional formats to a summary field:

1. Right-click the summary field and select **Conditional Formatting** from the shortcut menu.
2. In the Conditional Formatting dialog, click the button to add a condition.
3. Set the format that will be applied to the field values when the specified condition is fulfilled.
4. Repeat the above two steps to add more conditions and define the format for each condition.
5. When done, click **OK** to save the settings.

For more details about conditional formatting, see Adding conditional formats to a DBField.

See an example: The SampleComponents catalog, included with JReport Designer, contains reports that have examples of how each component type could be used in a report. For the summary field example, open `<install_root>\Demo\Reports\SampleComponents\UsingSummaries.cls`.

Parameter fields

A parameter in JReport is a variable whose value is determined at runtime. The runtime parameters help you dynamically control your report results.

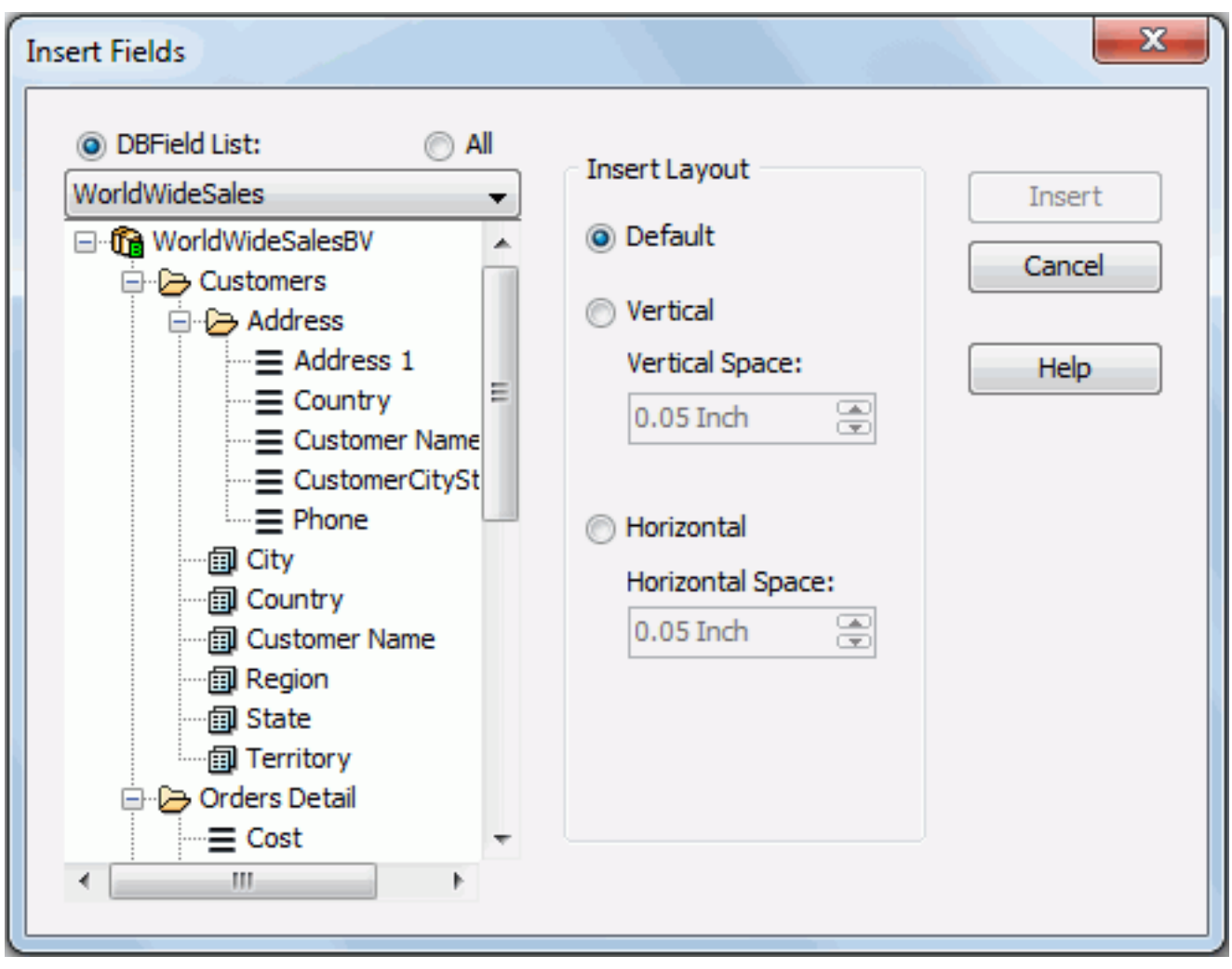
Inserting a parameter field in a report

Parameter fields can be inserted in the report areas listed in Component placement. Before you can insert a parameter field into a report, you need to create it. For details about how to create parameters, see Creating a parameter.

To insert a parameter field into a report, you can use either of the following methods:

- **Using dialog**
  1. Position the mouse pointer at the destination where you want to insert the parameter and do one of the following:
     - Click **Insert** > **Parameter** on the menu bar.
     - Click the **Insert Parameter** button on the Insert toolbar.

     The Insert Fields dialog appears.



  2. Select the required parameter from the Parameters node, then specify its layout in the Insert Layout box.
  3. Click the **Insert** button to insert the parameter into the destination.
- **By dragging and dropping**

  Select the required parameter from the Resource View panel, then drag it to the destination.

When a parameter field has been inserted into a report, you can customize its appearance and behavior by setting properties in the Report Inspector, which can be saved as a CSS style for future use (for details, see Creating a CSS style).

Notes:

- No matter which way you take, if the given parameters are not what your want, click the <Add Parameter...> item to create one as required.
- If you insert a parameter into the detail panel of a banded object or table, the name of the parameter will automatically be inserted as a label into the corresponding header panel. Otherwise the parameter and its name label will be placed in the same panel. If you do not want to insert the name label automatically, you can uncheck Insert field name label with field in the Options dialog (**File** > **Options** > **Component** > **Insert field name label with field**).

Changing the display type of a parameter field

The display type of a parameter field can be changed, that is to say, you can map the parameter field to another value or image to be displayed instead.

To change the display type a parameter field:

1. Right-click the parameter field and select **Display Type** from the shortcut menu.
2. In the Display Type dialog, specify the required display type in the Display As box, and set the options for the selected type.
   - To change the display type of a parameter to list or drop-down list, select the item List or Drop-down List, and then in the Web Options panel, set the name and title. You may notice that a default list item had been added, and its value is the parameter. If you check Use Runtime Value, when running the report, you need to specify a value for the parameter, and then the item same as the value will be selected in the list or drop-down list by default; if you uncheck the option, the Selected text box will be enabled for you to input a value and the item same as this value will be selected. For the List type, you can check Allow Multiple Selections to enable more than one item selected.
   - For other display types, refer to Changing the display type of a label for details.
3. Upon finishing, click the **OK** button to close the dialog.

Applying web actions to a parameter field

A parameter field can be bound with some web actions. This allows you to customize a parameter field to make it respond to some events, and execute corresponding actions, such as sorting and filtering.

To apply web actions to a parameter field:

1. Right-click the parameter field and select **Display Type** from the shortcut menu.
2. In the Web Behaviors box of the Display Type dialog, choose an event from the Events column, then click in the Actions column and click that appears in the text box.
3. In the Web Action List dialog, select a web action and define the action as required.
4. Click in the Display Type dialog to add more web behavior lines and specify the events and actions accordingly. If a web behavior is not required, click to remove it.
5. Adjust the order of the added web actions by clicking or . Then, when an event that has been bound with more than one action happens, the upper action will be triggered first.
6. When done , click **OK** to accept the settings.

For more information about web actions, see Applying web actions to a label.

Binding a link to a parameter field

You can make a parameter field linked with another report, a location specified by a URL, an e-mail address, or a Blob data type field. The link can either be a simple link or a conditional link. With conditional link, different targets can be loaded based on different conditions.

To bind a link to a parameter field:

1. Right-click the parameter field and click **Link** on the shortcut menu.
2. In the Insert Link dialog, specify whether it is a conditional link or a simple link, then select the link target and set the options for the link target as required.
3. When done, click **OK**.

For details about different link targets, see Binding a link to a label.

Adding conditional formats to a parameter field

You can add some conditional formats to a parameter field, then when a specified condition is fulfilled, the format bound with the condition will be applied to the field values automatically for distinguishing.

To add conditional formats to a parameter field:

1. Right-click the parameter field and select **Conditional Formatting** from the shortcut menu.
2. In the Conditional Formatting dialog, click the button to add a condition.
3. Set the format that will be applied to the field values when the specified condition is fulfilled.
4. Repeat the above two steps to add more conditions and define the format for each condition.
5. When done, click **OK** to save the settings.

For more details about conditional formatting, see Adding conditional formats to a DBField.

See an example: The SampleComponents catalog, included with JReport Designer, contains reports that have examples of how each component type could be used in a report. For the parameter field examples, open the following reports: `<install_root>\Demo\Reports\SampleComponents\Parameter.cls`, and `<install_root>\Demo\Reports\SampleComponents\ParameterField.cls`.

Special fields

Special fields are defined by JReport and allow you to easily obtain system information and report-related data and add it to your report. The special fields are as follows:

Date-time

Lists all the datetime type special fields.

- **Print Date**
  Prints today's date (or the current date from your computer).
- **Print Time**
  Prints the current time from your computer.
- **Fetch Date**
  Prints the date when the data is retrieved from the database.
- **Fetch Time**
  Prints the time when the data is retrieved from the database.
- **Modified Date**
  Prints the last modified date of the catalog.
- **Modified Time**
  Prints the last modified time of the catalog.

Computed

Lists all the special fields that are computed based on the report.

- **Record Number**
  Prints the record number (usually placed in the Details panel).
- **Group Name**
  Prints the group name (usually placed in the Group Header/Footer panel).
- **Total Records**
  Prints the total number of records after all the filter conditions are performed, except the ones created in the Filter dialog of Page Report Studio, and the Group Filter dialog and top N or bottom N condition in JReport Designer.
- **Total Fetched Records**
  Prints the total number of records which take part in grouping calculation. The possible result of the special field is as follows:
  - If you don't set any filter condition in the Filter dialog of Page Report Studio, print the very number of the record obtained after setting the property Maximum Records.
  - If you set filter conditions in the Filter dialog of Page Report Studio, print the number of records obtained after performing the filters, even though you have set the property Maximum Records before setting the filters.
- **Group Number**
  Prints the group number (usually placed in the Group Header/Footer panel).
- **Total Group Number**
  Prints the total group number (usually placed in the Group Header/Footer panel).
- **Page Number**
  Prints the page number of the report which it is placed in.
- **Global Page Number**
  Prints the global page number of the whole report wherever it is placed.
- **Total Page Number**
  Prints the total page number of the report which it is placed in.
- **Global Total Page Number**
  Prints the global total page number of the whole report wherever the field is placed.
- **Page N of M**
  Prints a specific page number out of the total page number of the report which the field is placed in. You can specify the format of this special field. To do this, select this special field, then in the Report Inspector, find the Format property and choose an item from the shortcut list. Besides the items listed here, you can also customize the format. For example, if you want it displayed as "This is page n of m pages", you can input the format string "This is page @PageNumber of @TotalPageNumber pages" into the Format text box.
- **Global Page N of M**
  Prints a specific global page number out of the global total page number. You can specify the format of this special field in the Report Inspector. The operation is the same as Page N of M.
- **SQL Statement**
  Prints the SQL statements used to execute the query.

Note: Page Number, Total Page Number and Page N of M in the following conditions in versions after V8 are different from those in versions prior to V7:

- In V7 and prior, the Page Number, Total Page Number and Page N of M which are placed in a subreport will calculate the page number of the subreport if the value of the property Embed in a subreport is set to true. And if you set Embed to false, the Page Number, Total Page Number and Page N of M will work as global special fields, it will calculate the page number of the primary report.
- While in V8 and after, the Page Number, Total Page Number and Page N of M which are placed in a subreport will calculate the page number of the subreport no matter you set the property Embed to true or fals

System

Lists all the special fields which get values from the system.

- **User Name**
  Prints the User ID used when logging onto JReport Server. When viewing the report in JReport Designer, it will return NULL, while when running the report on JReport Server, it will return the name used to log onto the server. The User Name special field can be used in different data sources as follows:

	Where Condition of Query	HDS (General)	HDS (XML)	UDS	SQL File
User Name	Supported.	Supported in the Parameter box in the Add General Hierarchical Data Source dialog.	Supported.	Supported in the Parameter box in the Add User Defined Data Source dialog.	Supported in the SQL statement.
User Name used in a formula	Supported.	Supported.	Supported.	Supported.	Supported.

  It should be used as *username* in formulas and stored procedures and as @*username* in UDS, imported SQL, and query filters.
- **Task ID**
  Prints the internal task ID (a unique time stamp). This special field only works in the JReport Server environment. It returns a NULL value when running in JReport Designer.
- **Report Path on Server**
  Prints the full path of the report on the JReport Server resource tree. If the report is not running on JReport Server, it returns the full disk path of the report. It returns a NULL value if the report is unsaved.
- **Report Path on Disk**
  Prints the full disk path of the report. It returns a NULL value if the report is unsaved.
- **Catalog Path on Server**
  Prints the full path of the report's catalog on the JReport Server resource tree. If the report is not running on JReport Server, it returns the full disk path of the catalog. It returns a NULL value if the report is unsaved.
- **Catalog Path on Disk**
  Prints the full disk path of the report's catalog. It returns a NULL value if the report is unsaved.

Referencing special fields in a formula

Some special fields can be used in formulas. They are User Name, Print Date, Print Time, Fetch Date, Fetch Time, Modified Date, Modified Time, Record Number, Page Number, Global Page Number, Total Records and Total Fetched Records. They can all be divided into two types (page level and constant level) according to the time when their values are ready.

Page level

The value of this special field is ready at the time when the report result is generated. A formula which references a page-level special field is treated as a page-level formula. In which, the formula is calculated more than once.

The special fields of this type are: Fetch Date, Fetch Time, Record Number, Page Number, Global Page Number, Print Date, Print Time, Total Records and Total Fetched Records.

Constant level

The value of this special field is ready at any time before the engine runs. A formula which references a constant-level special field is treated as a constant-level formula, and is calculated only once.

The special fields of this type are: User Name, Modified Date and Modified Time.

Inserting a special field into a report

Special fields can be inserted in the report areas listed in Component placement.

To insert a special field into a report:

1. Position the mouse pointer at the destination where the special field is to be inserted.
2. Do one of the following:
   - Click **Insert** > **Special Fields**, and click the corresponding command on the submenu.
   - Click the down arrow on the **Insert Special Field** button on the Insert toolbar and select the required special field from the drop-down list.
3. Click in the destination to insert the special field.

When a special field has been inserted into a report, you can customize its appearance and behavior by setting properties in the Report Inspector, which can be saved as a CSS style for future use (for details, see Creating a CSS style).

Changing the display type of a special field

The display type of a special field can be changed, that is to say, you can map the special field to another value or image to be displayed instead.

To change the display type a special field:

1. Right-click the special field and select **Display Type** from the shortcut menu.
2. In the Display Type dialog, specify the required display type in the Type box and set the options for the selected type in the Web Options panel.
3. Upon finishing, click the **OK** button to close the dialog.

For details about display types, see Changing the display type of a label.

Applying web actions to a special field

A special field can be bound with some web actions. This allows you to customize a special field to make it respond to some events, and execute corresponding actions, such as sorting and filtering.

To apply web actions to a special field:

1. Right-click the special field and select **Display Type** from the shortcut menu.
2. In the Web Behaviors box of the Display Type dialog, choose an event from the Events column, then click in the Actions column and click that appears in the text box.
3. In the Web Action List dialog, select a web action and define the action as required.
4. Click in the Display Type dialog to add more web behavior lines and specify the events and actions accordingly. If a web behavior is not required, click to remove it.
5. Adjust the order of the added web actions by clicking or . Then, when an event that has been bound with more than one action happens, the upper action will be triggered first.
6. When done , click **OK** to accept the settings.

For more information about web actions, see Applying web actions to a label.

Binding a link to a special field

You can make a special field linked with another report, a location specified by a URL, an e-mail address, or a Blob data type field. The link can either be a simple link or a conditional link. With conditional link, different targets can be loaded based on different conditions.

To bind a link to a special field:

1. Right-click the special field and click **Link** on the shortcut menu.
2. In the Insert Link dialog, specify whether it is a conditional link or a simple link, then select the link target and set the options for the link target as required.
3. When done, click **OK**.

For details about different link targets, see Binding a link to a label.

Adding conditional formats to a special field

You can add some conditional formats to a special field, then when a specified condition is fulfilled, the format bound with the condition will be applied to the field values automatically for distinguishing. However, conditional formatting applies only to the two special fields: Page Number and User Name.

To add conditional formats to the special field Page Number/User Name:

1. Right-click the special field and select **Conditional Formatting** from the shortcut menu.
2. In the Conditional Formatting dialog, click the button to add a condition.
3. Set the format that will be applied to the field values when the specified condition is fulfilled.
4. Repeat the above two steps to add more conditions and define the format for each condition.
5. When done, click **OK** to save the settings.

For more details about conditional formatting, see Adding conditional formats to a DBField.

See an example: The SampleComponents catalog, included with JReport Designer, contains reports that have examples of how each component type could be used in a report. For the special field example, open `<install_root>\Demo\Reports\SampleComponents\ForSpecialFields.cls`.

Text boxes

Similar to a word processing document, a text box is an object that holds text (a single character, a single word, entire sentences, or paragraphs). Editing in a text box is just like working in a word processor. You can cut, copy and paste selected text easily, and format the text as needed. In addition, a text box can hold other components.

Inserting a text box into a report

A text box can be inserted in the report areas listed in Component placement.

To insert a text box into a report, use either of the following ways:

- **Using menu/toolbar command**

  Click **Insert** > **Text Box** on the menu bar, or the **Insert Text Box** button on the Insert toolbar. If the mouse pointer is in the report body or a tabular cell, a text box will be inserted there upon finishing. Otherwise, you need to click the mouse button in the desired location to place the text box there.

- **By dragging and dropping**

  From the Toolbox panel, drag **Text Box** to the desired location in the report which allows the insertion of a text box, and a text box will then be created.

After a text box has been inserted into a report, you can double-click it to input text. You can also customize its appearance and behavior by setting properties in the Report Inspector, which can be saved as a CSS style for future use (for details, see Creating a CSS style).

Formatting paragraphs in a text box

Paragraphs are created when text is entered into a text box. In JReport, you can easily format the paragraphs, for example, the alignment style, indentation format or line spacing. To do this:

1. Right-click anywhere in the paragraph and click **Paragraph** on the shortcut menu.
2. In the Paragraph tab of the Format Text dialog,
   - To specify the alignment style of the paragraph, select the required option from the Alignment drop-down list: left, center, right or justified.
   - To set the indentation formats of the paragraph, in the Indentation box, specify the left and right indent distance in the Left and Right fields. If you want to set the paragraph to be indented, select **first line** or **hanging** from the Special drop-down list, then input a number in the By text box. This will create a first-line indent or hanging indent in the paragraph using the inputted number as the indent distance.
   - To adjust the spacing before or after the paragraph, enter the required amount of spacing in the Before or After field.
   - To change the spacing between the lines in the paragraph, select the required item from the Line Spacing drop-down list. If at least, exactly or multiple is selected, enter the required amount of space or the number of lines in the At box.
3. When done, click **OK** to accept the changes.

You can also set the indentation formats of a paragraph by simply moving the tap stop icons on the ruler. To do this, position the mouse pointer anywhere on the paragraph, then,

- To create a first-line indent or hanging indent, drag to the position where you want the text to start.
- To change the left indent, drag to the position where you want the paragraph to start.
- To change the right indent, drag to the position where you want the paragraph to end.

Formatting the text in a text box

The text you enter into a text box can be formatted to suit your requirements. For example, you can change the font size, color, alignment, and highlight the text with a distinguishing color. To do this:

1. Select the text you want to format, right-click and click **Font** on the shortcut menu.
2. In the Font tab of the Format Text dialog, specify the font settings.
3. When done, click **OK** to accept the changes.

You can also format the selected text by clicking the corresponding buttons on the Format toolbar, or commands of the Text Style submenu on the Format menu.

Finding or replacing a text string

To find a character string in a text box:

1. Right-click anywhere in the text box and select **Find a String** from the shortcut menu.
2. In the Find tab of the Find and Replace dialog, specify the string you want to search for and set the other settings as required.
3. Click the **Find Next** button to start finding the specified string.

Similar to finding text, if you want to replace a string in the text box with another one,

1. Right-click anywhere in the text box and select **Replace** from the shortcut menu, the Find and Replace dialog will then be displayed with the Replace tab selected.
2. Specify the settings according to your requirements and click the **Replace** or **Replace All** button to replace the next or all the specified string with another one.

Importing and exporting text

To import text into a text box from an external file (.txt or .rtf):

1. Position the mouse pointer at the location where you want to insert the imported text, then right-click and select **Import External File**.
2. In the Import a File dialog, select the .txt or .rtf file you want to import and click **OK**.

   The text of the file is inserted in the text box.

To export the text from a text box:

1. Position the mouse pointer on the text, right-click and select **Save to RTF File**.
2. In the Save RTF File dialog, specify the name of the file and click **OK**.

See an example: The SampleComponents catalog, included with JReport Designer, contains reports that have examples of how each component type could be used in a report. For the text box example, open `<install_root>\Demo\Reports\SampleComponents\ForTextBox.cls`.

Tables

Tables give you great control over how to present data, including placing fields, grouping them, and sorting them. A table is composed of rows and columns, and each contains several cells. With such a structure a table is a good way to show any two-dimensional dataset.

In JReport, tables are data containers, which means that a table displays the results of a query and you can apply some filters to narrow down the records displayed in it. Also, when a table is inserted into another data container, you can set up data container link between the table and its parent.

Meanwhile, in order to meet the requirements from voice agents used by people with eyesight disabilities, JReport supports exported table components in the HTML data table format to get a section 508 compliant result. For detailed descriptions, refer to Accessibility.

Pick a task from the following:

- Inserting a table
- Modifying a table
- Grouping the data
- Sorting the data
- Filtering the data
- Managing columns/rows in a table
- Formatting cells in a table
- Setting up a data container link

See an example: The SampleComponents catalog, included with JReport Designer, contains reports that have examples of how each component type could be used in a report. For the table component example, open `<install_root>\Demo\Reports\SampleComponents\TableReport.cls`.

Inserting a table

To insert a table into a report or a library component, take the following steps:

1. Position the mouse pointer at the destination where you want to insert the table. A table can be inserted in the report areas listed in Component placement.

2. Do one of the following:
   - Click **Insert** > **Table** on the menu bar.
   - Click the **Insert Table** button on the Insert toolbar.
   - Drag **Table** or **Horizontal Table** from the Toolbox panel.

3. In the Table Type dialog, choose a type for the table:
   - **Table (Group Above)**
     Creates a table with group information above the detail panel.
   - **Table (Group Left)**
     Creates a table with group information left to the detail panel.
   - **Table (Group Left Above)**
     Creates a table with group information left above the detail panel.
   - **Summary Table**
     Creates a table with only group and summary information.

4. Click **OK** to confirm the type and the Create Table wizard is then displayed.

5. In the Data screen of the wizard,
   - If you are inserting a table into a page report, define or select the dataset on which the table will be created.
     - **New**
       If checked, the table will be built on a new dataset created from the catalog resources.
     - **Existing**
       If checked, specify the dataset in the current report from which the table will be created.
     - **Current**
       If checked, the table will be built on the dataset which is used by its parent.
   - If you are inserting a table into a web report or a library component, select a business view in the current catalog on which the table will be created.

6. In the Display screen, add the fields from the Resources box to be displayed in the table. Modify the display name of any added field if needed by double-clicking its Display Name column. If required, click the **Sort Fields By** button to specify how to sort data in the table.

   If you are inserting a table into a web report or a library component, you can also specify a title for the table in the Title text field.

7. In the Group screen, add the fields as the grouping criteria, and then specify the sorting direction of each group level in the Sort column.

8. To add summaries, go to the Summary screen. In the sum on box (the box on the right), specify the group to which the summary will be applied (if you select Table, the summary will be based on the whole dataset), add a field from the Resources box as the sum on field, and then select the function for the summary from the Aggregate Function column. From the Position and Column columns, specify where to place the added sum on field in the table.

9. In the Filter screen, define some filter conditions to filter the data displayed in the table. If you are inserting the table into a web report or a library component, you can select a predefined filter of the specified business view if there is from the Filter drop-down list to apply, or select **User Defined** in the list to define a new filter.

10. In the Style screen, specify the style of the table as required.

    If you have specified to insert the table into a banded object in a page report, by default the table will inherit its parent's style. If you want to apply another style to the table, uncheck the **Inherit Style** option and then select the required style from the Style box. For more information, see Inheriting the style from a data container.

11. Click **Finish** to insert the table.

    If you have used the menu or toolbar command to insert the table and specified to insert it to position other than the report body or tabular cell, after clicking Finish in the wizard, you need to click the mouse button in the destination once again in order to insert the table there.

See also the Create Table wizard for page report or web report/library component for detailed information about options in the wizard.

After a table has been created, you can customize its appearance and behavior by setting properties in the Report Inspector, which can be saved as a CSS style for future use (for details, see Creating a CSS style).

Note:

- When you want to create a table on a hierarchical data source, pay attention to the unique features of HDS. For details, refer to Developing reports from HDS.
- When you insert a table into a banded report with the dataset inherited from the banded report, you cannot insert the table into the BandedPageHeader and BandedPageFooter panel.
- The content of the table in some pages that cannot be displayed will be cut off if the height of the table is higher than that of the page, however, when exporting the table to an Excel file with column format or data format, or to text with delimited format, or to XML, no content will be cut off.

Modifying a table

Once a table has been created, you can further modify it by accessing its shortcut menu wizard which is composed by a set of screens that are similar to the wizard screens used to create the table. For example, you can change the data used by the table, apply some grouping, sorting criteria to the table, and so on.

To modify a table after it has been created:

1. Right-click the table and select **Table Wizard** from the shortcut menu to display the Table Wizard.
2. In the Data screen, specify a new data source for the table if required.
3. In the Display screen, specify the fields you want to display in the table and how to sort records in the table.
4. In the Group screen, specify the grouping criteria, and then the sorting direction of each group level in the Sort column.
5. In the Style screen, select the style you want to apply to the table.
6. When done, click **Finish** to accept the changes.

See also the Table Wizard for page report or web report/library component for detailed information about options in the wizard.

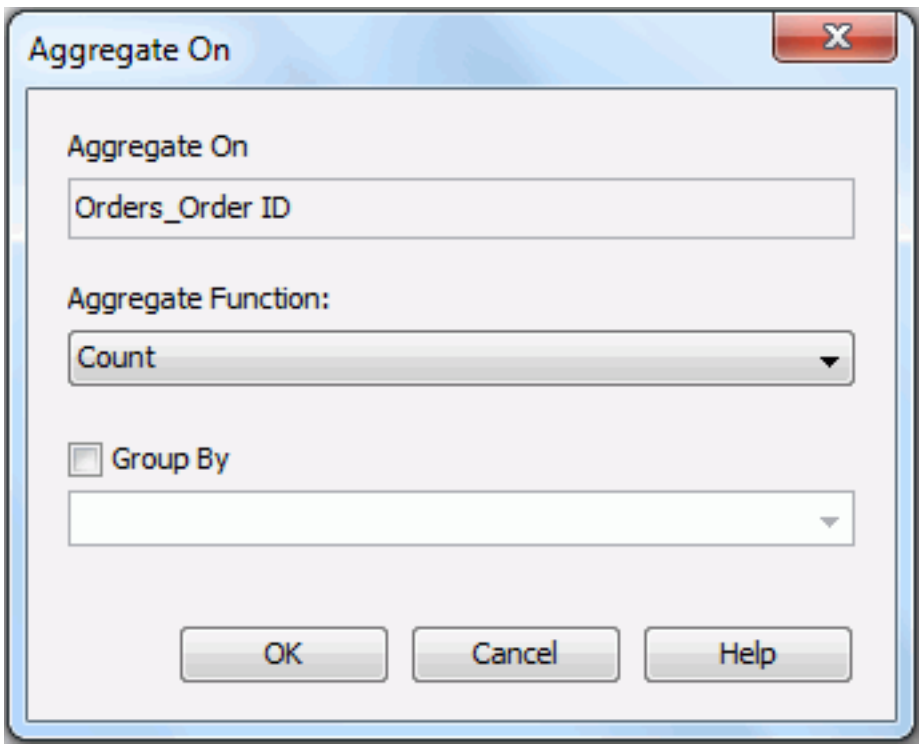
Notes:

- Due to the specialty of hierarchical data sources, you are not allowed to specify an HDS as the new dataset of the table.
- In the Display screen, the newly added fields will be put to the end of the existing fields, and you cannot re-adjust the order of the fields as you did when creating the table.
- For a page report, when you have specified a new dataset for the table, if there are existing fields in the table that cannot match the new dataset, they will be marked with a red cross icon. You can use the Replace button to select fields from the new dataset to replace them. Then when you finish the modification, the new dataset and fields you specify will be displayed in the table and those remaining fields that cannot be matched to fields in the new dataset will be removed from the table automatically.
- For a web report, if you specify a new business view for the table, the current table data will be lost and when you finish editing, a new table will be created.

Aggregating on a detail column

You can summarize the data in a detail column if required. To do this:

1. Right-click the detail column and select **Aggregate On** from the shortcut menu. The Aggregate On dialog appears.



2. From the Aggregate Function drop-down list, specify a function to summarize the field in the detail column.
3. Specify the Group By option.
   - If the table has groups and you want the summary to be applied on certain group level, check **Group By** and select the corresponding group by field from the drop-down list below.
   - If you want the summary to be applied on the whole dataset, check **Group By** and do not select any field from the drop-down list below.
   - If you want to create a dynamic summary, keep **Group By** unchecked. Then the summary will be applied on every group level and the whole dataset at the same time.
4. When done, click **OK**. Data in the detail column will be summarized based on the group by setting using the specified function.

At the same time,

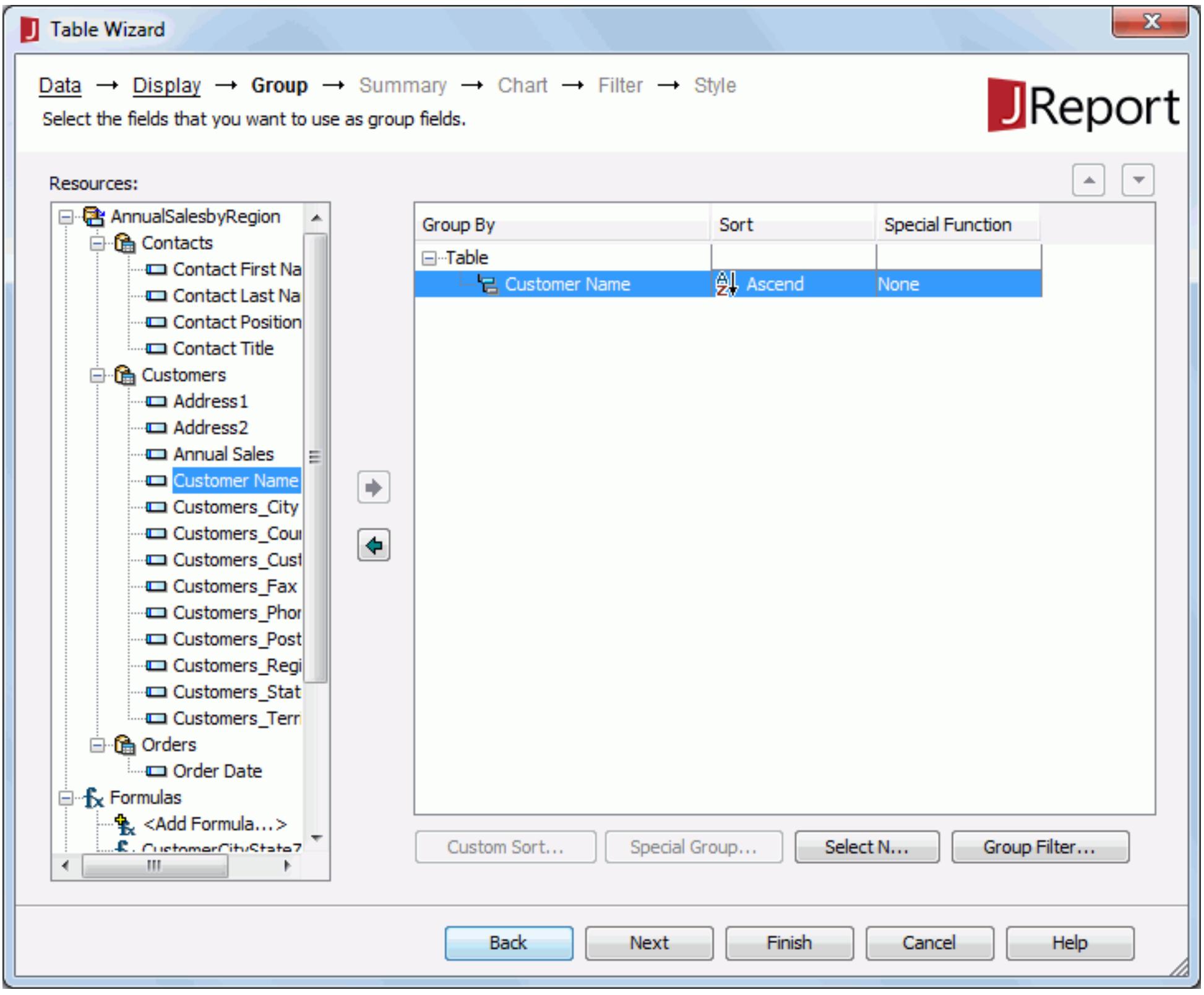
- For a page report, a summary which is given a default name Function_DetailFieldName will be created and saved in the current catalog. You can use it in other page reports if required.
- For a web report or library component, a dynamic aggregation which is given a default name Function_DetailFieldName will be created and saved along with the web report or library component as its resource. You can use it in this web report or library component only.

Grouping the data

Data in a table is usually organized into groups based on certain criteria. You can define multi-level groups in your table either when creating it or after the table has been built. In this section, you will be shown how to define groups in an existing table.

To group data of an existing table:

1. Select the table and do one of the following:
   - Click **Insert** > **Group** on the menu bar.
   - Click the **Insert Group** button on the Insert toolbar.
   - Right-click the table and select **Table Wizard** from the shortcut menu.
2. In the Table Wizard, select the **Group** screen.



3. Select the field from the Resources box as the grouping criterion, and then click to add it to the group by box (the box on the right). And if the type of the group by field is Numeric/String/Date/Time, you can select a special function for the field as required.
4. Set the sorting manner of groups at this group level in the Sort column.
5. Click the **Select N** button to specify the Select N condition.
6. If you want to filter groups at this group level, click the **Group Filter** button, and then specify the filter conditions.
7. Repeat the above steps to add more groups.
8. Click **Finish** to apply the settings.

One or more GH-GF (group header and group footer) row pairs with the selected group fields will have been added to the table.

Note: The following SQL type of data cannot be grouped: Db.SQL_BINARY, Db.SQL_BLOB, Db.SQL_CLOB, Db.SQL_LONGVARCHAR, Db.SQL_LONGVARBINARY, Db.SQL_VARBINARY and Db.SQL_OTHER.

The following are some specific topics about grouping:

- Specifying special function for group by field
- Setting the sort manner
- Specifying the Select N condition
- Filtering the groups
- Grouping data dynamically

Specifying special function for group by field

If the field which is added as the grouping criterion is of Numeric/String/Date/Time type, you can select a special function for the field in the Special Function column to group the data as required. This is called "grouping data by intervals" in JReport. It enables you to group data more clearly and logically, and to summarize data more effectively.

The following is a short description of each special function.

Functions for Numeric-typed group by fields

Function	Description
None	All the records that have the same field value will be displayed together as a group.
Up to 5	The records will be grouped by intervals of 5.
Up to 10	The records will be grouped by intervals of 10.
Up to 50	The records will be grouped by intervals of 50.
Up to 100	The records will be grouped by intervals of 100.
Up to 500	The records will be grouped by intervals of 500.
Up to 1000	The records will be grouped by intervals of 1000.
Up to 5000	The records will be grouped by intervals of 5000.
Up to 10000	The records will be grouped by intervals of 10000.

Functions for String-typed group by fields

Function	Description
None	All the records that have the same field value will be displayed together as a group.
For 1st letter	The records, of which the field values' first letter is the same, will be grouped together.
For first 2 letters	The records, of which the field values' first two letters are the same, will be grouped together.
For first 3 letters	The records, of which the field values' first three letters are the same, will be grouped together.
For first 4 letters	The records, of which the field values' first four letters are the same, will be grouped together.
For first 5 letters	The records, of which the field values' first five letters are the same, will be grouped together.
For last 1 letter	The records, of which the field values' last letter is the same, will be grouped together.
For last 2 letters	The records, of which the field values' last two letters are the same, will be grouped together.
For last 3 letters	The records, of which the field values' last three letters are the same, will be grouped together.
For last 4 letters	The records, of which the field values' last four letters are the same, will be grouped together.
For last 5 letters	The records, of which the field values' last five letters are the same, will be grouped together.

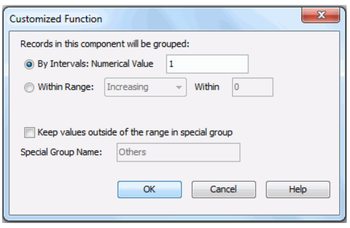
Functions for Date/Time-typed group by fields

Function	Description
None	All the records that have the same field value will be displayed together as a group.
For each second	The records, of which the field values are in the same second, will be grouped together.
For each minute	The records, of which the field values are in the same minute, will be grouped together.
For each hour	The records, of which the field values are in the same hour, will be grouped together.
For each day	The records, of which the field values are in the same day, will be grouped together.
For each week	The records, of which the field values are in the same week, will be grouped together.
For each bi-week	The records, of which the field values are in the same bi-week, will be grouped together.
For each half month	The records, of which the field values are in the same half month, will be grouped together.
For each month	The records, of which the field values are in the same month, will be grouped together.
For each quarter	The records, of which the field values are in the same quarter, will be grouped together.
For each half year	The records, of which the field values are in the same half year, will be grouped together.
For each year	The records, of which the field values are in the same year, will be grouped together.

Customized function

In the special function list for all the above data types, there is an item called Customize. By selecting this item, you can define special functions.

- **Defining special function for Numeric type fields**
  1. Select a Numeric-typed group by field and then click **Customize** in the Special Function drop-down list. The Customized Function dialog appears.

  2. Specify the way in which you want to group the data.

     If you want to group the data by intervals, check the **By Intervals** radio button, then define the interval in the Numerical Value text box.

     If you want to group data by certain range, check the **Within Range** radio button, specify the range in the Within text box and select how to apply the range: apply to increasing data or decreasing data.

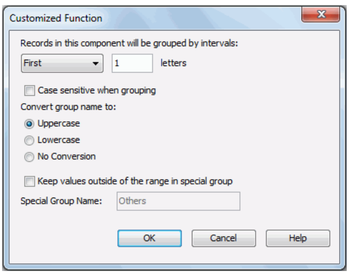
  3. Check **Keep values outside of the range in special group** if you want to put values that do not fall within the specified interval or range in a new group, then give a name for the group in the Special Group Name text box.
  4. Click **OK** to save the settings.

     Note: When grouping Numeric typed data by intervals, JReport follows this rule: all values in each range are >= the minimum value in the range and < the maximum value in the range, and 0 is the offset for the intervals.

     For example, if you specify to group the following values by intervals of 5: -8, -6, -5, -3, -1, 0, 1, 3, 4, 5, 8, the results will be as follows:

Ranges	Values
-10 to -5	-8, -6
-5 to 0	-5, -3, -1
0 to 5	0, 1, 3, 4
5 to 10	5, 8

- **Defining special function for String type fields**
  1. Select a String-typed group by field and then click **Customize** in the Special Function drop-down list to display the Customized Function dialog.

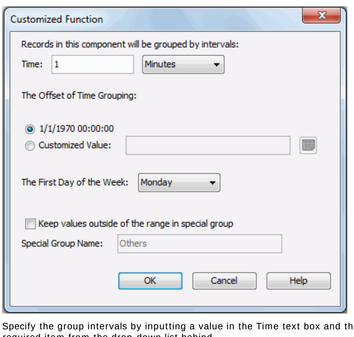
  2. Specify the group intervals as required.

     For example, to group records by the first N letters of the group by field values, select **First** from the drop-down list and then enter **N** in the text box. Note that the number of letters should be an integer no larger than 255.

  3. If you want to distinguish between uppercase and lowercase characters when grouping, check **Case sensitive when grouping**; otherwise, check in which way you want to convert the group names: Uppercase, Lowercase or No Conversion.

     Note: If you check Case sensitive when grouping, No Conversion will be checked as the default converting method.

- Check **Keep values outside of the range in special group** if you want to put values that do not fall within the specified interval in a new group, then give a name for the group in the Special Group Name text box.
- Click **OK** to save the settings.
- **Defining special function for Date/Time type fields**
  1. Select a Date/Time-typed group by field and then click **Customize** in the Special Function drop-down list to display the Customized Function dialog.

  2. Specify the group intervals by inputting a value in the Time text box and then selecting the required item from the drop-down list behind.
  3. Define the offset with which you want to group the data.

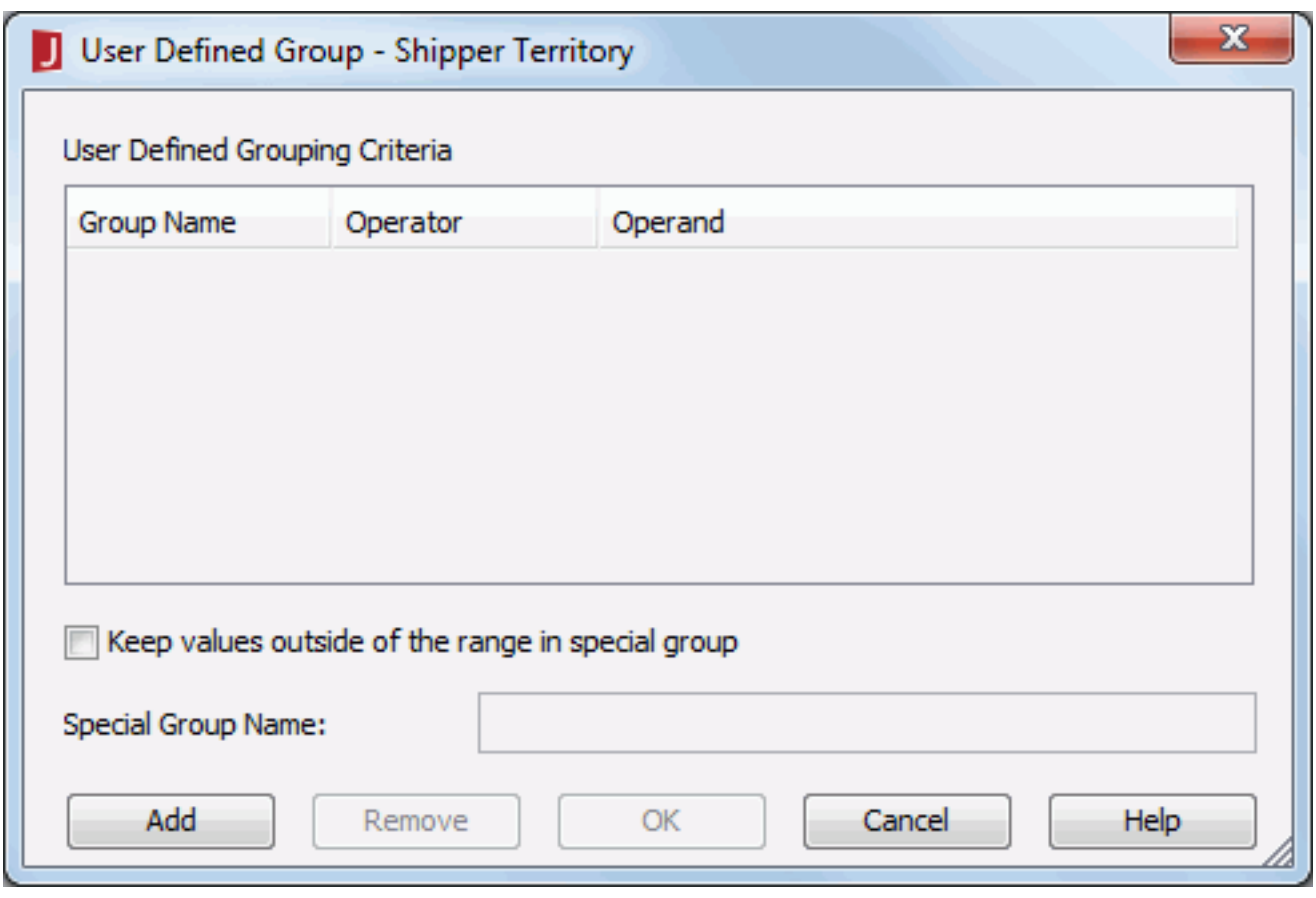
     By default, JReport takes 1/1/1970 00:00:00 as the offset. If you want to define the offset by your own, check the **Customized Value** radio button and then click the calendar button to choose the required offset.

  4. Specify which day will be used as the first day of a week as required.
  5. Check **Keep values outside of the range in special group** if you want to put values that do not fall within the specified interval in a new group, then give a name for the group in the Special Group Name text box.
  6. Click **OK** to save the settings.

Setting the sort manner

After a group has been added, you need to specify in which manner groups at current group level will be sorted.

- **Ascend**
  Groups will be sorted in ascending order.
- **Descend**
  Groups will be sorted in descending order.
- **No Sort**
  Groups will be arranged in their original order.
- **Special Group**
  Selecting this item will bring out the User Defined Group dialog for you to define how to group your information.



For example, if you place a field named Score for grouping which contains student scores that range from 0 to 100, and you want to group the students in 5 ranks, namely rank A: 90~100, B: 80~89, C: 70~79, D: 60~69, and E: 0~59. You can set the following with the User Defined Group dialog.

Group Name	Operator	Operand
A	between	Op1: 90, Op2: 100
B	between	Op1: 80, Op2: 89
C	between	Op1: 70, Op2: 79
D	between	Op2: 60, Op2: 69
E	<=	59

There will be five groups in the order from A to E. If you want to change the order of the groups, you can also do so via the User Defined Group dialog.

- **Custom Sort**
  Selecting this item will bring out the Custom Sort dialog for you to define how to sort the groups. You can specify some fields in the dialog, then the groups will be sorted by the values of the first record in each group on the related fields.

  The following example shows how to sort groups by a specific field.

  Suppose that you have created a table based on the query EmployeeInformation in the catalog file SampleReports.cat as follows: added the fields Hire Date, Name, Home Phone and Salary to be displayed in the table, set the field Employee Position as the group by field and Ascend as the sort order, then applied the Neutral style to the table.

Hire Date	Name	Home Phone	Salary
Marketing			
8/14/1990	Duke Edmonds	(410) 555-4346	49,000.00
9/30/1997	Mary Everett	(202) 555-7033	41,000.00
10/20/1993	Geoffery Schultz	(301) 555-2343	49,000.00
5/19/1989	Jeremy Miner	(410) 555-2834	50,000.00
Sales Representative			
8/7/1996	Maxine Olympus	(703) 555-6173	43,000.00
7/15/1990	Jackie Westray	(202) 555-8245	42,000.00
5/4/1995	Edward Packard	(703) 555-5293	30,000.00
2/7/1991	Laura Fung	(301) 555-2392	47,000.00
11/16/1989	William Penne	(410) 555-9872	55,000.00
7/23/2003	Jimmy Rayner	(703) 555-5923	45,000.00
6/9/1991	Jennifer Lee	(703) 555-4422	35,000.00
4/3/1997	Jonathan Hopkins	(301) 555-2349	42,000.00
9/12/2004	Laurena Croft	(301) 555-7424	50,000.00
Vice President			
5/25/1990	Joseph Compton	(301) 555-9864	60,000.00
10/5/1988	Katherine Chen	(202) 555-2345	65,000.00

For the example, specify the sorting criteria as follows:

1. Right-click the table and select **Table Wizard** from the shortcut menu.
2. In the Group screen of the Table Wizard, select the group by field **Employee Position** in the group by box, then select **Custom Sort** from the Sort column.
3. In the Custom Sort dialog, add the field **Salary** to the sort by box, set the sorting manner as **Ascend**, then click **OK**.
4. Click **Finish** in the Table Wizard to apply the settings.
5. View the table again and you will find that the groups are sorted based on their first record's Salary values.

Hire Date	Name	Home Phone	Salary
Sales Representative			
8/7/1996	Maxine Olympus	(703) 555-6173	43,000.00
7/15/1990	Jackie Westray	(202) 555-8245	42,000.00
5/4/1995	Edward Packard	(703) 555-5293	30,000.00
2/7/1991	Laura Fung	(301) 555-2392	47,000.00
11/16/1989	William Penne	(410) 555-9872	55,000.00
7/23/2003	Jimmy Rayner	(703) 555-5923	45,000.00
6/9/1991	Jennifer Lee	(703) 555-4422	35,000.00
4/3/1997	Jonathan Hopkins	(301) 555-2349	42,000.00
9/12/2004	Laurena Croft	(301) 555-7424	50,000.00
Marketing			
8/14/1990	Duke Edmonds	(410) 555-4346	49,000.00
9/30/1997	Mary Everett	(202) 555-7033	41,000.00
10/20/1993	Geoffery Schultz	(301) 555-2343	49,000.00
5/19/1989	Jeremy Miner	(410) 555-2834	50,000.00
Vice President			
5/25/1990	Joseph Compton	(301) 555-9864	60,000.00
10/5/1988	Katherine Chen	(202) 555-2345	65,000.00

Specifying the Select N condition

Sometimes, you may want to show data of certain range in a group or some groups in a table. To achieve this, you can use the Select N feature. By specifying a Select N condition, you can decide how many records or groups will be displayed in the table. You can also use an Integer-typed parameter to control the value of Select N.

- **To display the top or bottom N records in a table:**
  1. Right-click the table and select **Table Wizard** from the shortcut menu.
  2. In the Group screen of the Table Wizard, select the **Table** node in the group by box, and then click the **Select N** button.
  3. In the Select N dialog, you will see that The whole object is displayed in the In text field. Select an item from the Select N drop-down list.

     By default, All is selected, which means that all records will be displayed.

     If you select Top or Bottom, then you can further specify a number in the combo box below. You can also use an Integer-typed parameter to define the Top/Bottom N condition dynamically. Supposing the number or the value you will specify for the parameter is N, when you view the result, you will find that only the first or last N records in the whole table or in each group (if any group has been defined) are retrieved.
  4. Click **OK** to go back to the Table Wizard.
  5. When done, click **Finish** to apply the settings.
- **To display the top or bottom N groups in certain group level of a table:**
  1. Right-click the table and select **Table Wizard** from the shortcut menu.
  2. In the Group screen of the Table Wizard, select the group by field in the group by box, and then click the **Select N** button.
  3. In the Select N dialog, you will see that the field you have just selected is displayed in the In text field. From the Select N drop-down list, select **Top** or **Bottom**, and then further specify a number or an Integer-typed parameter. If the number or the parameter value is N, then the first or last N groups in that group level will be displayed.
  4. To display all the other groups that do not match the Select N condition (which are by default hidden) in an additional group, check the **Other** box and type a name for the additional group.
  5. Click **OK** to go back to the Table Wizard.
  6. Upon finishing, click **Finish** to apply the settings.

Notes:

- The Select N conditions for the whole object and each group are not mutually exclusive, so you can set them individually.
- If you use an Integer-typed parameter to define the Top/Bottom N condition dynamically at runtime, you should make sure that the parameter has at least one default value that is larger than 0, otherwise you will get exceptions when viewing the report.

Example of applying Select N conditions

The following example shows how to set the Select N conditions, both for the whole table and the groups in the table in detail.

1. Create a table based on the query EmployeeInformation in the catalog file SampleReports.cat as follows: add the fields Name, Hire Date, Employee Position and Notes to be displayed in the table, set the field Assigned Region as the group by field and Descend as the sort order, and apply the Neutral style to the table.

Name	Hire Date	Employee Position	Notes
North America			
Maxine Olympus	8/7/1996	Sales Representative	m.olympus@sales.gourmetjava.com
Joseph Compton	5/25/1990	Vice President	j.compton@gourmetjava.com
Duke Edmonds	8/14/1990	Marketing	d.edmund@marketing.gourmetjava.com
Jackie Westray	7/15/1990	Sales Representative	j.westray@sales.gourmetjava.com
Edward Packard	5/4/1995	Sales Representative	e.packard@sales.gourmetjava.com
Mary Everett	9/30/1997	Marketing	m.everett@marketing.gourmetjava.com
Laura Fung	2/7/1991	Sales Representative	l.fung@gourmetjava.com
William Penne	11/16/1989	Sales Representative	w.penne@gourmetjava.com
Jimmy Rayner	7/23/2003	Sales Representative	j.rayner@sales.gourmetjava.com
Latin America			
Katherine Chen	10/5/1988	Vice President	k.chen@gourmetjava.com
Europe, Middle East, Africa			
Geoffery Schultz	10/20/1993	Marketing	g.schultz@marketing.gourmetjava.com
Jennifer Lee	6/9/1991	Sales Representative	j.lee@gourmetjava.com
Jeremy Miner	5/19/1989	Marketing	j.miner@marketing.gourmetjava.com
Asia-Pacific			
Jonathan Hopkins	4/3/1997	Sales Representative	j.hopkins@sales.gourmetjava.com
Laurena Croft	9/12/2004	Sales Representative	l.croft@sales.gourmetjava.com

2. Right-click the table and select **Table Wizard** from the shortcut menu.
3. In the Group screen of the Table Wizard, select the **Table** node in the group by box, and then click the **Select N** button.
4. In the Select N dialog, select **Top** from the Select N drop-down list and enter **2** in the combo box.
5. Click **OK** to go back to the Table Wizard.
6. Select the group by field **Assigned Region** in the group by box and click the **Select N** button again.
7. In the Select N dialog, select **Top**, enter **2**, check the **Other** box and enter **Others** in the text field.
8. Click **OK** to close the Select N dialog, then click **Finish** in the Table Wizard to accept all settings.
9. View the table again.

Name	Hire Date	Employee Position	Notes
North America			
Maxine Olympus	8/7/1996	Sales Representative	m.olympus@sales.gourmetjava.com
Joseph Compton	5/25/1990	Vice President	j.compton@gourmetjava.com
Latin America			
Katherine Chen	10/5/1988	Vice President	k.chen@gourmetjava.com
Others			
Geoffery Schultz	10/20/1993	Marketing	g.schultz@marketing.gourmetjava.com
Jennifer Lee	6/9/1991	Sales Representative	j.lee@gourmetjava.com
Jonathan Hopkins	4/3/1997	Sales Representative	j.hopkins@sales.gourmetjava.com
Laurena Croft	9/12/2004	Sales Representative	l.croft@sales.gourmetjava.com

You can find that the first 2 records in each group (the group Latin America has only one record) have been retrieved, and then the first 2 groups (North America, and Latin America) are displayed, with the records in the rest group combined into an additional Others group.

Filtering the groups

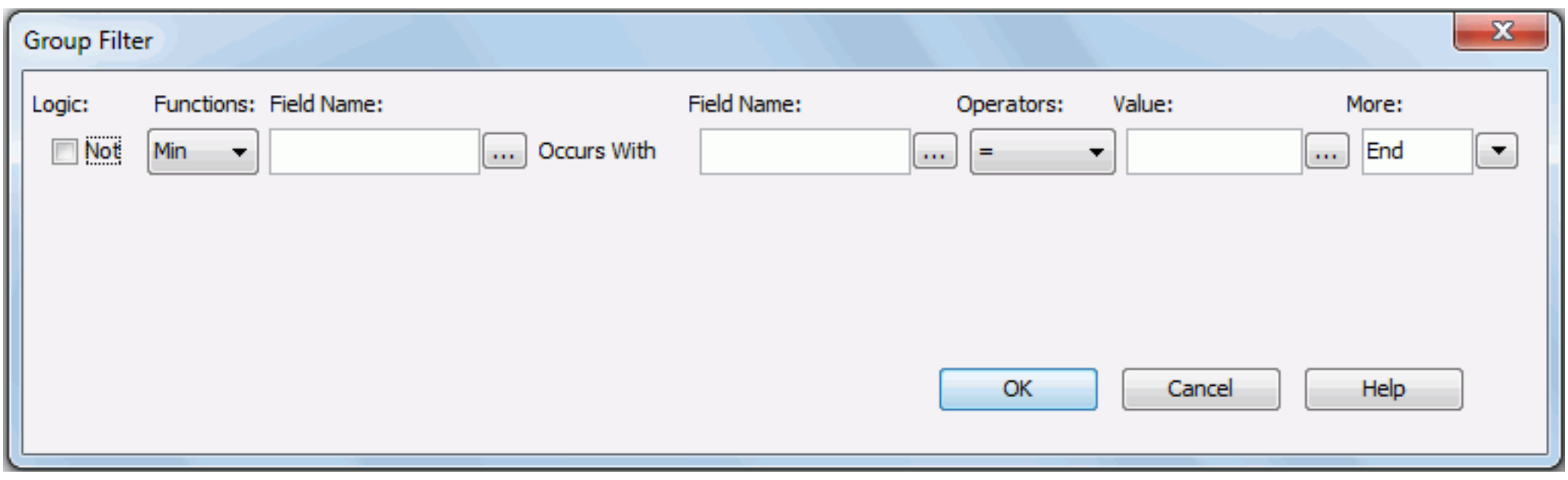
As the name suggests, the Group Filter feature is based on groups. If a group satisfies the filter condition, all the records in this group will be shown. If the group doesn't satisfy the condition, then the whole group will not be displayed in the table.

Suppose that you have created a table based on the query EmployeeInformation in the catalog file SampleReports.cat as follows: added the fields Hire Date, Name, Home Phone and Salary to be displayed in the table, set the field Employee Position as the group by field and applied the Neutral style to the table. Since no group filter condition has been set, there are three groups in the table: Marketing, Sales Representative and Vice President.

Hire Date	Name	Home Phone	Salary
Marketing			
8/14/1990	Duke Edmonds	(410) 555-4346	49,000.00
9/30/1997	Mary Everett	(202) 555-7033	41,000.00
10/20/1993	Geoffery Schultz	(301) 555-2343	49,000.00
5/19/1989	Jeremy Miner	(410) 555-2834	50,000.00
Sales Representative			
8/7/1996	Maxine Olympus	(703) 555-6173	43,000.00
7/15/1990	Jackie Westray	(202) 555-8245	42,000.00
5/4/1995	Edward Packard	(703) 555-5293	30,000.00
2/7/1991	Laura Fung	(301) 555-2392	47,000.00
11/16/1989	William Penne	(410) 555-9872	55,000.00
7/23/2003	Jimmy Rayner	(703) 555-5923	45,000.00
6/9/1991	Jennifer Lee	(703) 555-4422	35,000.00
4/3/1997	Jonathan Hopkins	(301) 555-2349	42,000.00
9/12/2004	Laurena Croft	(301) 555-7424	50,000.00
Vice President			
5/25/1990	Joseph Compton	(301) 555-9864	60,000.00
10/5/1988	Katherine Chen	(202) 555-2345	65,000.00

Example 1: Show groups where the lowest employee salary is less than 50000

1. Right-click the table and select **Table Wizard** from the shortcut menu.
2. In the Group screen of the Table Wizard, select the group by field **Employee Position**, the Group Filter button will then become available. Click it to display the Group Filter dialog.



3. Fill in the following information in the dialog:

Logic	Function	Field Name (before Occurs With)	Field Name (after Occurs With)	Operator	Value	More
Unchecked	Min	Salary	Salary	<	50000	End

4. Click **OK** to close the dialog and then **Finish** in the Table Wizard to apply the settings.
5. View the report result. You will find that only the two groups Marketing and Sales Representative are displayed, while the group Vice President that does not fit this condition is ignored.

Hire Date	Name	Home Phone	Salary
Marketing			
8/14/1990	Duke Edmonds	(410) 555-4346	49,000.00
9/30/1997	Mary Everett	(202) 555-7033	41,000.00
10/20/1993	Geoffery Schultz	(301) 555-2343	49,000.00
5/19/1989	Jeremy Miner	(410) 555-2834	50,000.00
Sales Representative			
8/7/1996	Maxine Olympus	(703) 555-6173	43,000.00
7/15/1990	Jackie Westray	(202) 555-8245	42,000.00
5/4/1995	Edward Packard	(703) 555-5293	30,000.00
2/7/1991	Laura Fung	(301) 555-2392	47,000.00
11/16/1989	William Penne	(410) 555-9872	55,000.00
7/23/2003	Jimmy Rayner	(703) 555-5923	45,000.00
6/9/1991	Jennifer Lee	(703) 555-4422	35,000.00
4/3/1997	Jonathan Hopkins	(301) 555-2349	42,000.00
9/12/2004	Laurena Croft	(301) 555-7424	50,000.00

Example 2: Show groups in which the highest-salary employee was hired before Jan. 1, 1989

In this example, you need to set the Group Filter dialog as follows:

Logic	Function	Field Name (before Occurs With)	Field Name (after Occurs With)	Operator	Value	More
Unchecked	Max	Salary	Hire Date	<	1989-01-01 00:00:00.000	End

When you view the report result, you will find that this time only the group Vice President is displayed.

Hire Date	Name	Home Phone	Salary
Vice President			
5/25/1990	Joseph Compton	(301) 555-9864	60,000.00
10/5/1988	Katherine Chen	(202) 555-2345	65,000.00

Example 3: Show groups where at least one "not highest" employee salary is larger than 50000

In this example, you need to set the Group Filter dialog as follows:

Logic	Function	Field Name (before Occurs With)	Field Name (after Occurs With)	Operator	Value	More
Checked	Max	Salary	Salary	>	50000	End

When you view the report result, you will find that this time only the group Vice President is displayed.

Hire Date	Name	Home Phone	Salary
Vice President			
5/25/1990	Joseph Compton	(301) 555-9864	60,000.00
10/5/1988	Katherine Chen	(202) 555-2345	65,000.00

Notes:

- In the Value field of the Group Filter dialog, you can input a constant or parameter.
- For String type constants, you can type the string without quotation marks. For Date/Time type constants, you have to follow the Date/Time format specified in the Date Format tab of the Get JDBC Connection Information dialog.
- Ensure that the report result contains at least one record; otherwise JReport Designer will throw out an error message.

Grouping data dynamically

You can make a multi-level group report by selecting a field as grouping criterion for the report. However, because the grouping criteria are definite, if you want to group several times according to different grouping criteria but based on the same table in the database it is cumbersome. For example, you want to make three employee list reports with different grouping criteria, the first one is grouped by their first name, the second one is by the hire date and the third one is by their salary, then you have to repeat the steps of setting query, selecting fields as grouping criteria, and so on, which is not efficient.

For this case you can use the Dynamic Grouping feature of JReport Designer, which means grouping criteria is a dynamic process. You don't need to repeat the same steps to make multiple reports with different grouping criteria. You can just predefine a parameter using String value type and add it to the group list. Then when you run the report, the parameter dialog prompts the end user to select a field to group by. All the acceptable group by fields are listed in a drop-down list. You can select any of them as grouping criterion.

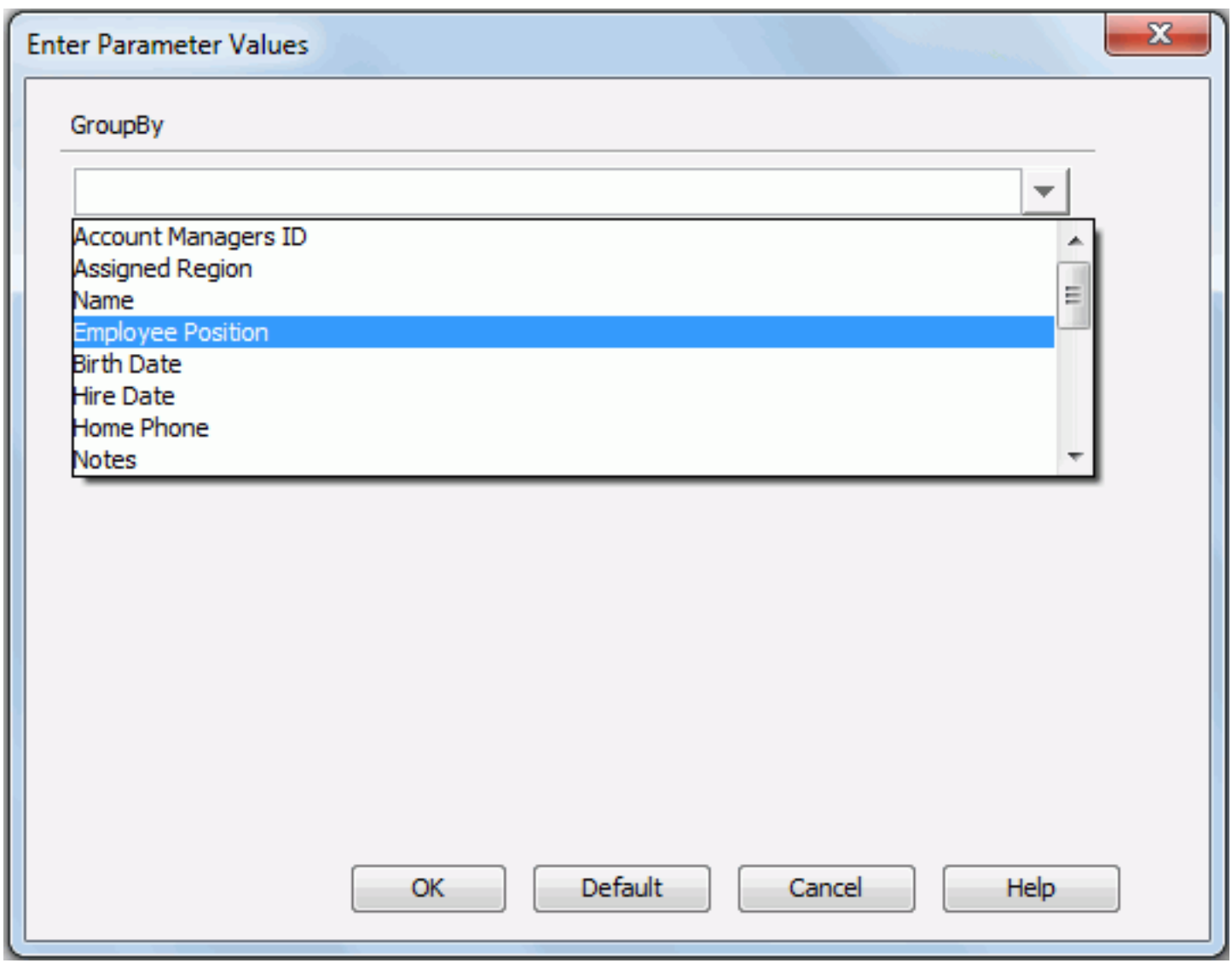
Example of using dynamic grouping

Assume that you have created a table based on the EmployeeInformation query in the SampleReports.cat catalog, which displays the fields Name, Home Phone, Employee Position and Salary, and applies the Neutral style.

Name	Home Phone	Employee Position	Salary
Laurena Croft	(301) 555-7424	Sales Representativ	50,000.00
Jonathan Hopkins	(301) 555-2349	Sales Representativ	42,000.00
Jeremy Miner	(410) 555-2834	Marketing	50,000.00
Jennifer Lee	(703) 555-4422	Sales Representativ	35,000.00
Geoffery Schultz	(301) 555-2343	Marketing	49,000.00
Katherine Chen	(202) 555-2345	Vice President	65,000.00
Jimmy Rayner	(703) 555-5923	Sales Representativ	45,000.00
William Penne	(410) 555-9872	Sales Representativ	55,000.00
Laura Fung	(301) 555-2392	Sales Representativ	47,000.00
Mary Everett	(202) 555-7033	Marketing	41,000.00
Edward Packard	(703) 555-5293	Sales Representativ	30,000.00
Jackie Westray	(202) 555-8245	Sales Representativ	42,000.00
Duke Edmonds	(410) 555-4346	Marketing	49,000.00
Joseph Compton	(301) 555-9864	Vice President	60,000.00
Maxine Olympus	(703) 555-6173	Sales Representativ	43,000.00

To apply the dynamic grouping conditions, follow the steps below:

1. Right-click the table and select **Table Wizard** from the shortcut menu to display the Table Wizard.
2. In the Resources box of the Group screen, click **<Add Parameter...>** in the Parameters node.
3. In the Create Parameter dialog, create a type-in parameter named GroupBy of String type (leave the other settings to their default), then click **OK**.
4. Specify to group the report on the just created parameter **GroupBy** and use **Ascend** as the sorting order.
5. Click **Finish** in the Table Wizard to accept the settings.
6. Click the **View** tab to view the report. You will then be prompted to specify the parameter value. The parameter value drop-down list will show all available fields (including the DBFields in the query EmployeeInformation and formulas which are related to these DBFields).



7. Select **Employee Position** from the value drop-down list and data in the table is then grouped by positions of the employees.

GroupBy	Name	Home Phone	Employee Position	Salary
Marketing				
	Duke Edmonds	(410) 555-4346	Marketing	49,000.00
	Mary Everett	(202) 555-7033	Marketing	41,000.00
	Geoffery Schultz	(301) 555-2343	Marketing	49,000.00
	Jeremy Miner	(410) 555-2834	Marketing	50,000.00
Sales Representative				
	Maxine Olympus	(703) 555-6173	Sales Representativ	43,000.00
	Jackie Westray	(202) 555-8245	Sales Representativ	42,000.00
	Edward Packard	(703) 555-5293	Sales Representativ	30,000.00
	Laura Fung	(301) 555-2392	Sales Representativ	47,000.00
	William Penne	(410) 555-9872	Sales Representativ	55,000.00
	Jimmy Rayner	(703) 555-5923	Sales Representativ	45,000.00
	Jennifer Lee	(703) 555-4422	Sales Representativ	35,000.00
	Jonathan Hopkins	(301) 555-2349	Sales Representativ	42,000.00
	Laurena Croft	(301) 555-7424	Sales Representativ	50,000.00
Vice President				
	Joseph Compton	(301) 555-9864	Vice President	60,000.00
	Katherine Chen	(202) 555-2345	Vice President	65,000.00

To group the data by hire date or salary, the end user would need to select Hire Date or Salary from the drop-down list.

Sorting the data

By default, the records in a table are displayed randomly; they are displayed in the order they are returned from the fetch operation. You can specify that JReport sort the records in a table, and also within the groups in table, if any.

The following example shows how to set the sorting criteria in a table.

Suppose that you have created a table based on the query EmployeeInformation in the catalog file SampleReports.cat as follows: added the fields Hire Date, Name, Home Phone and Salary to be displayed in the table, set the field Employee Position as the group by field and Ascend as the sort order, then applied the Neutral style to the table.

Hire Date	Name	Home Phone	Salary
Marketing			
8/14/1990	Duke Edmonds	(410) 555-4346	49,000.00
9/30/1997	Mary Everett	(202) 555-7033	41,000.00
10/20/1993	Geoffery Schultz	(301) 555-2343	49,000.00
5/19/1989	Jeremy Miner	(410) 555-2834	50,000.00
Sales Representative			
8/7/1996	Maxine Olympus	(703) 555-6173	43,000.00
7/15/1990	Jackie Westray	(202) 555-8245	42,000.00
5/4/1995	Edward Packard	(703) 555-5293	30,000.00
2/7/1991	Laura Fung	(301) 555-2392	47,000.00
11/16/1989	William Penne	(410) 555-9872	55,000.00
7/23/2003	Jimmy Rayner	(703) 555-5923	45,000.00
6/9/1991	Jennifer Lee	(703) 555-4422	35,000.00
4/3/1997	Jonathan Hopkins	(301) 555-2349	42,000.00
9/12/2004	Laurena Croft	(301) 555-7424	50,000.00
Vice President			
5/25/1990	Joseph Compton	(301) 555-9864	60,000.00
10/5/1988	Katherine Chen	(202) 555-2345	65,000.00

For the example, specify the sorting criteria as follows:

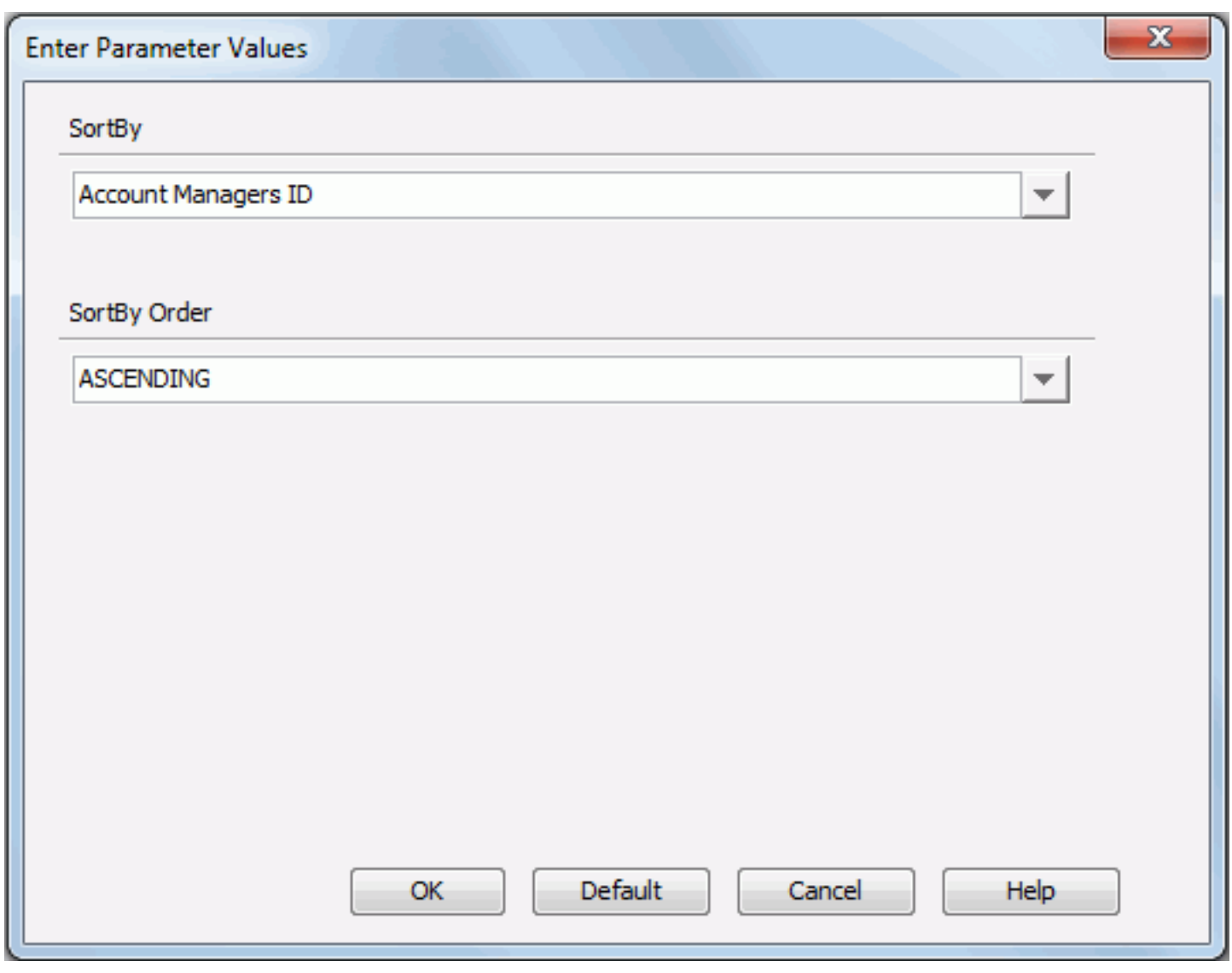
1. Select the table and do one of the following:
   - Click **Insert** > **Group** on the menu bar.
   - Click the **Insert Group** button on the Insert toolbar.
   - Right-click the table and select **Table Wizard** from the shortcut menu.
2. In the Table Wizard, click the **Display** screen, then click the **Sort Fields By** button.
3. In the Sort Fields By dialog, select the field **Hire Date** as the sort by field from the Resources box and click to add it to the sort by box, select **Ascend** from the Sort column, then click **OK** to accept the changes.
4. Click **Finish** in the Table Wizard to apply the settings.
5. View the table again and you will find that the records in each group are displayed in ascending order by Hire Date.

Hire Date	Name	Home Phone	Salary
Marketing			
5/19/1989	Jeremy Miner	(410) 555-2834	50,000.00
8/14/1990	Duke Edmonds	(410) 555-4346	49,000.00
10/20/1993	Geoffery Schultz	(301) 555-2343	49,000.00
9/30/1997	Mary Everett	(202) 555-7033	41,000.00
Sales Representative			
11/16/1989	William Penne	(410) 555-9872	55,000.00
7/15/1990	Jackie Westray	(202) 555-8245	42,000.00
2/7/1991	Laura Fung	(301) 555-2392	47,000.00
6/9/1991	Jennifer Lee	(703) 555-4422	35,000.00
5/4/1995	Edward Packard	(703) 555-5293	30,000.00
8/7/1996	Maxine Olympus	(703) 555-6173	43,000.00
4/3/1997	Jonathan Hopkins	(301) 555-2349	42,000.00
7/23/2003	Jimmy Rayner	(703) 555-5923	45,000.00
9/12/2004	Laurena Croft	(301) 555-7424	50,000.00
Vice President			
10/5/1988	Katherine Chen	(202) 555-2345	65,000.00
5/25/1990	Joseph Compton	(301) 555-9864	60,000.00

Sorting data dynamically

Similar to dynamic grouping, sorting can also be defined at runtime. Using the table in the above example, you can apply a dynamic sort on it as follows:

1. Right-click the table and select **Table Wizard** from the shortcut menu.
2. In the Table Wizard, click the **Display** screen, then click the **Sort Fields By** button.
3. In the Sort Fields By dialog, remove the sort by field **Hire Date** set in the above example.
4. In the Resources box, click **<Add Parameter...>** in the Parameters node.
5. In the Create Parameter dialog, create a type-in parameter named SortBy of String type (leave the other settings to their default), then click **OK**.
6. Add the just created parameter **SortBy** as the sort by field.
7. Click **Dynamic Sort** in the sort order drop-down list of the Sort column.
8. In the Specify Sort Order for dialog, input **SortBy Order**, then click **OK** to close the dialog.
9. Click **OK** in the Sort Fields By dialog to accept the changes.
10. Click **Finish** in the Table Wizard to apply the settings.
11. View the report result. In the Enter Parameter Values dialog, select a field from the SortBy drop-down list and specify the sorting manner in the SortBy Order drop-down list.



Here, we choose **Salary** from the SortBy list and **DESCENDING** from the SortBy Order list. The records within each group are displayed in descending order according to their salary values.

Hire Date	Name	Home Phone	Salary
Marketing			
5/19/1989	Jeremy Miner	(410) 555-2834	50,000.00
8/14/1990	Duke Edmonds	(410) 555-4346	49,000.00
10/20/1993	Geoffery Schultz	(301) 555-2343	49,000.00
9/30/1997	Mary Everett	(202) 555-7033	41,000.00
Sales Representative			
11/16/1989	William Penne	(410) 555-9872	55,000.00
9/12/2004	Laurena Croft	(301) 555-7424	50,000.00
2/7/1991	Laura Fung	(301) 555-2392	47,000.00
7/23/2003	Jimmy Rayner	(703) 555-5923	45,000.00
8/7/1996	Maxine Olympus	(703) 555-6173	43,000.00
7/15/1990	Jackie Westray	(202) 555-8245	42,000.00
4/3/1997	Jonathan Hopkins	(301) 555-2349	42,000.00
6/9/1991	Jennifer Lee	(703) 555-4422	35,000.00
5/4/1995	Edward Packard	(703) 555-5293	30,000.00
Vice President			
10/5/1988	Katherine Chen	(202) 555-2345	65,000.00
5/25/1990	Joseph Compton	(301) 555-9864	60,000.00

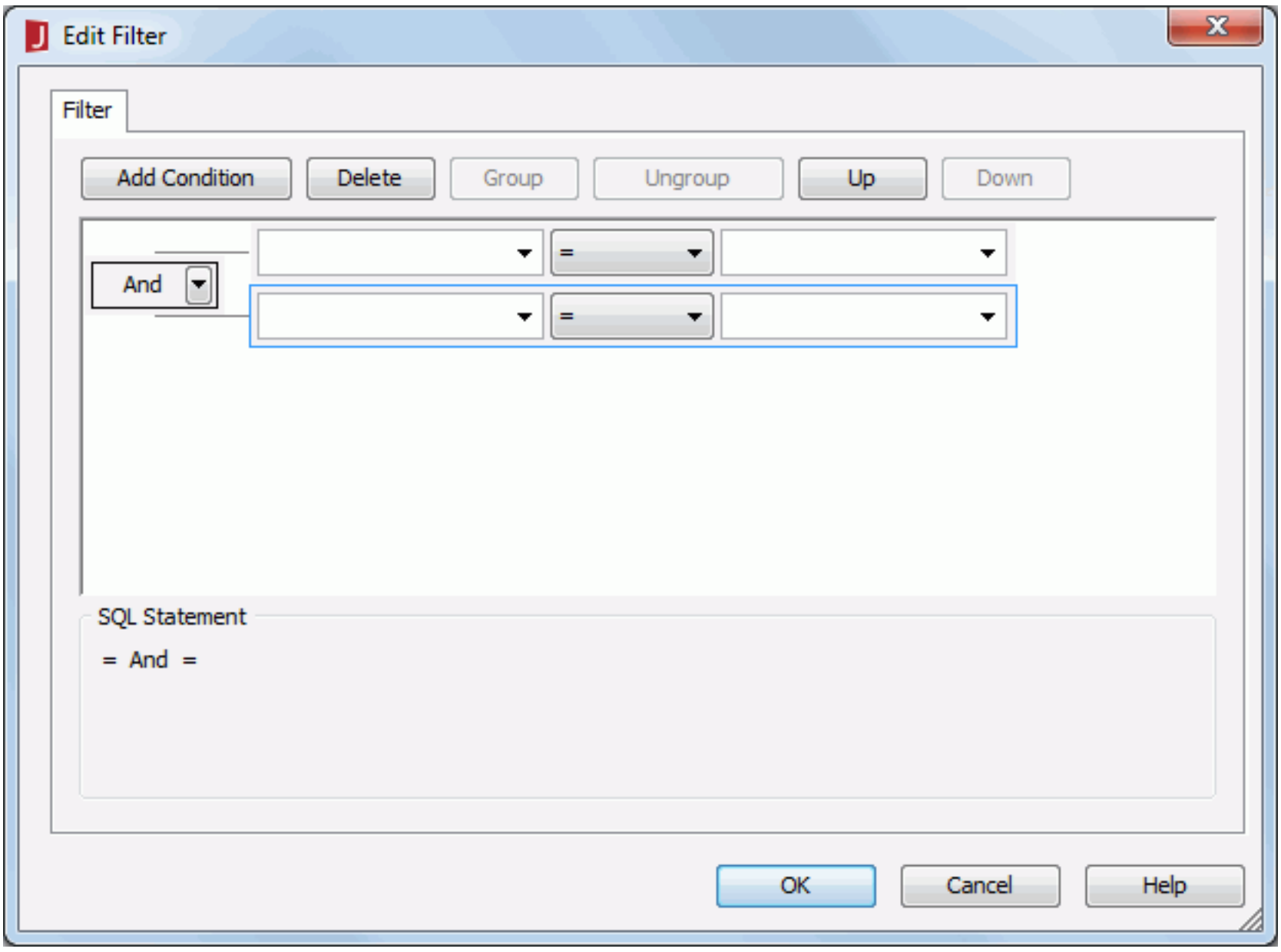
Note: The following SQL type of data cannot be sorted: Db.SQL_BINARY, Db.SQL_BLOB, Db.SQL_CLOB, Db.SQL_LONGVARCHAR, Db.SQL_LONGVARBINARY, Db.SQL_VARBINARY and Db.SQL_OTHER.

Filtering the data

You can apply some filters to a table so as to reduce the records displayed in the table.

To add filter conditions to a table:

1. Right-click the table and select **Format Filter** to display the Edit Filter dialog.



2. If you are working with a web report or a library component, the Filter drop-down is available, listing all the predefined filters of the business view on which the table is created. You can select one from the drop-down list to apply, or select **User Defined** in the list to define a new filter as required.
3. Click the **Add Condition** button to add a condition line.
4. In the field text field, specify the field on which the filter will be based.
5. From the operator drop-down list, set the operator with which to compose the filter expression.
6. In the value text field, specify the value of how to filter the field.

   If the table is in a web report or library component, you can use a parameter to specify the value dynamically and when the available parameters cannot meet your requirement, you can also create a local parameter to use in the filter. For more information about using parameters to dynamically filter data, see Dynamically filtering queries.
7. To add another line of filter expression, click the **Add Condition** button and define the expression as required. Then from the logic drop-down list, specify the relationship between the two filter expressions.
8. Repeat the above steps to add more filter expressions if necessary.

   To make some conditions grouped, select them and click the **Group** button, then the selected conditions will be added in one group and work as one line of filter expression. Conditions and groups together can be further grouped. To take any condition or group in a group out, select it and click **Ungroup.**

   To adjust the priority of the conditions, select it and click the **Up** or **Down** button.

   To delete a condition line, select it and click the **Delete** button.
9. When done, click **OK** to create the filter.

   Then when you preview the table, only data satisfying the specified filter conditions are shown.

See also the Edit Filter dialog for page report or web report/library component for details about options in the dialog.

Note: The following SQL type of data cannot be filtered: Db.SQL_BINARY, Db.SQL_BLOB, Db.SQL_CLOB, Db.SQL_LONGVARCHAR, Db.SQL_LONGVARBINARY, Db.SQL_VARBINARY and Db.SQL_OTHER.

Managing columns/rows in a table

Cells in a table are organized into columns and rows. You can resize, insert, and delete columns/rows, convert between group and detail columns, and specify to show or hide certain columns in a table.

Note that some operations on table rows and columns are achieved by first selecting a table cell, and you may find that sometimes, a cell is entirely occupied by a field or label, and it is difficult to select the cell, in which cases, you need to first resize the field or label or the cell and then click the blank in the cell so as to select it.

Resizing a column/row

You can take one of the following ways to resize columns/rows of a table:

- To resize a column, drag the boundary on the right side of the column to the required width. If the boundary is not the rightmost one, the column width will change, but the total width of the table will not change. If you want to change both the column width and the table width, press the **Shift** button on the keyboard while dragging. To resize a row, drag the boundary below the row to the required height. Both the row height and the table height will change. If you want to change just the row height, press the **Shift** button on the keyboard while dragging the boundary, which is not the last boundary.
- Select a table cell, right-click it and then select **Row Height** or **Column Width** from the shortcut menu to access the corresponding dialog. Type a value in the dialog, click **OK**, and the height/width of the row/column in which the cell is located will then be changed according to your specification.
- Select the column/row, right-click it and select **Column Width**/**Row Height** from the shortcut menu. In the Column Width/Row Height dialog, type a value in the text box and click **OK**. The width/height of the column/row will then be changed according to your specification.

Inserting a row

1. Select a cell or a row in the table, right-click it and then click **Insert** on the shortcut menu.
2. In the Insert dialog, specify where the row will be inserted, above or below the selected cell.
3. Click **OK**. A new row will then be inserted into the table according to your selection.

Inserting a column

A table can have the following types of column: common column, summary column, detail column and group column. You can insert any type of column into a table as you want.

To insert a common column:

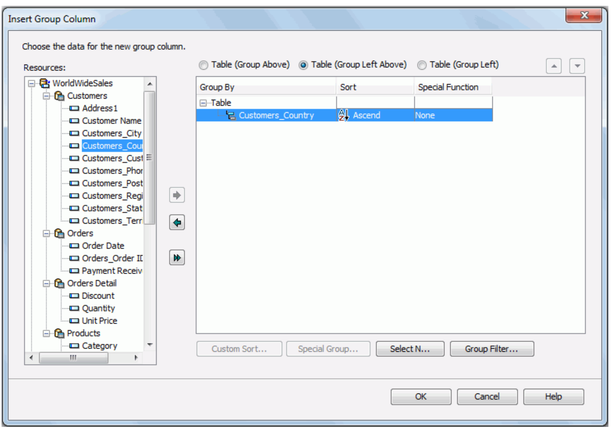
- Select a cell in the table, right-click it and click **Insert** on the shortcut menu. In the Insert dialog, specify where the column will be inserted, before or after the selected cell, then click **OK**. A new common column will then be inserted into the table in the position you specify.
- Select a column in the table, right-click it, then on the shortcut menu, click **Insert** > **Common Column**. A new common column will then be inserted before the selected column.
- Select the table, right-click it, then on the shortcut menu, click **Insert** > **Common Column**. A new common column will then be inserted as the last column in the table.

To insert a detail/summary column:

1. Select the table or a column in the table, right-click it, then click **Insert** > **Detail Column**/**Summary Column** on the shortcut menu.
2. In the corresponding insert column dialog, specify the resource you want to use for the new column.
3. Click **OK**. A new detail/summary column will then be inserted. However the position to which the column will be inserted differs according to where the shortcut menu is accessed. If you use the column shortcut menu to insert the column, the new column will be placed before the selected column; if you use the table shortcut menu to insert the column, it will be placed after the last detail/summary column, or as the last column in the table when there is no detail/summary column.

To insert a group column:

1. Select the table or a column in the table, right-click it, then click **Insert** > **Group Column** on the shortcut menu. The Insert Group Column dialog appears.



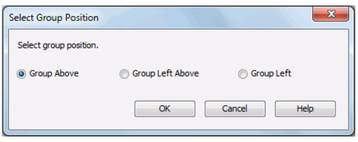
2. In the Resources box, select the field you want to use for the new group column and click to add it as the group by field, then specify the sorting direction of the newly added group in the Sort column, and if the type of the added field is Numeric/String/Date/Time, select a special function for the field from the Special Function column as required (for details, see group the data).
3. Specify the position of the group by field: Group Above, Group Left Above, or Group Left.
4. Repeat the above two steps to add more group columns if required.
5. Click **OK** to insert the group columns.

   The next time when you open the Insert Group Column dialog to add more group columns, all the added group by fields will be listed in the dialog. You can choose to remove or edit them if required.

Converting a column

You can convert a group column into a detail column, and vice versa.

- To convert a group column into a detail column, select the group column, right-click it, and select **Convert to Detail** from the shortcut menu, then the conversion is done.
- **To convert a detail column into a group column:**
  1. Select the detail column you want to convert, right-click it and select **Convert to Group** from the shortcut menu. The Select Group Position dialog appears.

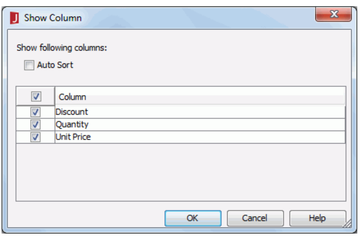


  2. Specify the position for the newly converted group by field.
     - **Group Above**
       Specifies to place the group by field in its own row above the detail columns.
     - **Group Left Above**
       Specifies to place the group by field in its own row and column above and left of the detail columns.
     - **Group Left**
       Specifies to place the group by field in its own column left of the detail columns.
  3. Click **OK** to save the changes.

     Note: A detail column cannot be converted to a group column in the following two conditions:
     - The field bound with the detail column is also added as group by field at the same time.
     - The field bound with the detail column is unable to be used as group by field.

Showing certain columns

1. Right-click the table and select **Show Column** from the shortcut menu. The Show Column dialog appears.



2. All the columns in the table are listed, with text in cells of the table header row representing corresponding columns. By default, the columns are listed according to their order in the table. Check **Auto Sort** if you want them to be listed alphabetically.
3. Check the checkboxes ahead of the columns you want to show.
4. Click **OK** to accept the changes.

Hiding a column/row

To hide a table row, select the row, right-click it, and select **Hide** from the shortcut menu.

To hide a table column, you can use any of the following methods:

- Select the column to be hidden, right-click it, and select **Hide Column** from the shortcut menu.
- Select the table, right-click it and select **Show Column** from the shortcut menu. In the Show Column dialog, uncheck the column to be hidden.
- Select the cell in the table header, which is in the column you want to hide, right-click the cell and select **Hide Column** from the shortcut menu. If the table has more than one header row, use the header in the first row to access the Hide Column command.

Deleting a column/row

1. Select a cell which is in the column/row to be deleted, right-click it, and then select **Delete** from the shortcut menu.
2. In the Delete dialog, select **Column** or **Row**.
3. Click **OK**. The column or row will then be deleted from the table.

Besides, you can also select the specific column or row to be deleted, right-click it and select **Delete** from the shortcut menu to delete it.

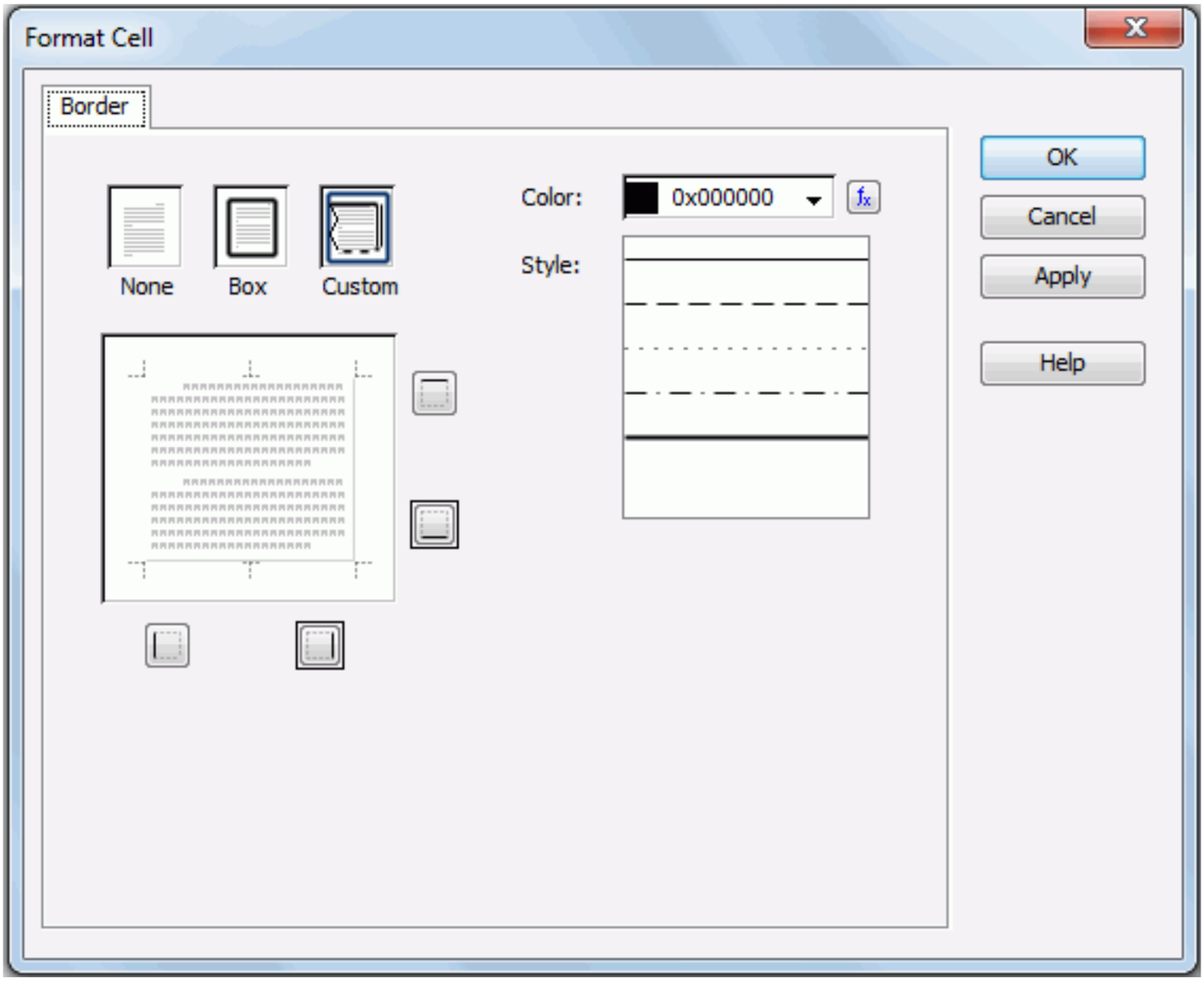
Note: There must be at least one table detail row (TD) in a table. Therefore, if a row is the only detail row of a table, it cannot be deleted.

Formatting cells in a table

Cells in a table can be easily formatted.

- To merge cells in the group header/group footer/table footer panel, select adjacent cells, right-click and select **Merge** (or click the **Merge** button on the Standard toolbar).
- To unmerge cells, right-click it and select **Unmerge** from the shortcut menu. The cell will then be split into multiple cells.

  Unmerging is the reverse operation to merging, and therefore only previously merged cells can be unmerged.
- To format the borders of a table cell, take the following steps:
  1. Right-click the cell and select **Format Border** from the shortcut menu. The Format Cell dialog appears.



  2. Specify whether the borders will be displayed or not by checking None, Box or Custom. If Custom is checked, you can click , , , or to set the visible/invisible status of the left, right, top, or bottom border.
  3. Specify the border color from the Color drop-down list. If you want to use a formula to control the color, click and select a formula from the drop-down list. If the given formulas cannot meet your requirements, click **<Create Formula>** to create one.
  4. Set the border style in the Style list box.
  5. When done, click **OK** to accept the changes.

Setting up a data container link

A data container link is used to set up a relationship between the parent and child data containers. It includes:

- A set of link conditions which are used to dynamically filter out data of the child data container.
- A set of parameter links (if the child data container has parameters) which are used to assign some fixed values to the child container parameters so that at runtime, you do not need to enter values for them.

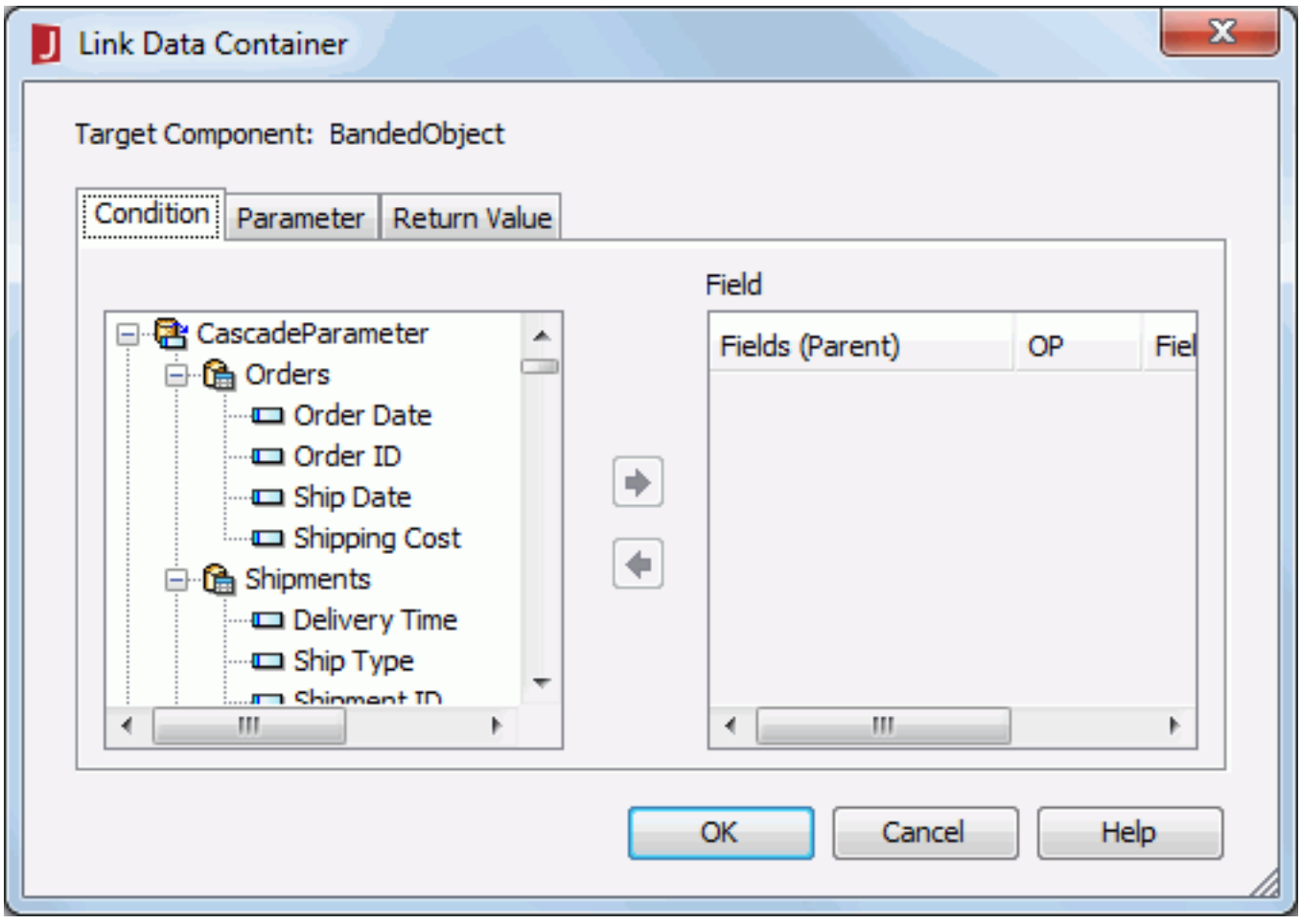
Assume that you have created a standard banded report showing the customer information. Then, you insert a table into the detail panel of the banded object, which displays the order information. Now you want to set up some link conditions between these two containers, so that when you view the report, a table will be generated according to the ID of the customer as follows:

Customer ID	Country	Phone
1	USA	(212) 555-3462

Customer ID	Product Name	Discount	Quantity
1	Jamaica Blue Mountain	.00	20
1	Almond Tosted	.00	15
1	Chocolate Hazelnut	.00	15
1	Hazelnut Crosse	.00	20
1	Irish Crosse	.00	25
1	Jamaica Blue Mountain	.00	20
1	Torrefazione Italia	.00	20
1	Macadamia Nut	.00	40
1	Almond Tosted	.00	15
1	Jamaica Blue Mountain	.00	20
1	Torrefazione Italia	.00	20
1	Macadamia Nut	.00	40
1	Almond Tosted	.00	15
1	USA	(212) 555-3462	

Customer ID	Product Name	Discount	Quantity
1	Jamaica Blue Mountain	.00	20
1	Almond Tosted	.00	15
1	Chocolate Hazelnut	.00	15

To achieve this, select the table and right-click it. On the shortcut menu, click **Data Container Link**, then specify the settings as required in the Link Data Container dialog.



Generally, data container link functions the same as subreport links. The only difference is that you do not need to specify a specific component that is linked with the parent data container, since you have selected the child as the target of the links. For details about how to set up data container link, refer to Subreports.

Banded objects

A banded object is a kind of component that can present grouped data and detailed data. A banded object is composed of several banded panels with which you can easily organize data fields and other elements.

In JReport, banded objects are data containers, which means that a banded object displays the results of a query and you can apply some filters to narrow down the records displayed in it. Also, when a banded object is inserted into another data container, you can set up data container link between the banded object and its parent.

The following topics describe banded objects:

- Inserting a banded object
- Modifying a banded object
- Managing data of a banded object
- Managing banded panels

See an example: The SampleComponents catalog, included with JReport Designer, contains reports that have examples of how each component type could be used in a report. For the banded object example, open `<install_root>\Demo\Reports\SampleComponents\BandedObjectReport.cls`.

Inserting a banded object

To insert a banded object into a report, perform either of the following:

- **Using the report wizard**

  1. Position the mouse pointer at the destination where you want to insert the banded object. A banded object can be inserted in the report areas listed in Component placement.

  2. Do one of the following to open the Create Banded Object wizard:

     - Click **Insert** > **Banded Object** on the menu bar.

     - Click the **Insert Banded Object** button on the Insert toolbar.

  3. In the Data screen of the wizard, define or select the dataset on which the banded object will be created.

     - **New**
       If checked, the banded object will be built on a new dataset created from the catalog resources.

     - **Existing**
       If checked, specify the dataset in the current report from which the banded object will be created.

     - **Current**
       If checked, the banded object will be built on the dataset which is used by its parent.

  4. In the Display screen, add the required fields from the Resources box to be displayed in the banded object, then modify the display name of any added field if necessary by double-clicking its Display Name column.

     If you want to have data in the banded object sorted in a desired order, click the **Sort Fields By** button, then specify the fields on which the sorting will be based and the sorting manner as required (for details, see the same topic Sorting the data on table).

  5. In the Group screen, add the fields as the grouping criteria, and then specify the sorting direction of each group level in the Sort column (for details, see the same topic Grouping the data on table).

  6. To add summaries, go to the Summary screen. In the sum on box (the box on the right), specify the group to which the summary will be applied (if you select Banded Object, the summary will be based on the whole dataset), add a field from the Resources box as the sum on field, and then select the function for the summary from the Aggregate Function column.

  7. In the Filter screen, define some filter conditions to filter the data displayed in the banded object.

     To define a filter condition, click the **Add Condition** button to add a condition line, then from the field drop-down list, select the field on which the filter will be based, set the operator with which to compose the filter expression from the operator drop-down list, and in the value combo box, type the value of how to filter the field or select the value from the drop-down list.

  8. In the Style screen, specify the style of the banded object.

     If you have specified to insert the banded object into another banded object, by default the banded object will inherit its parent's style. If you want to apply another style to the banded object, uncheck the **Inherit Style** option and then select the required style from the Style box. For more information, see Inheriting the style from a data container.

  9. Click **Finish** to insert the banded object.

     If you have specified to insert the banded object in the report body or tabular cell, the banded object will be inserted there upon clicking the Finish button; otherwise, click the mouse button in the location to insert the banded object.

- **By dragging and dropping**

  1. From the Toolbox panel, drag **Banded Object** to the destination in the report where allows the insertion of a banded object, and a blank banded object will then be created.

  2. In the Resource View panel, select the dataset you want to use in the banded object from the dataset drop-down list. If you want to create a new dataset for the banded object, select **<Choose Data from...>** from the list and new a dataset in the New Dataset dialog.

  3. Drag the required fields in the specified dataset from the Resource View panel into the banded panels.

After a banded object has been created, you can customize its appearance and behavior by setting properties in the Report Inspector, which can be saved as a CSS style for future use (for details, see Creating a CSS style).

Notes:

- When you want to create a banded object on a hierarchical data source, pay attention to the unique features of HDS. For details, refer to Developing reports from HDS.

- When inserting a banded object using the report wizard,

  - If you have added several group by fields in the Group screen, when the banded object is created, these groups will be indented according to the Customize group indent option setting in the Options dialog. You can configure this option in advance to have the groups indented according to your requirements.

  - If you have added some summaries in the Summary screen, when the banded object is created,

    - JReport will automatically give the summaries some proper name labels to help you clarify the meaning of the numbers. The auto generated labels are controlled by the Text property in the Report Inspector. You can also manually edit the text of the label later by setting this property.

    - By default, the summaries that have the same function will be aligned horizontally in the banded object. If you want to align summaries of the same function vertically, check the option "Align summaries vertically" in the Options dialog in advance.

- When you insert an object whose height is determined at runtime into the BandedPageFooter panel (for example, a subreport), but do not set the height of the panel high enough to hold this object, the object might get overlapped with the ones that are in the panel which is above the BPF panel at runtime.

- When you insert a banded object into a banded report with the dataset inherited from the report, you cannot insert the banded object into the BandedPageHeader and BandedPageFooter panel.

Modifying a banded object

Once a banded object has been created, you can further modify it by accessing its shortcut menu wizard which is composed by a set of screens that are similar to the wizard screens used to create the banded object. For example, you can change the data or even the dataset used by the banded object, apply some grouping, sorting criteria to the banded object, and so on.

To modify a banded object after it has been created:

1. Right-click the banded object and select **Banded Wizard** from the shortcut menu to display the Banded Wizard.
2. In the Data screen, specify a new dataset for the banded object if required.
3. In the Display screen, specify the fields you want to display in the banded object and define how data will be sorted in the banded object.
4. In the Group screen, specify the grouping criteria, and then the sorting direction of each group level for the banded object.
5. In the Style screen, select the style you want to apply to the banded object.
6. Upon finishing, click **Finish** to accept the changes.

Notes:

- Due to the specialty of hierarchical data sources, you are not allowed to specify an HDS as the new dataset of the banded object.
- In the Group screen, the newly added fields will be put to the end of the existing fields, and you cannot re-adjust the order of the fields as you did when creating the banded object.
- When you have specified a new dataset for the banded object, if there are existing fields in the banded object that cannot be found in the new dataset, they will be marked with a red cross icon.

  You can use the Replace button to select fields from the new dataset to replace them. Then when you finish the modification, the new dataset and fields you specify will be displayed in the banded object and those remaining fields that cannot be matched to fields in the new dataset will be removed from the banded object automatically.

Managing data of a banded object

With JReport Designer, you can group, sort, and filter data in a banded object easily so as to organize data in the banded object in a more readable way. The operations for setting group, sort and filter conditions for a banded object are similar to those for a table, so you can refer to the following topics on table for details:

- Grouping the data
- Sorting the data
- Filtering the data

Note: When a banded object is created on an HDS, even though you can define some filter conditions on the banded object via the Format Filter dialog, the conditions will not be applied when the banded object runs due to the specialty of HDS. However, if you want to narrow down the data displayed in such kind of banded objects, you can make use of the dataset filter.

In addition, you can insert a web control, Expand/Collapse Group, to the group header panel of a banded object so that end users in Page Report Studio can use the web control to expand/collapse records in groups of the banded object. Note that, this feature works only in continuous page mode.

To insert to a Expand/Collapse Group web control in a banded object:

1. Make sure to uncheck the **Page Layout** command on the View menu.
2. Select the group header panel, then do one of the following:
   - Click **Insert** > **Web Control** > **Expand/Collapse Group** on the menu bar.
   - Select **Expand/Collapse Group** from the **Insert Control Objects** drop-down list on the Insert toolbar.
3. Click on the group header panel, and the web control will be inserted there.

After publishing the report to JReport Server and run it in Page Report Studio, end users will be able to use the web control which appears like a plus or minus sign to expand or collapse a group (that is, to show or hide the details of that group). Furthermore, you can use two properties of a group panel to set the expanding/collapsing state of all groups in the corresponding group level. That is, the Expand Detail Data property controls whether or not to expand the groups, and the Shrink Footer property controls whether or not to hide the group footer panel when collapsing the group.

Managing banded panels

By default, a banded object consists of these panels: a Banded Header (BH) panel, a Banded Page Header (BPH) panel, a Detail (DT) panel, a Parallel Detail (PDT, this is only for the banded objects which are created on HDS) panel, a Banded Page Footer (BPF) panel, a Banded Footer (BF) panel, and several Group Header (GH) and Group Footer (GF) panels if the data in the banded object is grouped.

JReport Designer allows you to perform the following tasks related to the banded panels.

- **Adding a banded panel**

  Select a panel of the same type, right-click it and select **Insert Panel After**. A blank panel will be inserted into the banded object.

- **Deleting a banded panel**

  To delete a banded panel from a banned object, right-click the panel and select **Delete**. Note that you cannot delete a panel of a certain type if it is the only one of that type.

- **Hiding a banded panel**

  To hide a banded panel from view, right-click the panel, and then click **Hide**.

- **Showing a hidden banded panel**

  1. Check the item **Show Invisible Objects** on the View menu, and the hidden panel will then be displayed and marked with two triangles.
  2. Right-click the banded panel, and click **Show** on the shortcut menu to release its hidden state.

- **Resizing a banded panel**

  - For a vertical or cross banded object, drag the boundary below the banded panel until the panel is at the required height.
  - For a horizontal banded object, drag the boundary on the right side of the banded panel until the panel is at the required width.

Crosstabs

A crosstab is a statistical process that summarizes categorical data from one or more sources in a concise format for analysis or reporting. It usually represents the joint distribution of two or more variables in compact row and column format. In business intelligence, engineering, survey and scientific research, the crosstab is widely used.

In JReport, crosstabs are data containers, which means that when a crosstab is inserted into another data container, you can set up data container link between the crosstab and its parent (for details, see the section Setting up data container link).

Meanwhile, in order to meet the requirements from voice agents used by people with eyesight disabilities, JReport supports exported crosstab components in the HTML data table format to get a section 508 compliant result. For detailed descriptions, refer to Accessibility.

Pick a task from the following:

- Inserting a crosstab
- Modifying a crosstab
- Defining comparison functions in a crosstab
- Managing the aggregate fields in a crosstab
- Expanding/Collapsing a crosstab
- Using crosstab formulas

See an example: The SampleComponents catalog, included with JReport Designer, contains reports that have examples of how each component type could be used in a report. For the crosstab component example, open `<install_root>\Demo\Reports\SampleComponents\Crosstab.cls`.

Inserting a crosstab

To insert a crosstab into a report or a library component, use either of the following two ways:

- **Using the wizard**

  1. Position the mouse pointer at the destination where you want to insert the crosstab. A crosstab can be inserted in the report areas listed in Component placement.

  2. Do one of the following to open the Create Crosstab wizard:

     - Click **Insert** > **Crosstab** on the menu bar.

     - Click the **Insert Crosstab** button on the Insert toolbar.

     - If you are inserting the crosstab into a web report or a library component, you can also drag **Crosstab** from the Toolbox panel to display the wizard.

  3. In the Data screen of the wizard,

     - If you are inserting a crosstab into a page report, define or select the dataset on which the crosstab will be created.

       - **New**
         If checked, the crosstab will be built on a new dataset created from the catalog resources.

       - **Existing**
         If checked, specify the dataset in the current report from which the crosstab will be created.

       - **Current**
         If checked, the crosstab will be built on the dataset which is used by its parent.

     - If you are inserting a crosstab into a web report or a library component select a business view in the current catalog on which the crosstab will be created.

  4. In the Display screen, add the required fields from the Resources box to the Columns and Rows boxes respectively to make them displayed on the columns and rows of the crosstab. For the added fields, you can specify their display names in the Label column which will label the rows and columns (by default the Label column is blank and no label will be created for the rows and columns), the background color in the Color column, set the sorting manner of the field values by clicking , and adjust their display order by clicking or . Then, add the fields as aggregate fields to the Summaries box, specify the aggregate functions for the added fields, and edit the display names of the fields in the Label column to label the summaries. If necessary, click the **Comparison Function** button to set the comparison function for the aggregate fields. If the crosstab is inserted into a page report, you can also create some crosstab formulas and add them as aggregate fields so as to apply custom aggregate functions in the crosstab. If you are inserting a crosstab into a web report or a library component, you can also specify a title for the crosstab in the Title text field.

  5. To apply some filters to the crosstab so as to reduce the data displayed in the crosstab, go to the Filter screen and define the filter conditions. If you are inserting the crosstab into a web report or a library component, you can select a predefined filter of the specified business view if there is from the Filter drop-down list to apply, or select **User Defined** in the list to define a new filter. For how to define a filter, refer to Filtering the data.

  6. In the Layout screen, specify the layout of the crosstab.

  7. In the Style screen, set a style for the crosstab.

     If you have specified to insert the crosstab into a banded object in a page report, by default the crosstab will inherit its parent's style. If you want to apply another style to the crosstab, uncheck the **Inherit Style** option and then select the required style from the Style box. For more information, see Inheriting the style from a data container.

  8. Click **Finish** to insert the crosstab.

     If you have specified to insert the crosstab in the report body or tabular cell, the crosstab will be inserted there upon clicking the Finish button; otherwise, you need to click the mouse button in the destination once again in order to insert the crosstab there.

See also the Create Crosstab wizard for page report or web report/library component for detailed information about options in the wizard.

- **Dragging a blank crosstab to a page report**

  1. From the Toolbox panel, drag **Crosstab** to the destination in the report which allows the insertion of a crosstab, and a blank crosstab will then be created.

  2. In the Resource View panel, select the dataset you want to use in the crosstab from the dataset drop-down list. If you want to create a new dataset for the crosstab, select **<Choose Data from...>** from the list and new a dataset in the New Dataset dialog.

  3. Drag the required fields from the Resource View panel to the position of row header, column header or aggregate in the crosstab. Then the rows, columns and aggregates will be created.

When a crosstab has been created, you can customize its appearance and behavior by setting properties in the Report Inspector, which can be saved as a CSS style for future use (for details, see Creating a CSS style).

Note:

- When you want to create a crosstab on a hierarchical data source, pay attention to the unique features of HDS. For details, refer to Developing reports from HDS.

- When you insert a crosstab into a banded report with the dataset inherited from the banded report, you cannot insert the crosstab into the BandedPageHeader and BandedPageFooter panel.

Modifying a crosstab

Once a crosstab has been created, you can further modify it by accessing its shortcut menu wizard which is composed by a set of screens that are similar to the wizard screens used to create the crosstab. For example, you can change the data used by the crosstab, reset the layout and style of the crosstab.

To modify a crosstab after it has been created:

1. Right-click the crosstab and select **Crosstab Wizard** from the shortcut menu to display the Crosstab Wizard.
2. In the Data screen, specify a new data source for the crosstab if required.
3. In the Display screen, specify the fields you want to display in the crosstab.
4. In the Layout screen, specify the layout of the crosstab.
5. In the Style screen, set a style for the crosstab.
6. Upon finishing, click **Finish** to accept the changes.

See also the Crosstab Wizard for page report or web report/library component for detailed explanation about options in the wizard.

After finishing the modification, you can also apply some filter conditions to narrow down the records displayed in the crosstab the same as you do for a table (for details, see Filtering the data). However, if the crosstab is created on an HDS, the conditions you define in the Format Filter dialog will not be applied when the crosstab runs due to the specialty of HDS. Therefore, if you want to filter the data displayed in such kind of crosstabs, you need to make use of the dataset filter.

Notes:

- Due to the specialty of hierarchical data sources, you are not allowed to specify an HDS as the new dataset of the crosstab.
- For a page report, when you have specified a new dataset for the crosstab, if there are existing fields in the crosstab that cannot match the new dataset, they will be marked with a red cross icon. You can use the Replace button to select fields from the new dataset to replace them. Then when you finish the modification, the new dataset and fields you specify will be displayed in the crosstab and those remaining fields that cannot be matched to fields in the new dataset will be removed from the crosstab automatically.
- For a web report, if you specify a new business view for the crosstab, the current crosstab data will be lost and when you finish editing, a new crosstab will be created.

•html>

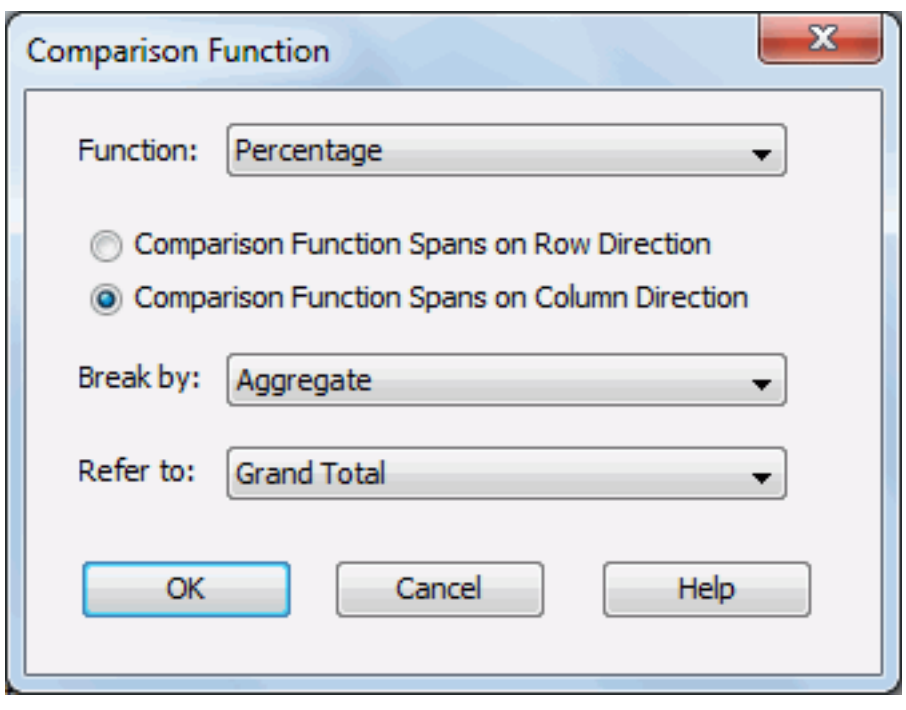
Defining comparison functions in a crosstab

A comparison function refers to calculations of percentage, permillage, or difference between:

- subtotal and grand total
- subtotal of inner group and subtotal of outer group
- values of aggregate field and subtotal
- values of aggregate field and grand total

To define comparison functions in a crosstab:

1. Right-click the crosstab and select **Crosstab Wizard** from the shortcut menu to open the Crosstab Wizard.
2. In the Display screen, select an aggregate field in the Summaries box and then click the **Comparison Function** button. The Comparison Function dialog appears.



3. From the Function drop-down list, select the required function: Percentage, Permillage or Difference.
4. Specify a position for the comparison function.
   - **Comparison Function Spans on Row Direction**
     The comparison function will be placed into the column total cell of the crosstab.
   - **Comparison Function Spans on Column Direction**
     The comparison function will be placed into the row total cell of the crosstab.
5. Numbers that form the calculation of the comparison function are determined by the Break by and Refer to drop-down lists.

   Items in the Break by drop-down list vary with the position of the comparison function. It specifies the first parameter of the comparison function: the aggregate field or a subtotal.

   All available items are displayed in the Refer to drop-down list according to what you have selected from the Break by drop-down list. These items are outer group subtotals and the grand total. Select one as the other parameter of the comparison function.
6. Click **OK** and you can see that a new field is added into the Summaries box. Set the display name for the field in the Label column as required.
7. Repeat the above steps to define more comparison functions.
8. When done, click **Finish** in the Crosstab Wizard to apply the settings.
9. View the report. You will get the values of the comparison function.

Example of using the comparison function

Assume that you have created a crosstab on the query WorldWideSales in the catalog file SampleReports.cat as follows: added Product Type Name and Category in the Products table as the column fields, Country and State in the Customers table as the row fields, Quantity in the Orders Detail table as the aggregate field and specified Sum as the aggregate function, applied a filter "Country = Canada OR Country = France", set the crosstab to be Vertical Layout (Number of Rows: 1), and applied the style Classic. With these settings, the crosstab shows information about product sales volume in each state of Canada and France as follows:

		Decaf						Regular
		Blends	Bold	Espresso	Flavored	Mild	Total	Bold
Canada	BC					22671	22671	40900
	ON	3171			2646		5817	
	Total	3171			2646	22671	28488	40900
France	Paris	6260	10020	426		11229	27935	9254
	Total	6260	10020	426		11229	27935	9254
Total		9431	10020	426	2646	33900	56423	50154

		Regular					Total
		Espresso	Exotic	Flavored	Mild	Total	
Canada	BC		4670	2376	2114	50060	72731
	ON		699		1584	2283	8100
	Total		5369	2376	3698	52343	80831
France	Paris	1624	13865	6941	6778	38462	66397
	Total	1624	13865	6941	6778	38462	66397
Total		1624	19234	9317	10476	90805	147228

Now, you want to define a comparison function in the crosstab to show the percentage of each state's sales volume to the grand total. To do this:

1. Right-click the crosstab and click **Crosstab Wizard** on the shortcut menu.
2. In the Display screen of the Crosstab Wizard, select **Quantity** in the Summaries box, then click the **Comparison Function** button.
3. In the Comparison Function dialog, select **Percentage** from the Function drop-down list, check **Comparison Function Spans on Row Direction**, specify **Product Type Name** as the break by field and choose **Grand Total** from the Refer to list.
4. Click **OK** in the Comparison Function dialog to return to the Crosstab Wizard, then click **Finish** in the wizard to accept the settings.
5. In the Report Inspector, modify the value of the Format property of the fields corresponding to the percentage to **#,###.## %** (the fields are represented as QUANTITY9, QUANTITY10 and QUANTITY11 respectively in the Report Inspector).
6. View the crosstab again and you will find that a percentage is added to the right of each state's sales volume.

		Decaf						Regular
		Blends	Bold	Espresso	Flavored	Mild	Total	Bold
Canada	BC					22671	22671	40900
	ON	3171			2646		5817	
	Total	3171			2646	22671	28488	40900
France	Paris	6260	10020	426		11229	27935	9254
	Total	6260	10020	426		11229	27935	9254
Total		9431	10020	426	2646	33900	56423	50154

		Regular					Total	
		Espresso	Exotic	Flavored	Mild	Total		
Canada	BC		4670	2376	2114	50060	72731	49.4%
	ON		699		1584	2283	8100	5.5%
	Total		5369	2376	3698	52343	80831	54.9%
France	Paris	1624	13865	6941	6778	38462	66397	45.1%
	Total	1624	13865	6941	6778	38462	66397	45.1%
Total		1624	19234	9317	10476	90805	147228	100%

•html>

Managing the aggregate fields in a crosstab

The following fields can be used as aggregate fields to summarize data in crosstabs:

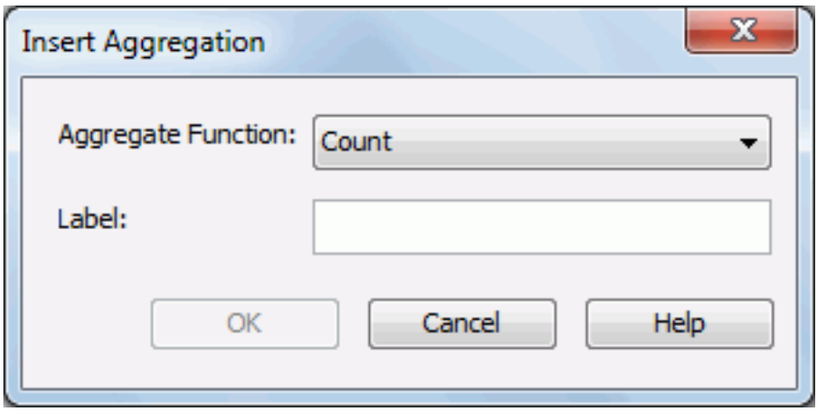
- Query fields, formulas or crosstab formulas for page reports
- Detail fields, aggregation fields, or dynamic formulas/aggregations for web reports or library components

JReport provides you with the following methods to manage the aggregate fields in crosstabs easily:

Inserting an aggregate field by dragging and dropping

When a crosstab is created, you can further drag and drop fields from the Resource View panel to the crosstab to add more aggregate fields to summarize data in the crosstab.

1. Click **View** > **Resource View** on the menu bar to display the Resource View panel if it is not displayed.
2. In the Resource View panel, select a field that can be used as an aggregate field and drag it above or under the desired subtotal or grand total cell until a blue line appears, then release the cursor. The Insert Aggregation dialog appears.



3. From the Aggregate Function drop-down list, specify the function to summarize the selected field.

   If the selected field has been predefined with an aggregate function, you cannot edit the function any more, and the drop-down list is changed to the Aggregation text box with the name of the field displayed in it.
4. In the Label text field, specify a display name for the aggregate field if needed, which will label the newly created summaries.
5. Click **OK** and a new aggregate cell is added to the crosstab.

When you have added an aggregate field using this way, in the Display screen of the Crosstab Wizard, you can find the new aggregate field in the Summaries box along with the specified function and display name.

Switching the aggregate function of an aggregate field

For the aggregate field that has no predefined aggregate function in a crosstab, you can switch the function it uses at anytime. To do this:

1. Right-click the aggregate field and select **Switch Aggregate Function** from the shortcut menu.
2. Choose a new function from the submenu.
3. View the crosstab and the values are recalculated based on the new function.

Removing an aggregate field

To remove an aggregate field from a crosstab, right-click the field and click **Remove Aggregation** on the shortcut menu.

Expanding/Collapsing a crosstab

If a crosstab has more than one row/column group level, you can specify whether or not to enable the crosstab to be expanded at runtime, and set the default expanding/collapsing state of groups in outer level. This behavior is controlled by two properties: Expand Data and Expand Detail Data.

- **Expand Data**: A property on both crosstab, and DBField or formula added into crosstab as dimension (group level). The property on crosstab controls whether or not to enable end users to expand/collapse dimensions in a crosstab, and when it is enabled, you can use the property on certain DBField or formula to specify whether this dimension can be expanded/collapsed respectively.
- **Expand Detail Data**: A property on DBField or formula added into the crosstab as dimension (group level). It specifies whether or not to display the details of this field by default.

Note: These two properties works only in continuous page mode, and after setting the two properties, if you have further modified the crosstab layout, they may not take effect.

The following example shows how to use the two properties:

1. Create a crosstab report on the query WorldWideSales in the catalog file SampleReports.cat as follows: add Product Type Name and Category in the Products table as the column fields, Region, Country and State in the Customers table as the row fields, Quantity in the Orders Detail table as the aggregate field and specify Sum as the aggregate function, set the crosstab to be Vertical Layout (Number of Rows: 1), and apply the style Classic to the crosstab.
2. In the Report Inspector, set the Expand Data property of the crosstab to **true**.
3. Set the Expand Data property of the DBField Country to **false**, and the Expand Detail Data property of the DBField Product Type Name to **false**.
4. Uncheck **Page Layout** on the View menu.
5. Save the crosstab.
6. Click **View** > **Preview As** > **Page Report Result** to preview the report. You can then click the buttons with a plus or minus sign to expand/collapse the Region and Product Type Name dimensions, and since the Expand Detail Data property of Product Type Name is set to false, you will find the details of this dimension are not expanded by default.

			⊞ Decaf	⊞ Regular	Total
			Total	Total	
⊟	Australia	New South Wa	5796	9956	15752
		Total	5796	9956	15752
	China	Beijing	6482	9591	16073
		Hong Kong		8540	8540
		Shanghai	818	13628	14446
		Taipei	9228	13960	23188
		Total	16528	45719	62247
	Japan	Tokyo	9228	13960	23188
		Total	9228	13960	23188
Asia-Pacific	Kazakhstan	Almaty	14605		14605
		Total	14605		14605
	Malaysia	Selangor	890	601	1491
		Total	890	601	1491
	Singapore	Singapore		13976	13976
		Total		13976	13976
	Thailand	Bangkok	11684		11684
		Total	11684		11684
	Vietnam	Hanoi		8540	8540
		Total		8540	8540
	Total		58731	92752	151483
⊟	Belgium	Brussels	6347	13519	19866
		Total	6347	13519	19866
	France	Paris	27935	38462	66397
		Total	27935	38462	66397
	Germany	Bavaria	13926	28857	42783
		Hesse	8002	15631	23633
		Munich	3157	17571	20728
		Total	25085	62059	87144
	Italy	Milan		7906	7906
		Total		7906	7906
	Netherlands	Amsterdam	9235	10509	19744
		Total	9235	10509	19744
	Poland	Mazowsze Voi	3864	7351	11215
Europe, Middl		Total	3864	7351	11215
	Russia	Moscow		8634	8634

Using crosstab formulas

Crosstab formula is a kind of extended formula used only in page report crosstabs. By using crosstab formulas, you can create custom aggregate functions in a crosstab to get the desired results. Crosstab formulas are private resources on crosstab level. They cannot be used beyond its crosstab.

Applying custom aggregate functions in a crosstab by creating crosstab formulas

The following example shows how to apply custom aggregate function in a crosstab simply by creating a crosstab formula:

1. Open the catalog file JinfonetGourmetJava.cat in `<install_root>\Demo\Reports\TutorialReports`.
2. Create a crosstab report with the query WorldWideSales stored in Data Source 1 of the catalog.
3. In the Display screen of the Crosstab Wizard, add Customers_Country as the row field and Customers_Region as the column field.
4. Click **<Add Crosstab Formula...>** in the Crosstab Formulas node in the Resources box.
5. In the Input Crosstab Formula Name dialog, enter **CustomAggregate** and click **OK**. The Crosstab Formula Editor is displayed. Define the formula as follows:

```
currency ctsv1 = @(Sum(@Price));
currency ctsv2 = @(@Customers_Country:ALL,@Customers_Region:ALL,Sum(@Price));
if(ctsv1/ctsv2 >0.005)
return ToText(ctsv1)
else
return "N/A"
```

6. Save the crosstab formula and add it to the Summaries box as the aggregate field.
7. Switch to the **Style** screen and select **Neutral** as the report style from the Style list.
8. Click **Finish** to create the crosstab and preview it. The crosstab will be shown somewhat as follows. You can see that in the aggregate cell, based on the formula expression, "N/A" is displayed if the price value equals to or is less than 5 ‰ of the grand total price $16,337.85, while the actual price is displayed if the price is more than 5 ‰ of the grand total price.

	Asia-Pacific	Europe, Middle	Latin America	North America	Total
Australia	$219.35				$219.35
Belgium		$269.75			$269.75
Brazil			$173.45		$173.45
Canada				$445.60	$445.60
Chile			$168.00		$168.00
China	$432.00				$432.00
France		$715.90			$715.90
Germany		$685.40			$685.40
Italy		$266.95			$266.95
Japan	$208.20				$208.20
Kazakhstan	N/A				N/A
Malaysia	N/A				N/A
Mexico				$258.40	$258.40
Netherlands		$223.40			$223.40
Poland		$163.45			$163.45
Russia		$269.25			$269.25
Saudi Arabia		$160.05			$160.05
Singapore	N/A				N/A
South Africa		$308.70			$308.70
Spain		$281.45			$281.45
Thailand	N/A				N/A
USA				$10,173.10	$10,173.10
United Kingdo		$627.30			$627.30
Vietnam	N/A				N/A
Total	$1,147.70	$3,971.60	$341.45	$10,877.10	$16,337.85

Managing crosstab formulas

You can manage the crosstab formulas of a crosstab in the Resource View panel as follows:

- **Creating crosstab formulas**
  To create a crosstab formula, click **<Add Crosstab Formula...>** in the Crosstab Formulas node, enter a name for the crosstab formula in the Input Crosstab Formula Name dialog and click **OK**, then in the Crosstab Formula Editor, compose the formula as required. You can then insert the crosstab formula as aggregate field to the crosstab or use it to control the properties of objects in the crosstab.
- **Editing a crosstab formula**
  To edit a crosstab formula, select it from the Crosstab Formulas node, then right-click the crosstab formula and click **Edit Formula** on the shortcut menu. In the Crosstab Formula Editor window, edit the crosstab formula as required.
- **Deleting a crosstab formula**
  To delete a crosstab formula, right-click the crosstab formula and select **Delete** from the shortcut menu.
- **Renaming a crosstab formula**
  To rename a crosstab formula, right-click the crosstab formula and select **Rename** from the shortcut menu. Input a new name in the Input Crosstab Formula Name dialog and click **OK** to confirm the change.
- **Viewing properties of a crosstab formula**
  To view the properties of a crosstab formula, right-click the crosstab formula and select **Properties** to show the Properties table.

Notes:

- Crosstab formulas can only be used as aggregate fields in a crosstab.
- Global variables are not supported in crosstab formulas.
- Crosstab formulas cannot reference detail fields and record level pass one formulas, except for referencing them in simple Sum expression.
- In crosstab formulas, the Array functions cannot be used as array_function (field_variable, groupby). Here array_function refers to any Array function defined in JReport. For example, Maximum (@dbfield, "group_field") and Average(@formula, @parameter) are not allowed in crosstab formulas.

•

Charts

Chart organizes and graphically presents data in a way that makes it easy for end users to see comparisons, trends, and patterns in data. It represents the report data in a visually straightforward form. A chart is based on a chart platform. On the platform, chart paper, legend, and labels make up the normal chart. You can create a chart that contains only simple DBFields, or a complicated chart that contains DBFields, groups, summaries, and even formulas. Normally, DBFields, summaries, and formulas in a report are represented in a chart using chart data markers, and groups are used to produce category names and data series names. DBFields can also be used as category names.

Normally, a chart displays values in a static way and you cannot change the values on it once it is created. However, JReport provides you with options to make the chart interactive and dynamic. For example, if your data source uses data that changes quickly over time such as stock market values, you can create a real time chart, so that the chart will update itself based on a defined interval by using the real time data from the data source. You can make a chart move at runtime based on the value changes of a motion field by creating a motion chart. In a motion chart, the chart is playable. You can start or stop the chart to play the dynamic trend of the motion field, control the moving speed of the chart, and if you create a bubble motion chart, you can even use a trail control to make the chart move showing a bubble or line trail. JReport also supports dynamic charts in Excel, that is you can make data in a chart map to a single cell in Excel, then when you change the data in a cell, the corresponding change will also be displayed in the chart.

In addition, you can make your chart scrollable by adding a scrollbar on the chart, using which you can control the visible value range on the X axis of the chart. You can also use the mouse cursor to select the values you are interested in to have the values zoomed in.

In JReport, charts are data containers, which means that when a chart is inserted into another data container, you can set up a data container link between the chart and its parent (for details, see the section Setting up data container link).

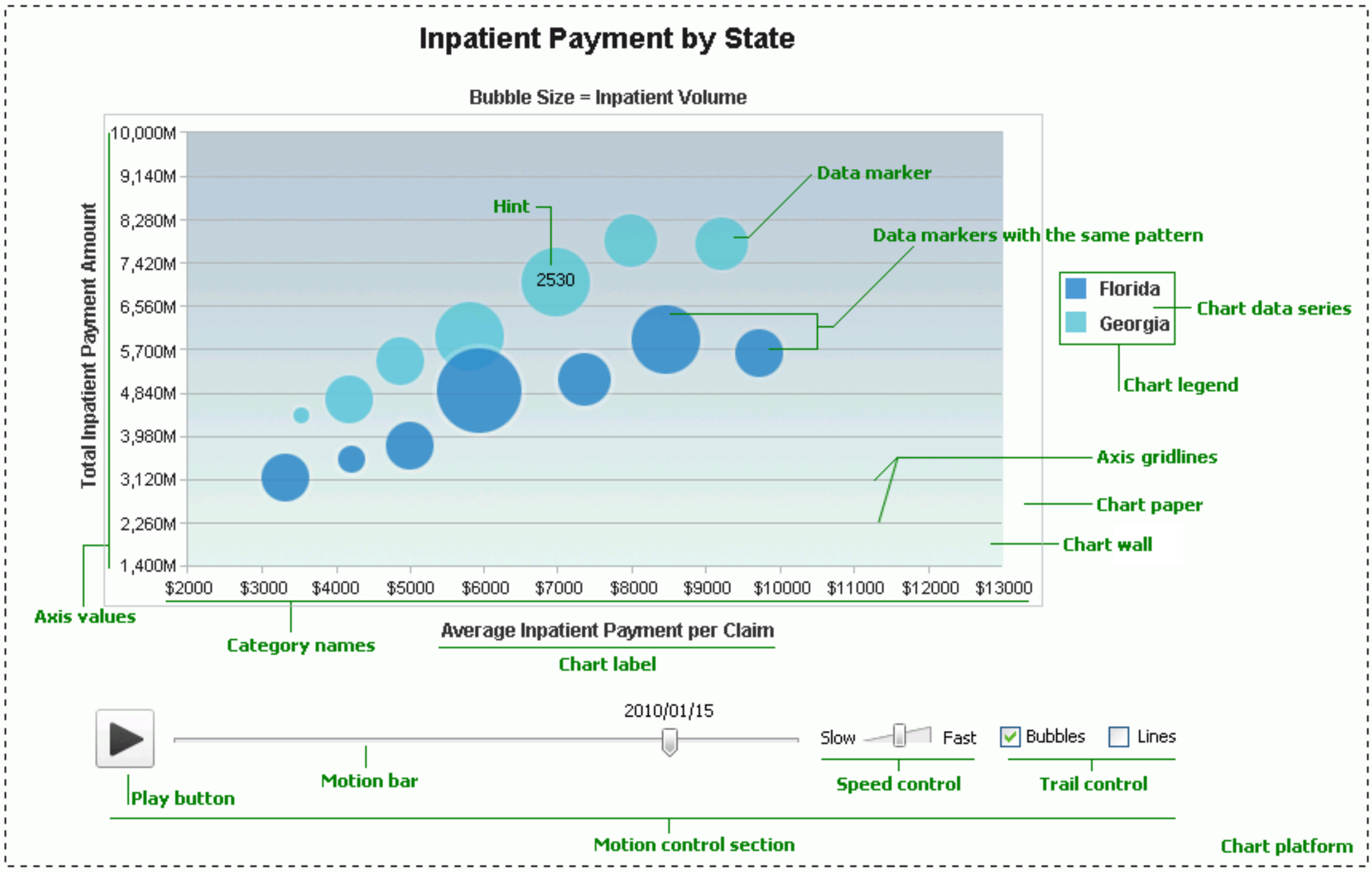
This section describes the following tasks related to charts:

- Chart elements
- Chart types
- How data is represented in a chart
- Inserting a chart
- Modifying a chart
- Formatting chart elements
- Adding conditional color fill to a chart
- Binding a link to a chart
- Creating a scrollable chart
- Creating a real time chart
- Creating a motion chart
- Creating dynamic charts in Excel

See an example: The SampleComponents catalog, included with JReport Designer, contains reports that have examples of how each component type could be used in a report. For the chart component example, open `<install_root>\Demo\Reports\SampleComponents\Chart.cls`.

Chart elements

In a report template, a chart consists of three basic areas: the platform, paper, and legend. The chart platform is a container for other chart elements, and the chart paper holds walls, axes, data markers, axis values, and category names.



- **Chart platform**
  The container of other chart elements, such as chart paper and chart legend.
- **Chart wall**
  The area that graphically displays the data being charted.
- **Chart paper**
  The canvas on which the main chart is drawn. The chart paper contains walls, axes, data markers, axis values, and category names.
- **Chart legend**
  The chart legend is a box which identifies the patterns or colors assigned to the data series or categories in a chart. The chart legend may contain one or more legend entries; each entry is composed of an icon and a label.
- **Chart label**
  A chart label can be used as the chart title or axis title. You can insert a label anywhere inside a chart.
- **Axis values**
  Each axis is divided by some tick marks, and each tick mark is associated with a value which is by default along the axis.
- **Axis gridlines**
  Lines that extend from the tick marks on an axis across the wall of a chart. They help readers of the chart easily find out what value is represented by a tick mark.
- **Category names**
  DBField and formula values can be used as category axis names.
- **Data series names**
  DBField and formula values can be used as series names.
- **Data markers**
  Data markers with the same pattern represent one data series. Each data marker represents one value from the chart data.
- **Hint**
  A hint message that displays the value of a data marker when the mouse pointer points at the data marker in Designer view mode, in HTML result, or at server runtime.
- **Motion control section**
  A section of a motion chart. It is composed of the following elements, using which you can make a chart move based on the value changes of the motion field bound in the motion bar.
  - **Play button**
    The play button controls whether the chart will be moving or static.
  - **Motion bar**
    The motion bar is bound with a motion field. When the value of the motion field changes, the chart will display its dynamic trend.
  - **Speed control**
    It allows you to control the moving speed of the chart.
  - **Trail control**
    Available only for bubble charts. It controls whether the chart will show a trail of previous values in bubbles or lines.

Chart types

The following table lists the JReport chart types.

Area	Clustered Bar 3-D	Clustered Bar: Displays and compares data values across categories.
	Clustered Bar 2-D	Clustered Bar: Displays and compares the contribution of each data value to a total across categories.
	100% Stacked Bar 2-D	100% Stacked Bar: Displays and compares the percentage that each data value contributes to a total across categories.
	Clustered Bar 3-D	Clustered bar with a 3-D visual effect.
	Stacked Bar 2-D	Stacked bar with a 3-D visual effect.
	100% Stacked Bar 3-D	100% stacked bar with a 3-D visual effect.
	Line 2-D	2-D line: Displays and compares data values across categories and series.
Bubble	Stacked Bar3-D	Clustered Bar3: Displays and compares data values across categories and series.
	Stacked Bar3-D 2-D	Stacked Bar3: Displays and compares the contribution of each data value to a total across categories.
	100% Stacked Bar3-D 2-D	100% Stacked Bar3: Displays and compares the percentage that each data value contributes to a total across categories.
	Clustered Bar3-D 3-D	Clustered bar3 with a 3-D visual effect.
	Stacked Bar3-D 3-D	Stacked bar3 with a 3-D visual effect.
	100% Stacked Bar3-D 3-D	100% stacked bar3 with a 3-D visual effect.
	Bar3-D 3-D	3-D Bar3: Displays and compares data values across categories and series.
Line	Line 3-D	Line: Displays trend over categories.
	Stacked Line 2-D	Stacked line: Displays the trend of the contribution of each data value over categories.
	100% Stacked Line 2-D	100% Stacked line: Displays the trend of the percentage each data value contributes over categories.
	Line 3-D	Line with a 3-D visual effect.
Pie	Area 2-D	Displays the trend of the values over time or categories.
	Stacked Area 2-D	Stacked Area: Displays the trend of the contribution of each data value over categories.
	100% Stacked Area 2-D	100% Stacked Area: Displays the trend of the percentage each data value contributes over categories.
	Area 3-D	Area with a 3-D visual effect.
Radar	Pie 2-D	Displays and compares the data values relative to a center point.
	Pie 3-D	Displays the trend of the values over time or categories.
	Exploded Pie	Pie: Displays the contribution of each data value to a total over time or categories.
Stock	Exploded Donut	Donut: Displays the contribution of each data value to the Category total. The Exploded Donut chart displays the contribution of each data value to the Category total, creates a new instance of the chart.
	Donut 2-D	Donut with a 3-D visual effect.
	Donut 3-D	Displays and compares the data values relative to a center point.
Gauge	Clip 2-D	Displays the structure of an organization and the relationships and status ranks of its parts, in the similar encyclopedia.
	Indicator 2-D	Displays each data value with a colored indicator.
Funnel	Gauge Dial 2-D	Usually displays the performance of each member in a group, using three colors to represent the status of each member: green, yellow, and red. The gauge dial, like the gauge clip, displays each data value by a dial.
	Gauge Bar 2-D	Displays each data value with a bar.
Scatter	Gauge Bubble 2-D	Displays each data value with a colored bubble.
	Scatter 2-D	Scatter: Displays the relationship between two values.
	Scatter 3-D	Compares pairs of values.
Gantt	Scatter Bubble Line 2-D	Scatter: with data points connected by smoothed lines.
	Scatter Curved Line 2-D	Scatter: with data points connected by curved lines.
Bubble	Bubble 2-D	Compares sets of 3 values. Similar to a scatter chart with the third value displayed as the size of the bubble marker.
Stock	High-Low 2-D	High-Low: Requires values of two fields.
	High-Low-Close 2-D	High-Low-Close: Requires values of three fields.
Funnel	Open-High-Low-Close 2-D	Open-High-Low-Close: Requires values of four fields.
Gantt	Bullet 2-D	Bullet: Replaces the meters and gauges that are often used on dashboards. Its more design-lite gives a cleaner look and feel, but does support more efficient use of dashboard space.
Heat Map	Heat Map	Displays values by colors and alone in a matrix.
Combo	Combo	Combination charts. Two or more types of data markers are used to represent different data series.

Organization chart

Organization charts (referred to as org chart) is a one-root-node-tree-structure diagram showing the ownership or reporting relationship among the nodes which are mapped to a specific entity. It consists of the following elements:

Child field

The child field identifies the entity.

Parent field

The parent field defines the "reporting to" relationship among the entity members, that is, which child field member reports to or belongs to which parent field member.

- Each parent can have many parent field and child field.
- Both fields are of the same data type.
- The values of the parent field are a subset of those of the child field.
- The two fields should not use the same one.
- One child can only have one parent.
- One parent cannot be the child of one of its child.
- There can be only one root parent.

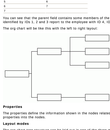
For example, if the parent field is the Employee ID, the parent field should also be about IDs, but the parent field can be different from the child IDs and they set about which ID is the parent of which ID.

Here is a table:

Child field Employee ID	Parent Field Report-To
1	2
2	3
3	4
4	5
5	6
6	7
7	8
8	9
9	10
10	11

You can see that the parent field contains some members of the child field. From the parent field column, the employees identified by ID 2, and 3 report to employee with ID 4; ID 5 reports to ID 6, and ID 6 and 8 report to ID 7.

The org chart will be like this with the left-to-right layout:



Properties

The properties define the information shown in the nodes related to the entity. Data fields, labels and images can be added as properties of the nodes.

Layout mode

The org chart tree structure can be laid out in one of the three modes:

- **Top down**
  The root tree elements from top to bottom.
- **Left to right**
  The org chart tree expands from bottom to top. It is the reverse of the top down type.
- **Left to right**
  The org chart tree expands from left to right.

Heat map

Heat map is composed of rectangles marked by colors and sizes.

Asia
Europe
Africa
Oceania
Americas
Afghanistan
Albania
Algeria
Angola
Argentina
Australia
Austria
Bahrain
Bangladesh
Belgium
Bolivia
Botswana
Brazil
Bulgaria
Burundi
Cambodia
Canada
Chile
China
Colombia
Congo (Brazzaville)
Congo (Kinshasa)
Costa Rica
Croatia
Cuba
Czech Republic
Denmark
Dominican Republic
Ecuador
Egypt
El Salvador
Estonia
Ethiopia
Finland
France
Gabon
Germany
Ghana
Greece
Guatemala
Haiti
Honduras
Hungary
India
Indonesia
Iran
Iraq
Ireland
Israel
Italy
Jamaica
Japan
Jordan
Kenya
Kuwait
Lebanon
Libya
Lithuania
Luxembourg
Madagascar
Malaysia
Mali
Mauritania
Mexico
Mozambique
Nepal
Netherlands
New Zealand
Nicaragua
Niger
Nigeria
Norway
Oman
Pakistan
Panama
Paraguay
Peru
Philippines
Poland
Portugal
Puerto Rico
Romania
Russia
Rwanda
Saudi Arabia
Senegal
Sierra Leone
Singapore
Slovakia
South Africa
South Korea
Spain
Sudan
Sweden
Switzerland
Syria
Taiwan
Tanzania
Thailand
Togo
Trinidad and Tobago
Tunisia
Turkey
Uganda
United Arab Emirates
United Kingdom
United States
Uruguay
Venezuela
Vietnam
Yemen
Zambia
Zimbabwe
Antarctica
Bermuda
Bhutan
Brunei
Cameroon
Cayman Islands
Central African Republic
Chad
Cyprus
Dominica
Equatorial Guinea
Fiji
Gambia
Guinea
Guinea-Bissau
Guyana
Iceland
Kiribati
Laos
Lesotho
Liberia
Malawi
Maldives
Mauritius
Mongolia

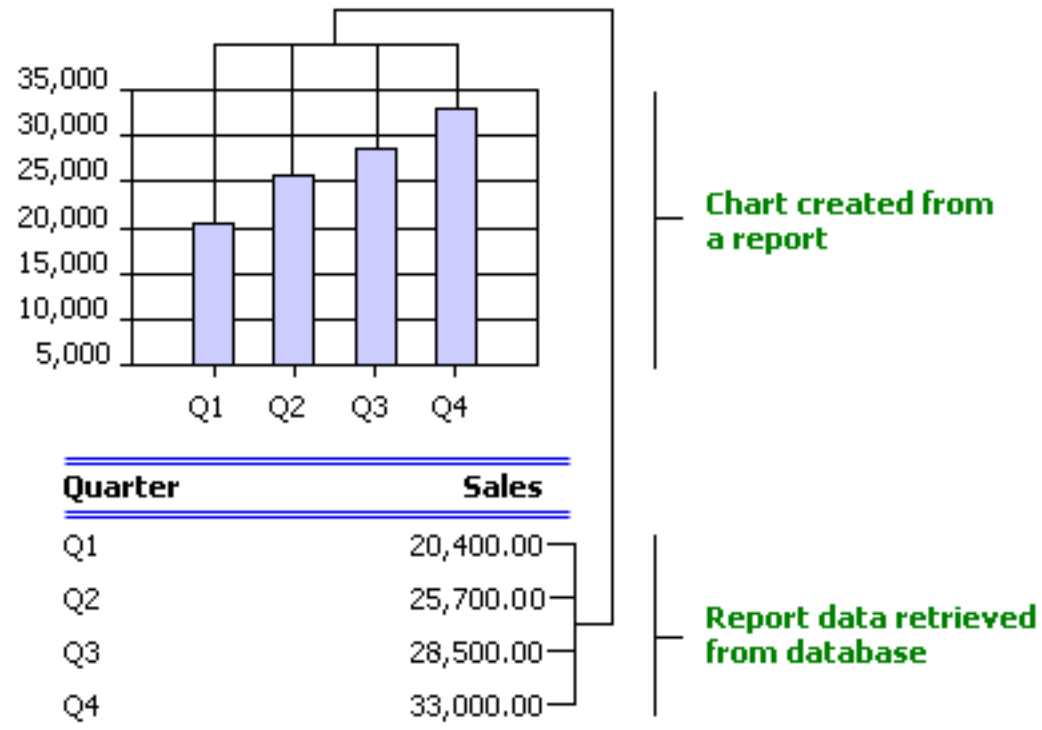
•

How data is represented in a chart

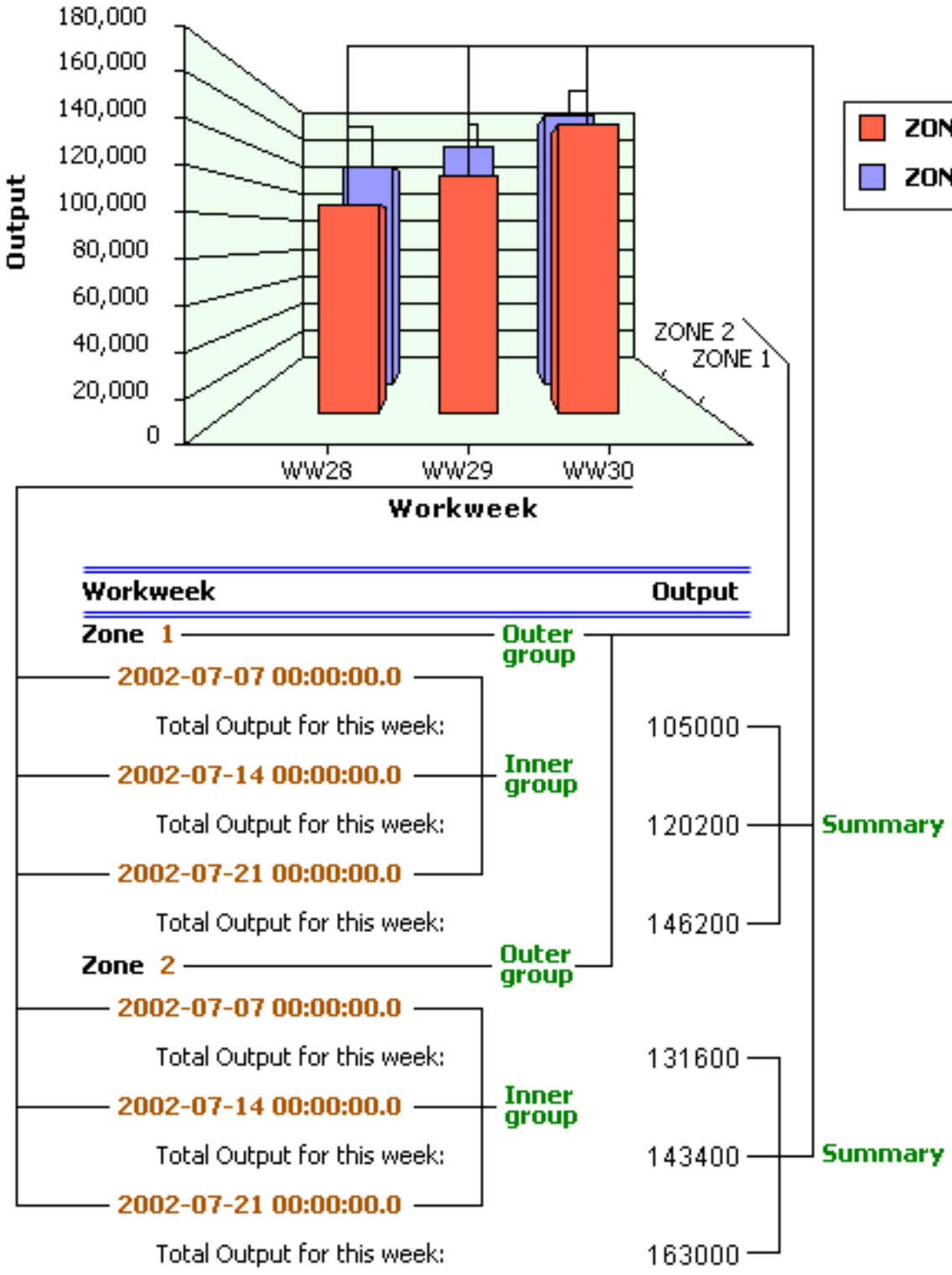
This section describes how data is represented in a chart.

A single chart (except the Stock type) can generally display two- or three-dimensional data, each dimension corresponding to an axis of the chart. A chart displaying three-dimensional data includes the three axes: category axis, series axis, and value axis. A chart will not contain the series axis if it only shows two-dimensional data. The data field displayed on the value axis must be of numerical type. If the chart is of a 3-D subtype, or any subtype of the Pie, Radar, Gauge, or Surface, its value axis can only show one data field; otherwise the value axis can show more than one data field.

A single chart can be based on only detailed records of a dataset, that is, no summary information is to be revealed by the chart. This kind can only display two-dimensional data. For example:



You can also create a chart containing summary information. This kind of chart can display either two- or three-dimensional data. The following is an example showing a three-dimensional-data chart. Notice that the data field in the series axis always holds a higher group level (outer group) than that in the category axis.



If you would like the value axis to display more than one data field, the data fields must be of a same level, that is, all these fields are DBFields or formulas of the dataset (for two-dimensional-data chart), or they are summaries of a same group level (for three-dimensional-data chart).

With respect to a Stock chart, it is designed specially for representing stock trends. You can think of it as a two-dimensional-data chart which must show more than one data field in its value axis. The number of data fields is decided by the subtype of the Stock chart.

A combo chart is different from a single chart in that it has two value axes -- the primary value axis and the secondary value axis, and each value axis can represent one or more subtypes. You can add one or more data fields to any subtype of the two axes and all data fields in the two value axes must be of a same level. Notice that some subtypes are not applicable for combo charts.

Inserting a chart

To insert a chart into a report or a library component, follow the steps below:

1. Position the mouse pointer at the destination where you want to insert the chart. A chart can be inserted in the report areas listed in Component placement.
2. Do one of the following to open the Create Chart wizard:
   - Click **Insert** > **Chart** on the menu bar.
   - Click the **Insert Chart** button on the Insert toolbar.
   - Drag **Chart** from the Toolbox panel to the destination.
3. In the Data screen of the wizard,
   - If you are inserting a chart into a page report, define or select the dataset on which the chart will be created.
     - **New**
       If checked, the chart will be built on a new dataset created from the catalog resources.
     - **Existing**
       If checked, specify the dataset in the current report from which the chart will be created.
     - **Current**
       If checked, the chart will be built on the dataset which is used by its parent.
   - If you are inserting a chart into a web report or a library component, select a business view in the current catalog on which the chart will be created.
4. In the Type screen, specify the type of the chart. For web report or library component, only 2-D types are supported.
   - To create a single chart, check the **Single Chart** radio button, and then select the required type from the Chart Type box and its subtype accordingly.
   - To create a combo chart, check the **Combo Chart** radio button. You can see that the Chart Type Groups box lists the two value axes: the primary axis (Y1) and the secondary axis (Y2), and an <Add Combo Type> item is displayed for each axis. A default chart subtype has been added to the primary axis. To add an additional subtype to an axis, click the corresponding **<Add Combo Type>**, and you will find that a subtype has been added. To change any subtype listed in the box, select it, then choose your required chart type from the Chart Type box, and click the thumbnail for the subtype you want in the Sub Type box. To delete a subtype, select it and click .
5. The Display screen varies with chart types: common chart, organization chart, heat map.
   - **For common chart types**

     Add the required fields from the Resources box to be displayed on the category axis, series axis and value axis of the chart, and define some special group, special function and Order/Select N condition on the category/series field if required.

     If you select the bubble chart type, you need to specify the fields to be shown on the bubble X axis, Y axis and the value you want to show as the bubble radius in the Show Values box. Note that when you specify a value for the bubble X axis, this value will be displayed on the category axis instead of the one specified in the Category box. However, the value defined in the Category box will also be included in data calculation.

     You can also add additional values including constant values and average values to be displayed on the value axis of the chart. To achieve this:

     a. From the Resources box, select **Constant Value**/**Average Value** in the Additional Value node and click beside the Show Values box. The Edit Additional Value dialog appears.

     b. In the Name text box, specify the display name for the constant/average value.

     c. Input the constant value with numeric type in the Value text box, or select a summary/formula field based on which the average value will be calculated from the Based On drop-down list.

     d. Click **OK**, and the defined constant/average value will be added into the Show Values box.

     If you want to further modify a constant/average value, select the value in the Show Values box, then click . In the Edit Additional Value dialog, edit the value as required.

     If you are inserting a chart into a web report or a library component, you can also specify a title for the chart in the Title text field, and choose to make the chart a real time chart or a motion chart.

   - **For organization chart**

     a. Select a field from the Child drop-down list which defines the entity that the org chart nodes are mapped to.

     b. Select a field from the Parent drop-down list which defines the ownership or reporting to relations among the entity members. For example, if the child field is Employee ID, the parent field can be the one about IDs showing which employee ID reports to which employee ID. Note that the parent field should use different one from the child field.

     c. The Properties box displays a node model in the org chart. You can add objects, including data fields, labels and images, from the Resources box into the node and then adjust their positions and sizes in the node and the size of the node. Those added objects will be displayed in each node as the information about the entity members.

     - To add an object, dragging it from the Resources box and dropping to the specific position in the node area in the Properties box, or select it from the Resources box and then click .
     - To remove an object from the node, select it and then click or press Delete on the keyboard.

     For more information about this chart type, see Organization chart.

   - **For heat map**

     a. Add the fields used to group the data into the Groups box. Define some special group, special function and Order/Select N condition on the group field if required.

     b. Add proper summaries into the Summaries box.

     The summaries should match the groups. For example, if the groups level is A > B > C, the static summaries grouped by C can be inserted into the Summaries box, but the static summaries grouped by A, B or other fields cannot.

     If no group is specified in the Groups box, you can insert any static summary. And its group-by field will be inserted into the Groups box automatically.

     c. From both Groups and Summaries boxes, specify the fields to do color by and size by, and select the fields as label-by fields to display in the innermost rectangle in the heat map.

     For more information about this chart type, see Heat Map.
6. To apply some filters to the chart so as to reduce the data displayed in the chart, go to the Filter screen and define the filter conditions.
7. In the Layout screen,
   - For common chart types, with the Data category selected in the Options list box, you can specify the category/series start and end offset, whether or not to reverse the category/series field value sequence, and whether or not to swap groups. Besides, you can set whether or not to show the titles, axes, walls, gridlines, and legend if available. If yes, set the titles for the chart, the category axis and the value axis respectively. If you choose not to display a certain axis, the labels and tick marks along the axis will be hidden as well.
   - For organization chart, in the Type category, choose a way of expanding the org chart tree. Then configure the other settings if required.
   - For heat map, you can specify the chart title and whether to show the chart legend.
8. In the Style screen, select a style for the chart.

   If you have specified to insert the chart into a table or a banded object in a page report, by default the chart will inherit its parent's style. If you want to apply another style to the chart, uncheck the **Inherit Style** option and then select the required style from the Style box. For more information, see Inheriting the style from a data container.
9. Click **Finish** to insert the chart.

   If you have used the menu or toolbar command to insert the chart and specified to insert it to position other than the report body or tabular cell, after clicking Finish in the wizard, you need to click the mouse button in the destination once again in order to insert the chart there.

See also the Create Chart wizard for page report or web report/library component for detailed information about options in the wizard.

When a chart has been created, you can further format its elements according to your requirements. You can also customize its appearance and behavior by setting properties in the Report Inspector, which can be saved as a CSS style for future use (for details, see Creating a CSS style).

For a 2-D chart of bar, bench, line, area or stock type, you can select the values you are interested in to have them zoomed in at Designer preview and at runtime (not available for page reports running in Page Report Studio). To do this:

1. Drag the mouse from the start value to the end value you want to select, then release the mouse. The selected values will be zoomed in. If the chart is equipped with a scroll bar, the bar will be adjusted automatically at the same time.
2. If you want to return to the initial status, just click the return button on the upper right corner of the chart paper.

Notes:

- When you want to create a chart on a hierarchical data source, pay attention to the unique features of HDS (for details, refer to Developing reports from HDS). Moreover, when a chart is created on HDS, only the fields that are of Integer or Number data type in the HDS can be added to the value axis of the chart.
- If you want to insert a chart into the BandedPageHeader panel of a banded object, make sure the chart has its own dataset.
- If you specify to create a pie chart, you can add several fields to be displayed on the value axis of the chart as long as the fields are of Numeric type. However, you can add formulas and DBFields to the axis at the same time, but when a summary has been added to the axis, no other DBFields or formulas can be added.
- If you have defined a subtype of the Stock type, in the Display screen, you should add fields for the subtype in the order indicated by the subtype name, e.g., if you define the Open-High-Low-Close 2-D subtype, then you should add four fields to the Show Values box for this subtype, and arrange them to the order of Open-High-Low-Close by clicking the buttons and .

<html>

Modifying a chart

Once a chart has been created, you can further modify it by accessing its shortcut menu wizard which is composed by a set of screens that are similar to the wizard screens used to create the chart. For example, you can change the data used by the chart, modify the chart type, and so on.

To modify a chart after it has been created:

1. Right-click the chart and select **Chart Wizard** from the shortcut menu to display the Chart Wizard.
2. In the Data screen, specify a new data source for the chart if required.
3. In the Type screen, specify the type of the chart.

   If you want to change the chart type, you may need to first remove all chart data. Specially, if you change a combo type to a single type or vice versa, JReport will prompt you to make sure because all chart data will be removed automatically.
4. In the Display screen, specify the fields to be displayed on the chart.
5. In the Layout screen, specify settings for the chart elements.
6. In the Style screen, select a style for the chart.
7. When done, click **Finish** to accept the changes.

For more detailed steps about how to define a chart, see Inserting a chart.

For an org chart, you can also edit the node model in the design area in the following ways:

- Resize the node model.
- Adjust the sizes and positions of the objects in the node model.

For a heat map, you can also edit it in the design area. Only the first rectangle which represents the innermost rectangle is editable and can contain objects. You can insert labels and images into it, and can also input text in the rectangle directly. For the objects in the rectangle, you can edit or delete them just like they are in a text box.

See also the Chart Wizard for page report or web report/library component for detailed explanation about options in the wizard.

Notes:

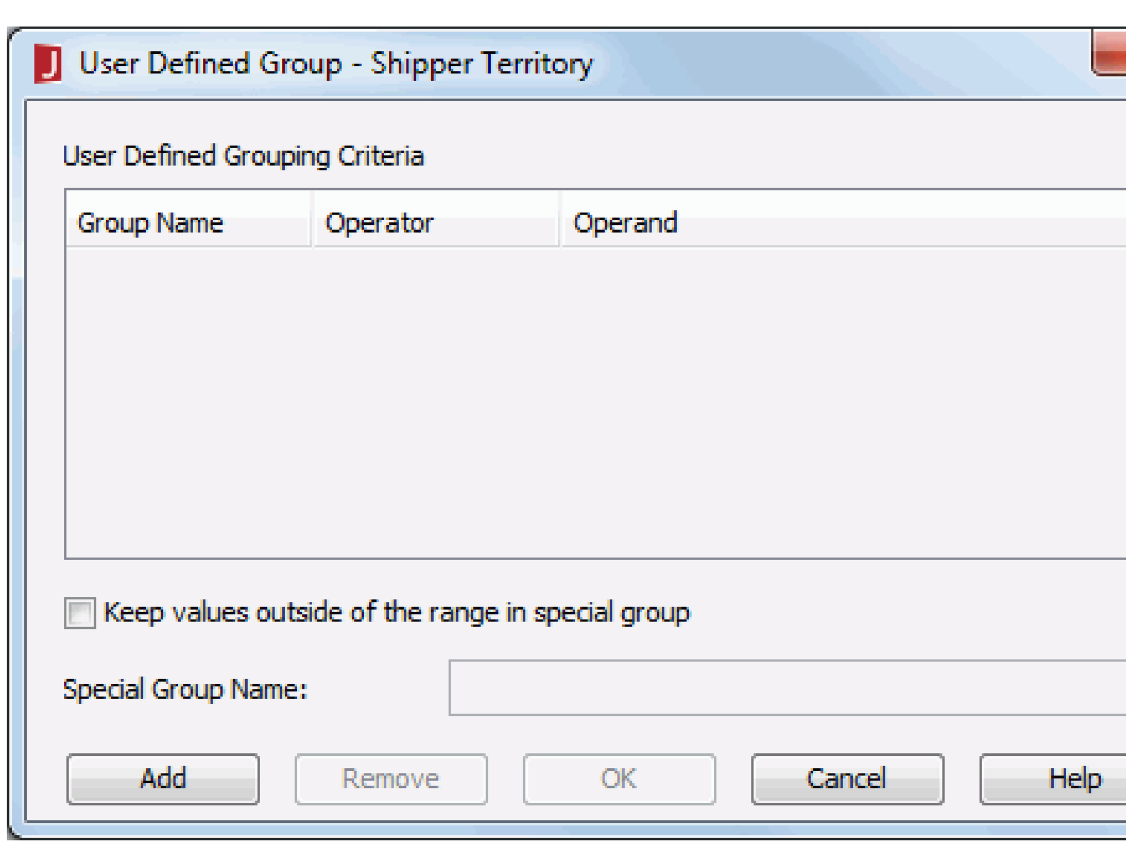
- Due to the specialty of hierarchical data sources, you are not allowed to specify an HDS as the new dataset of the chart.
- While changing a type of 2-D chart to another type of 2-D chart, or a type of 3-D chart to another type of 3-D chart, all the previous properties will be kept. Nevertheless, when changing a 2-D chart to a 3-D chart, the properties of the Z axis, the Y-Z wall, and the floor of the 3-D chart will be set to default values (since a 2-D chart does not have the Z axis, the Y-Z wall, or the floor), and all other properties will be taken from the 2-D chart. When changing a 3-D chart to a 2-D chart, the properties of the Y2 axis of the 2-D chart will be set to default values (since a 3-D chart does not have the Y2 axis).
- When both special functions and Select N conditions are defined in a chart, the former takes higher priority. That is to say, grouping data by intervals takes effect before Top/Bottom N.
- For a page report, when you have specified a new dataset for the chart, if there are existing fields in the chart that cannot match the new dataset, they will be marked with a red cross icon. You can use the Replace button to select fields from the new dataset to replace them. Then when you finish the modification, the new dataset and fields you specify will be displayed in the chart and those remaining fields that cannot be matched to fields in the new dataset will be removed from the chart automatically.
- For a web report, if you specify a new business view for the chart, the current chart data will be lost and when you finish editing, a new chart will be created.

After finishing the modification, you can also apply some filter conditions to narrow down the records displayed in the chart the same as you do for a table (for details, see Filtering the data). However, if the chart is created on an HDS, the conditions you define in the Format Filter dialog will not be applied when the chart runs due to the specialty of HDS. Therefore, if you want to filter the data displayed in such kind of charts, you need to make use of the dataset filter.

Defining special groups for category/series/group values

You can define how to group values on the category/series axis of a chart or on a heat map group. To do this:

1. In the Display screen of the Chart Wizard, click the **Special Group** button. The User Defined Group dialog appears.



2. Click the **Add** button to add a grouping line and specify the Group Name, Operator and Operand as required. Repeat this to add more group criteria.

   For example, if you place a field named Score for grouping which contains student scores that range from 0 to 100, and you want to group the students in 5 ranks, namely rank A: 90~100, B: 80~89, C: 70~79, D: 60~69, and E: 0~59. You can set the groups as follows:

Group Name	Operator	Operand
A	between	Op1: 90, Op2: 100
B	between	Op1: 80, Op2: 89
C	between	Op1: 70, Op2: 79
D	between	Op2: 60, Op2: 69
E	<=	59

   There will be five groups in the order from A to E. If you want to change the order of the groups, you can also do so via the User Defined Group dialog.
3. Check **Keep values outside of the range in special group** checkbox if you want to put the values that are not included in the specified criteria in a new special group, and then provide a name for the special group in the Special Group Name text field.
4. Click **OK** to accept the settings.
5. When done, click **Finish** in the Chart Wizard to apply the changes.

Grouping data on the category/series axis or on a heat map group by intervals

If the data type of the field displayed on the category/series axis of a chart, or the data type of a heat map group field is of Numeric/String/Date/Time type, you can specify some special function to the field so as to group data on the axis or on the heat map group by intervals.

To group data on the category/series axis of a chart or on a heat map group by intervals:

1. In the Display screen of the Chart Wizard, click the **Special Function** button.
2. In the Special Function dialog, select the special function for the category/series/group field from the Function drop-down list (for details about the special functions, refer to Specifying special function for group by field), then click **OK**.
3. Click **Finish** in the Chart Wizard to apply the selected special function to the field.

Notes:

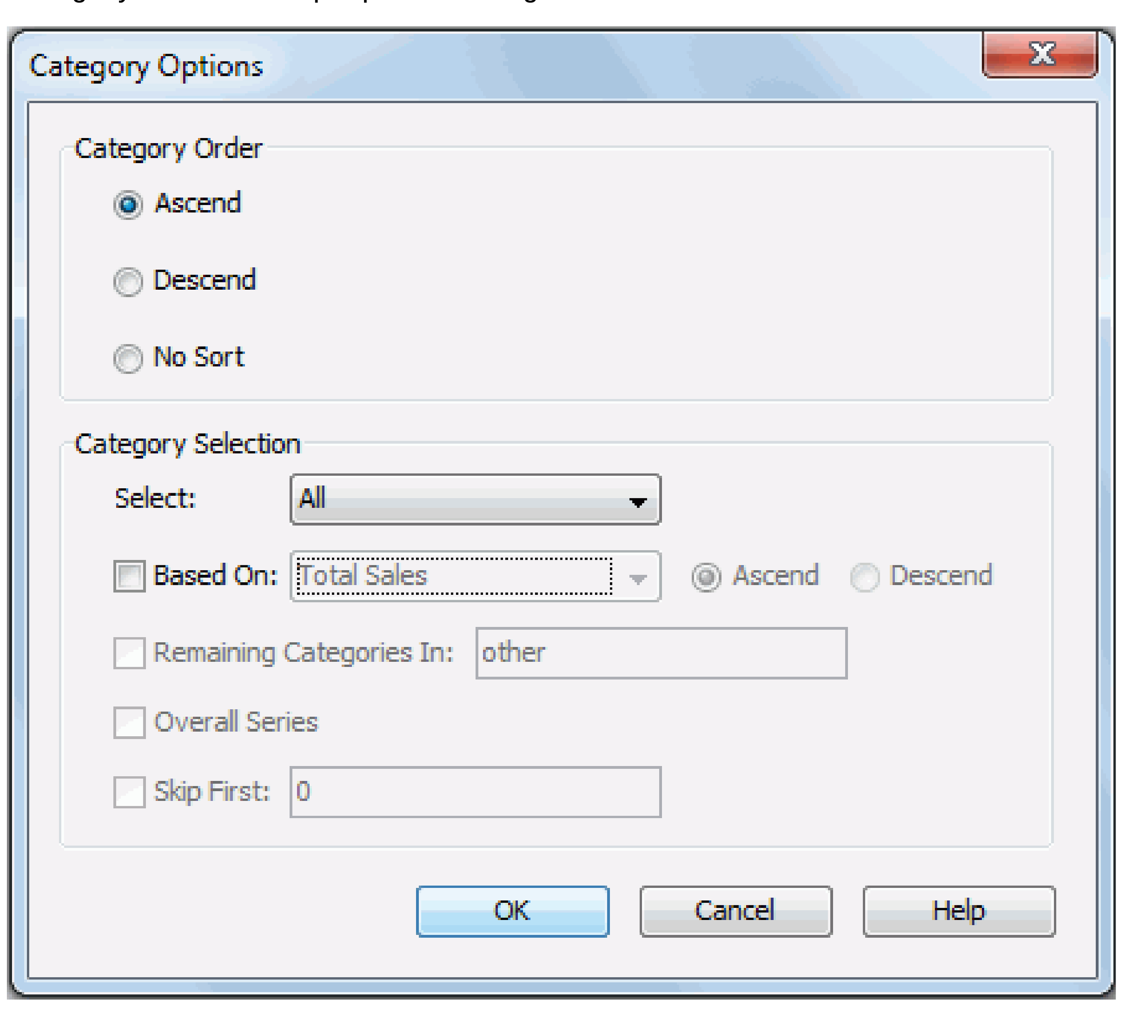
- The special function for field on the category axis should be the same as that of the summary displayed on the value axis. If you add a summary with a special function to the value axis, the special function will be automatically applied to the field on the category axis and cannot be changed. If you want to apply a special function to the summary on the value axis, you should make sure the same special function is added to the category field as well. When the summary has no special function applied, you can add special function to the category field as you want.
- If you have defined a special function to the group by field of a table or banded object, and insert a chart in the table or banded object which inherits dataset from the parent and has the same grouping structure with the parent, then when you view the report, you will find that the special function defined for the very field will be applied to the chart too. However, you can specify another special function to the field on the chart axis according to your requirements.

Setting Order/Select N condition for category/series/group field

Sometimes, you may want to show only certain category/series values in a chart or certain group values in a heat map and make the values sorted according to a specified order. To achieve this, you can use the Order/Select N feature. By specifying a Order/Select N condition, you can decide how many category/series/group values will be displayed in a chart and how the values will be sorted. You can also use an Integer-typed parameter to control the value of Select N.

To specify the Order/Select N condition for the category/series/group field in a chart:

1. In the Display screen of the Chart Wizard, click the **Order/Select N** button to display the Category/Series/Group Options dialog.



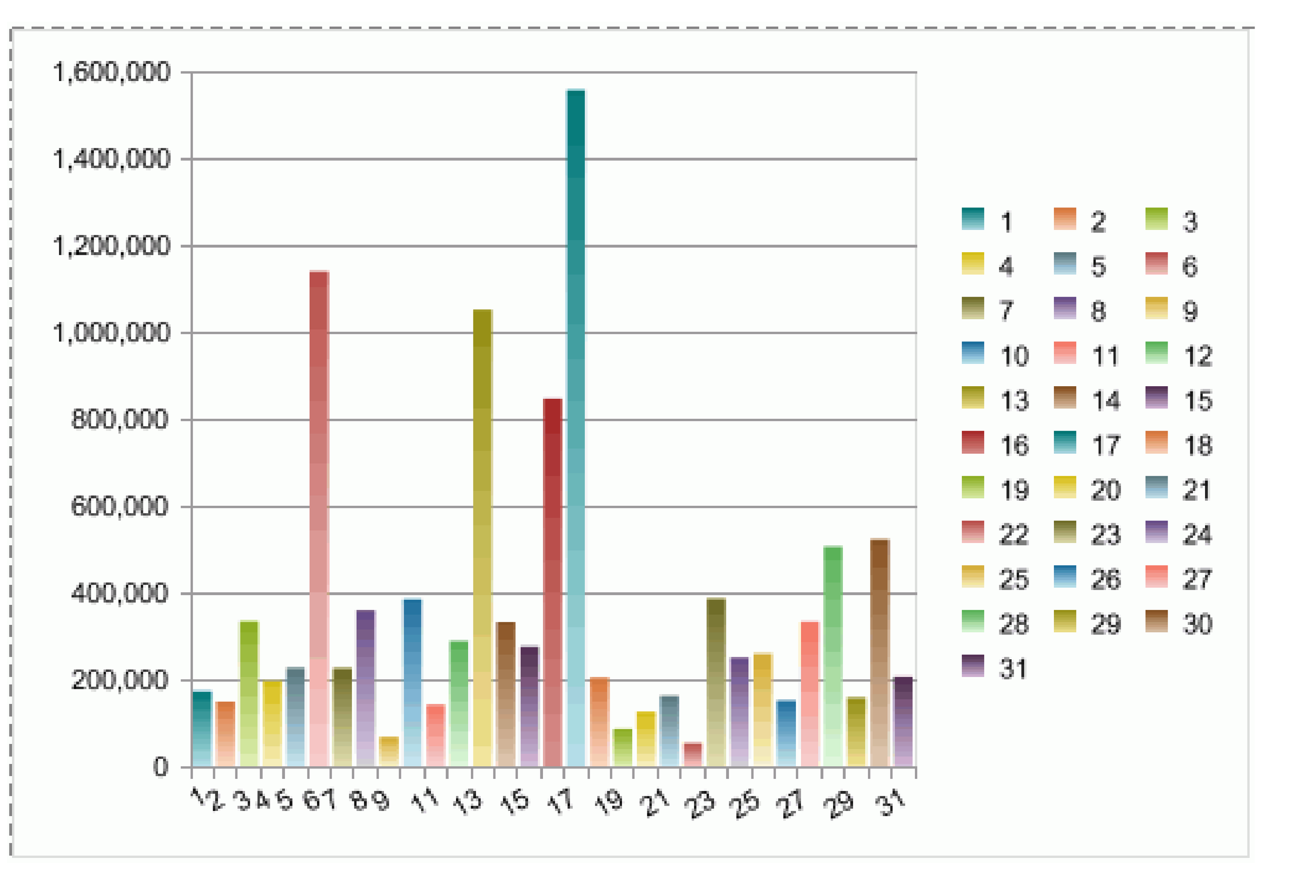
2. In the Category/Series/Group Order box of dialog, specify the sort order of the category/series/group values.
   - **Ascend**
     Values will be sorted in an ascending order.
   - **Descend**
     Values will be sorted in a descending order.
   - **No Sort**
     Values will be sorted in the original order in database.
3. Specify the Select condition to All, Top N or Bottom N.
   - **All**
     If selected, all category/series/group values will be shown in the chart.
   - **Top/Bottom N**
     If selected, the combo box to the right will be enabled and you can specify an integer number in the text field, or select a parameter which returns an integer from the drop-down list to dynamically define the Top/Bottom N condition, then only the first or last N (N is the number you specify by the integer or parameter) category/series/group values will be shown in the chart.

     Note: If you use an Integer-typed parameter to define the Top/Bottom N condition dynamically at runtime, you should make sure that the parameter has at least one default value that is larger than 0, otherwise you will get exceptions when viewing the report.
4. Check the **Based On** checkbox and specify values for the options that follow.

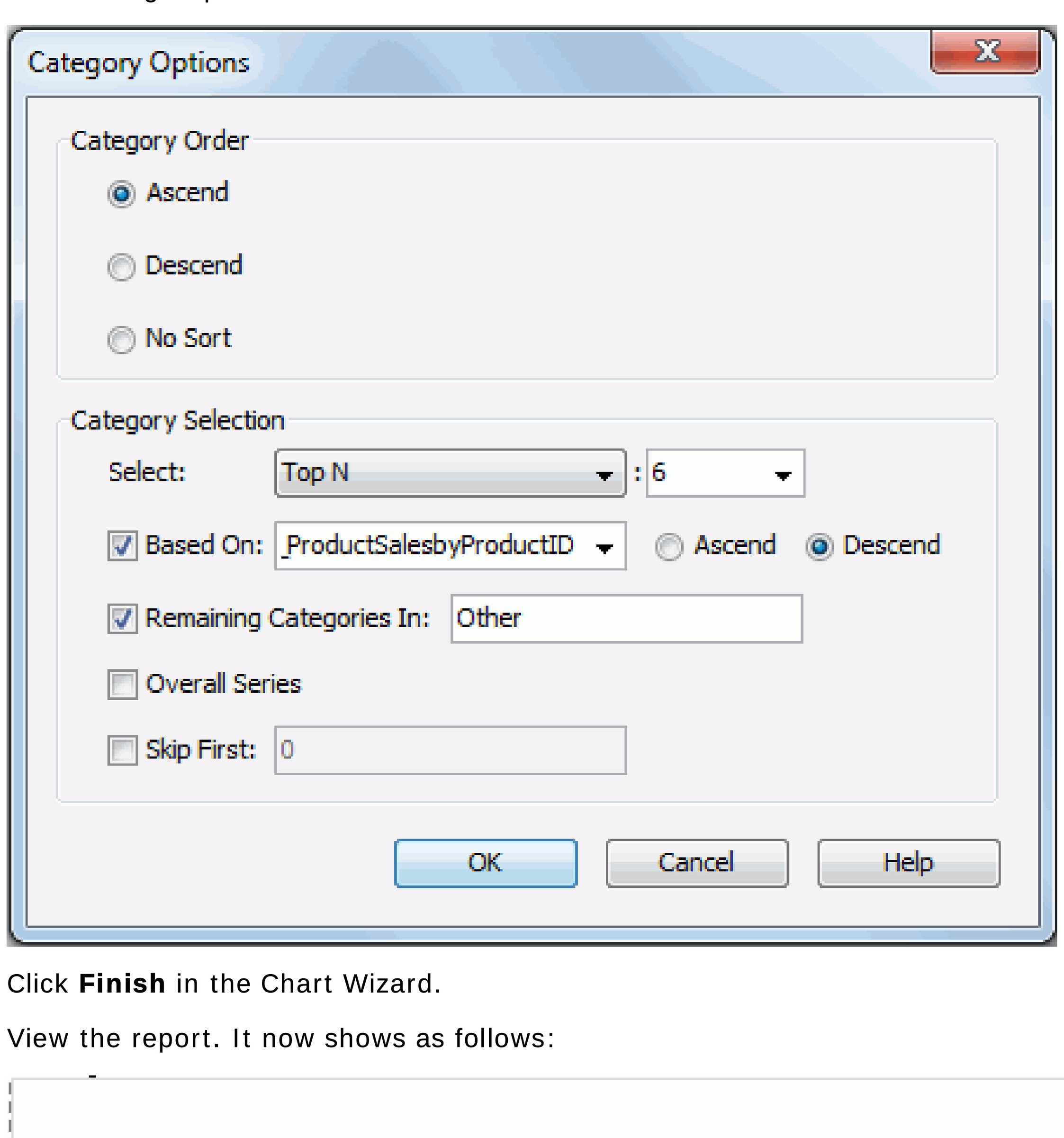
   If Based On is unchecked, the order for the first or last N values will be based on what you specify in the Category/Series/Group Order box of the dialog; if you check it, the order will be based on values of the summary field and the sort direction you specify via the drop-down list and radio buttons next to Based On.
5. If you have selected Top N or Bottom N from the Select drop-down list, you can check the **Remaining Categories In** checkbox and then type a character string in the text field to the right, so that the category/series/group values beyond the first or last N range will be merged into the group with the name as that character string.
6. If you are specifying the Order/Select N condition for the category field, you can also check **Overall Series** to specify to calculate the top or bottom N category values based on the series values.
7. If necessary, check **Skip First** and input a number M in the text field to the right, then the first M category/series/group values will be skipped, and the Select N condition will begin with M+1. The skipped values will be merged into the Remaining Categories group.
8. When done, click **OK** to accept the settings.
9. Click **Finish** in the Chart Wizard to apply the changes.

The following example shows how to control the number of field values that will be displayed on a chart.

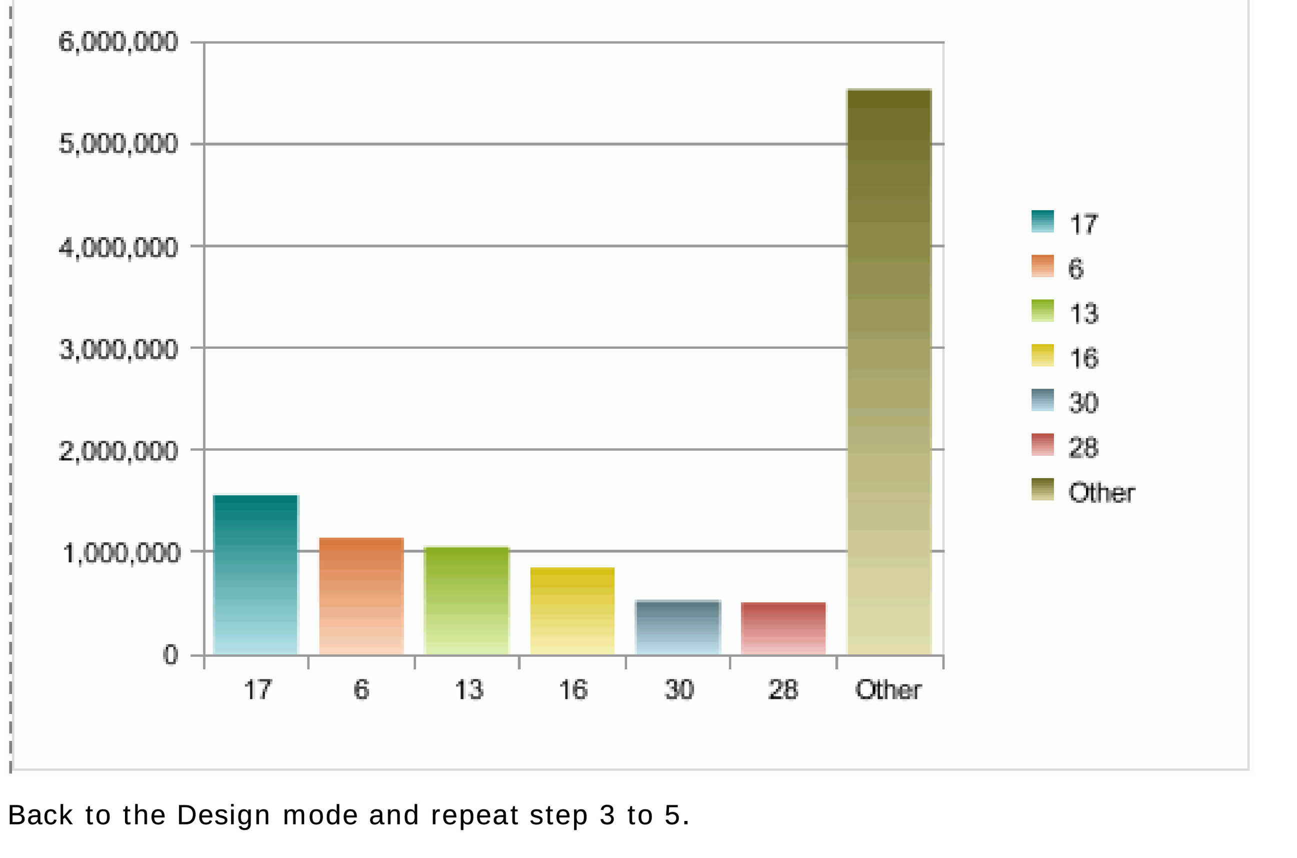
1. Open the catalog file JinfonetGourmetJava.cat in `<install_root>\Demo\Reports\TutorialReports`.
2. Create a chart report based on the OrderListbyDate query, select the Clustered Bar 2-D chart type, add the summary Sum_ProductSalesbyProductID to the value axis, and use Neutral as the report style. The report result looks as follows:



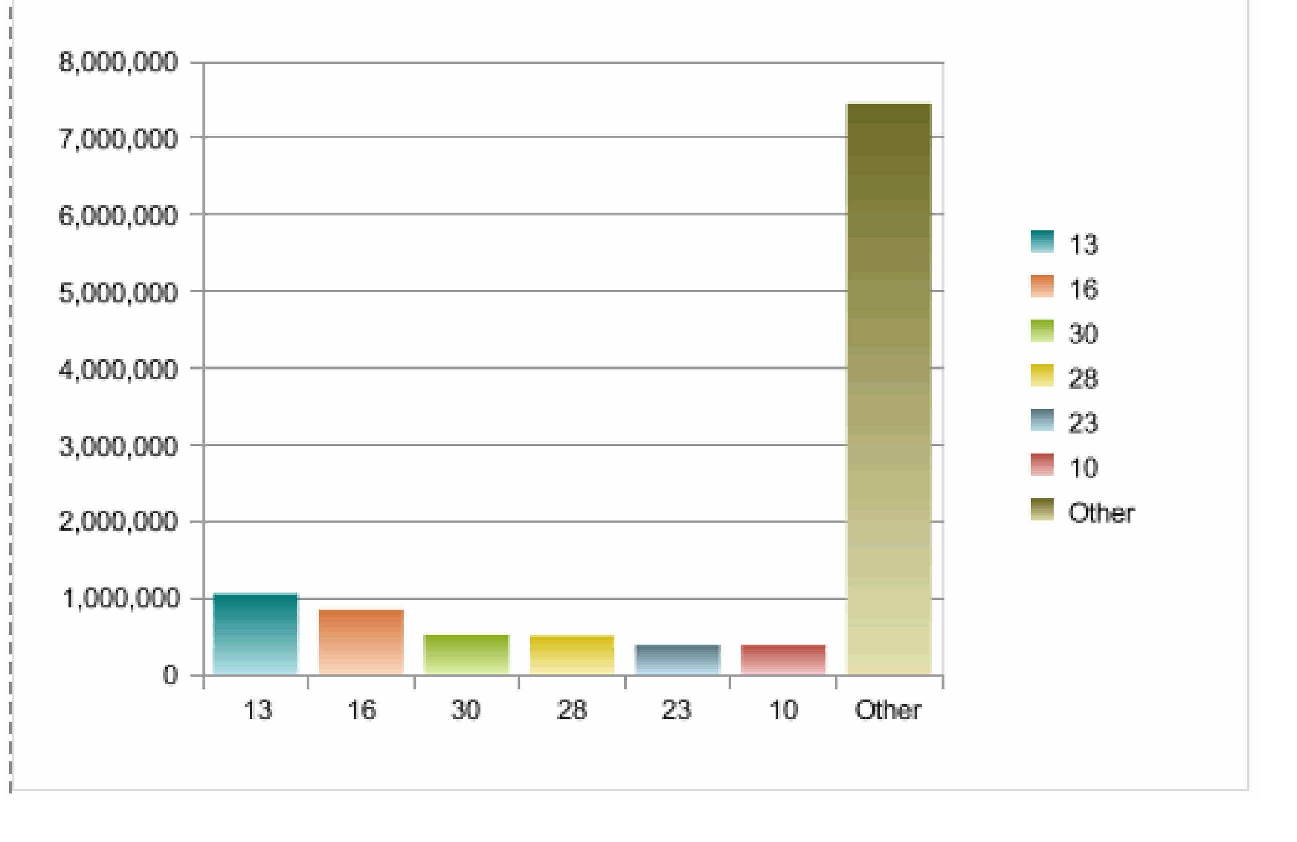
3. Right-click the chart in the report and select **Chart Wizard** from the shortcut menu to open the Chart Wizard.
4. In the Display screen, click the **Order/Select N** button below the Category box.
5. In the Category Options dialog, set the Select condition to Top 6, based on Sum_ProductSalesbyProductID descendingly, then check the **Remaining Categories In** checkbox and input **Other** into the text field so as to put the categories that do not meet the condition into the Other group. Click the **OK** button.



6. Click **Finish** in the Chart Wizard.
7. View the report. It now shows as follows:



8. Back to the Design mode and repeat step 3 to 5.
9. In the Category Options dialog, check the **Skip First** checkbox and input **2** in the text field, then click **OK**.
10. Click **Finish** in the Chart Wizard.
11. View the report again. The result changes as follows:



Formatting chart elements

After a chart has been created, you can further format it according to your requirements.

This section describes how to format elements of a chart as follows:

- Formatting the platform
- Formatting the paper
- Formatting the legend
- Formatting the axes
- Formatting the wall/floor
- Formatting the gridlines
- Formatting the labels
- Formatting the bars
- Formatting the lines
- Formatting the areas
- Formatting the pies/donuts
- Formatting the radar
- Formatting the indicators
- Formatting the gauge
- Formatting the surface
- Formatting the scatter
- Formatting the bubbles
- Formatting the stock
- Formatting the bullets
- Formatting the rectangles
- Formatting the rectangle titles

•

Formatting the platform

The platform of a chart is the container of the other chart elements, such as the paper, legend, and labels.

To format the platform of a chart:

1. Right-click any chart element and select **Format Platform** from the shortcut menu to bring up the Format Platform dialog.

2. In the Fill tab, set the fill and transparency settings of the platform.

   Click the color image button and select a color swatch to fill the platform, and specify the transparency of the color schema (you can also directly type the hexadecimal value of the desired color in the text box). If the color swatches cannot meet your requirement, click **More Colors** on the color palette to open the Pick a Color dialog, with which you can set a full range of color, or click **More Fill Effects** to specify the fill effect in the Fill Effects dialog.

3. In the Border tab, set the border mode, including the border style, color, transparency, line style, thickness, end caps style, and line joint mode, and if you specify the line joint mode to be joint round, you can set the the radius for the border joint. You can also specify whether the border is to be outlined, and whether or not to automatically resize the border dashes if the border is drawn with dashes.

4. When done, click **OK** to apply the settings and close the dialog.

•

Formatting the paper

The paper of a chart is the canvas on which the main chart is drawn. It contains walls, axes, data markers, axis values, category names, and so on.

To format the paper of a chart:

1. Right-click any chart element and select **Format Paper** from the shortcut menu to bring up the Format Paper dialog.

2. Set the scaling ratios or rotation angles for axes in the paper.
   - If the chart is of 2-D type, in the General tab, drag the slider to scale up or down the display size of the chart along the X or Y axis. Check the **Constrain Proportion** option to set the same scaling ratios for both X and Y axes.
   - If the chart is of 3-D type, in the Rotate tab, set the rotation angle for the chart, and specify the scaling ratios for the X, Y and Z axes accordingly. You can either click , , , or , and then drag the chart thumbnail in the preview box to rotate the chart around the origin point, X axis, Z axis, or Y axis, or adjust the degrees directly in the spinners.

     Tip: You can rotate a 3-D chart in another intuitive way as explained below:

     1. Right-click anywhere in the chart and select **Rotation Mode** from the shortcut menu.
     2. Hold the mouse button and move the pointer to the desired position, then release the mouse button to confirm.

        To quit the rotation mode, right-click anywhere in the chart and click **Editing Mode** on the shortcut menu.

3. In the Fill and Border tabs, set the fill and border modes (for details, refer to Formatting the platform).

4. When done, click **OK** to apply the settings and close the dialog.

•

Formatting the legend

Chart legend is a box to identify the patterns or colors assigned to the data categories in a chart. A chart legend may contain one or more legend entries, and each entry is composed of a mark and a label. You can show or hide the legend by right-clicking the chart and then selecting Show/Hide legend form the shortcut menu. An indicator chart whose values are represented by images and an org chart does not have the legend.

To format the legend of a chart:

1. Right-click any chart element and select **Format Legend** from the shortcut menu to bring up the Format Legend dialog.
2. In the Fill and Border tabs, set the fill and border modes for the legend (for details, refer to Formatting the platform).
3. In the Placement tab, specify the way in which the legend will be placed.

   In the Position box, specify whether the legend will be placed at the left, right, top, or bottom of the chart, and then set the alignment style accordingly in the Alignment box. If Auto is selected, the legend will be placed somewhere automatically.

   In the Legend-Label Gap box, specify the vertical and horizontal margins between the legend entries and the legend boundaries, and the vertical and horizontal spacing between legend entry labels.

   Check **Reverse Legend** if you want to have the legend entries re-arranged in a reverse order.

   Check **Show Scrollbar** if you want to show the legend scrollbar to fully view the legend content when the content does not fit into the legend.

   Check **Truncate** if you want to truncate the legend entry label text when the text overflow the labels.
4. In the Font tab, specify the font formats of the legend entry labels, including the font face, size, color, transparency, rotation angle, shearing angle, and effects such as style, strikeout, underline and so on.
5. In the Orientation tab, set an angle for legend entry labels so as to rotate them. You can either drag the Text hand in the Orientation box or type a value in the Angle spinner to set the angle.
6. In the Format tab, specify the data format of the legend entry labels. However, the settings in this tab take effect only when the property Label Format Source of the chart legend in the Report Inspector has been set to Legend Label Format.

   Select a category from the Category box, then select a format from the Format box and click **Add** to add it to the Stack box. You can add more than one format, and in the event that one of the formats is not necessary, you can clear it by clicking Remove.
7. In the Mark tab, specify the options for the legend entry marks.
   - By default, Customize is checked, which enables you to customize the format of the legend entry marks.

     JReport Designer uses 16 styles for the marks by default. To add a style, select a mark item from the Mark Items drop-down list, and then click to add it to the Mark Items box. The legend marks will repeat within the mark items you have defined. If you have defined more mark items than the actual legend marks, the redundant mark items will not be adopted. You can click , , or to remove a mark item, or to arrange the order of the these items.

     To change the style of a mark item, select it in the Mark Items box and then choose a new one from the Mark Items drop-down list.

     In the Icon box, specify the horizontal and vertical alignment styles, and the gap between each entry mark and entry label. You can also set the width and height for the legend entry marks.

     In the Border box, set the color, style, thickness and transparency properties of the icon border.
   - For a line chart, you can also check Use Node as Mark or Use Line and Node as Mark to apply the style and color of the line nodes or lines and line nodes automatically to the legend entry marks. When either of these two is checked, you can also specify the properties of icons in the Icon box.
8. In the Behaviors tab (this tab is only available to library components), select a trigger event from the drop-down list in the Events column, then click in the Actions column and click that appears in the text box to open the Web Action List dialog, where you can bind a web action to the legend which will be triggered when the specified event occurs on the legend.

   To add a web behavior line, click , and if a web behavior is not required, click to remove it.

   Click or to adjust the order of the behaviors. Then, when an event that has been bound with more than one action happens, the upper action will be triggered first.

   The web actions you can bind to the legend include Filter, Sort, Parameter, Property and SendMessage. For details about these web actions, refer to Applying web actions to a label.
9. When done, click **OK** to apply the settings and close the dialog.

See also the Format Legend dialog for page report, web report, or library component for detailed explanation about options in the dialog.

•

Formatting the axes

A chart may contain the following axes (pie, indicator, heat map and org charts do not have axes), this section introduces how to format each axis in detail:

- Formatting the category axis
- Formatting the value axis
- Formatting the series axis

Formatting the category axis

To format the category axis of a chart, take the following steps:

1. Right-click any chart element and click **Format Axes** > **Format Category (X) Axis** on the shortcut menu to bring up the Format Category (X) Axis dialog.
2. In the Axis tab, specify settings for the axis.

   In the Option box, specify whether to show gridlines perpendicular to the axis, and if necessary, check **Show Gridlines**. When creating a scatter chart or the axis is used to show numeric data of a bubble chart, you can also specify the maximum and minimum values to be displayed and the difference between two adjacent values on the axis, and the number of tick marks to be shown on the axis.

   In the Line box, set the line color, style, transparency, and thickness of the axis.

   In the Gap box, set the gap between the labels and the axis, and check **Best Effect** to draw the axis labels for the best result effect.
3. Go to the Tick Mark tab to specify properties for the tick marks.
   a. In the Major Tick Mark and Minor Tick Mark sub tabs, set properties of the major and minor tick marks respectively.

      In the Type box, specify whether to show the tick marks and their position which can be inside/outside the chart or across the axis.

      In the Line box, specify the color, style, transparency and thickness of the tick marks. If you want to use the same line setting as that of the axis for the tick marks, check **Correlate with Axis**. Then specify the length of the tick marks.

      In the Option box, specify whether to show the tick mark labels. When the tick mark labels are shown on the axis, you can control the frequency at which the tick marks will be labeled, whether to display the complete label text when the mouse pointer points at a label on the axis, and the number of the tick mark labels to be displayed on the axis.
   b. In the Scale sub tab, check **Use Constant Interval** to label the major and minor tick marks with constant intervals and specify the minimum and maximum values used to label the tick marks, and the unit between two adjacent tick marks. Then the values for the tick marks will be increased continually based on the specified interval instead of just using values from database. This handles cases where data is missing such as weekend days may not show any activity but still need to be shown in the chart. This sub tab is available only when the field on the category axis is one of the following types: Number, Date, DateTime, and Time, and not applied to scatter charts or when the category axis is used to show numeric data in bubble charts.
4. In the Font tab, format the font of the major and minor tick mark labels respectively, including the font face, size, color, transparency, rotation angle, shearing angle, and effects such as style, strikeout, underline and so on. If you want to apply the font setting of the major tick mark labels to the minor tick marks, check **Correlate with Major Label** in the Minor Label sub tab.
5. In the Orientation tab, set the angles for the major and minor tick mark labels respectively so as to rotate them. You can either drag the Text hand in the Orientation box or type a value in the Angle spinner to perform the operation. You can also make the minor tick marks take the same orientation setting as that of the major tick marks.
6. In the Format tab, specify the data format of the major and minor tick mark labels respectively.

   Select a category from the Category box, then select a format from the Format box and click **Add** to add it to the Stack box. If the formats listed in the Format box cannot meet your requirement, define the format in the Properties text field and click **Add** to add it as the format of the selected category. You can add more than one format, while for each category, only one format can be added. In the event that a format is not necessary, select it in the Stack box click **Remove** to clear it.

   To make the minor tick mark labels apply the format setting of the major tick marks, check **Correlate with Major Label** in the Minor Label sub tab.
7. In the Behaviors tab (this tab is only available to library components), select a trigger event from the drop-down list in the Events column, then click in the Actions column and click [...] that appears in the text box to open the Web Action List dialog, where you can bind a web action to the axis which will be triggered when the specified event occurs on the axis.

   To add a web behavior line, click [+], and if a web behavior is not required, click [delete] to remove it.

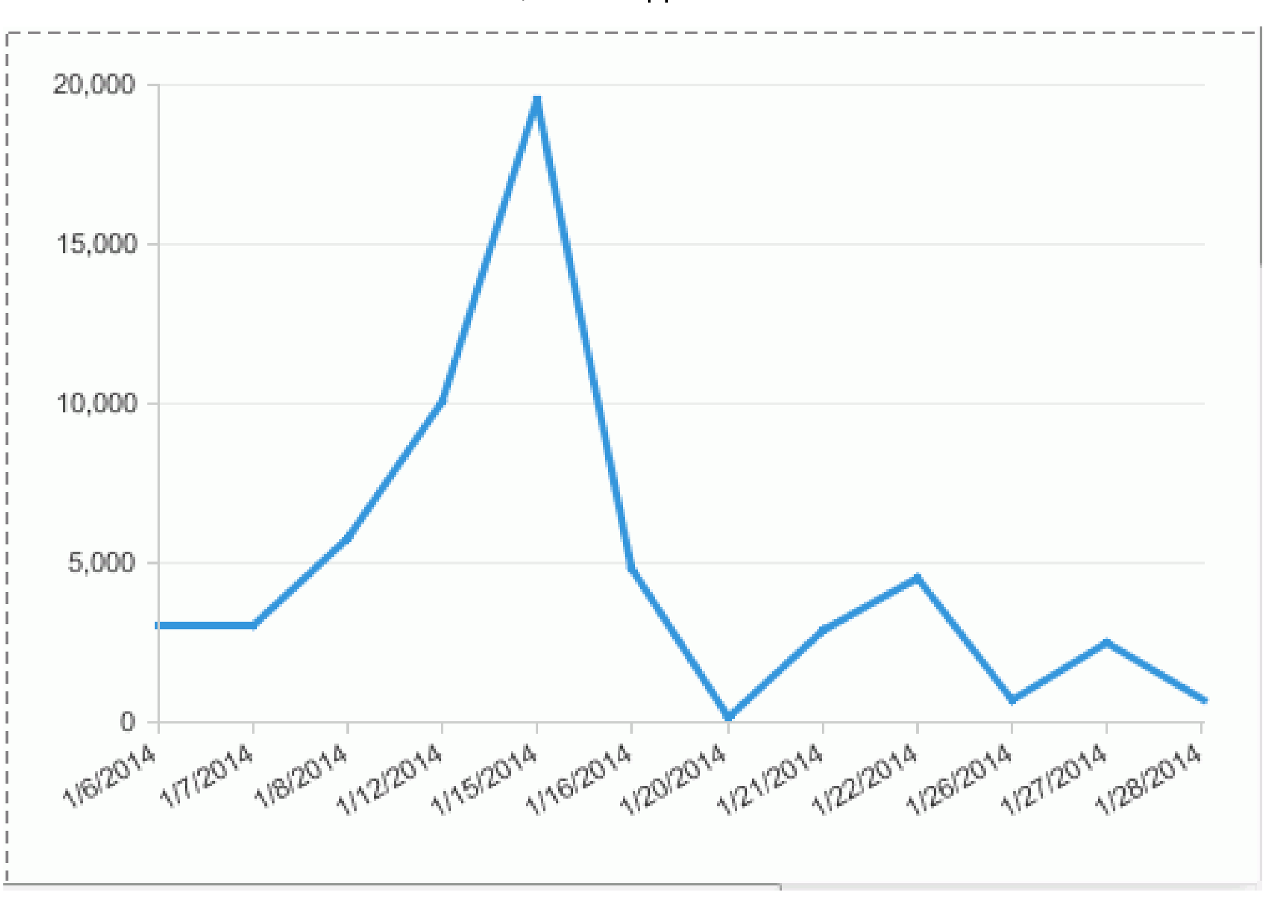
   Click [up] or [down] to adjust the order of the behaviors. Then, when an event that has been bound with more than one action happens, the upper action will be triggered first.

   The web actions you can bind to the axis include Filter, Sort, Parameter, Property and SendMessage. For details about these web actions, refer to Applying web actions to a label.
8. When done, click **OK** to accept the changes.

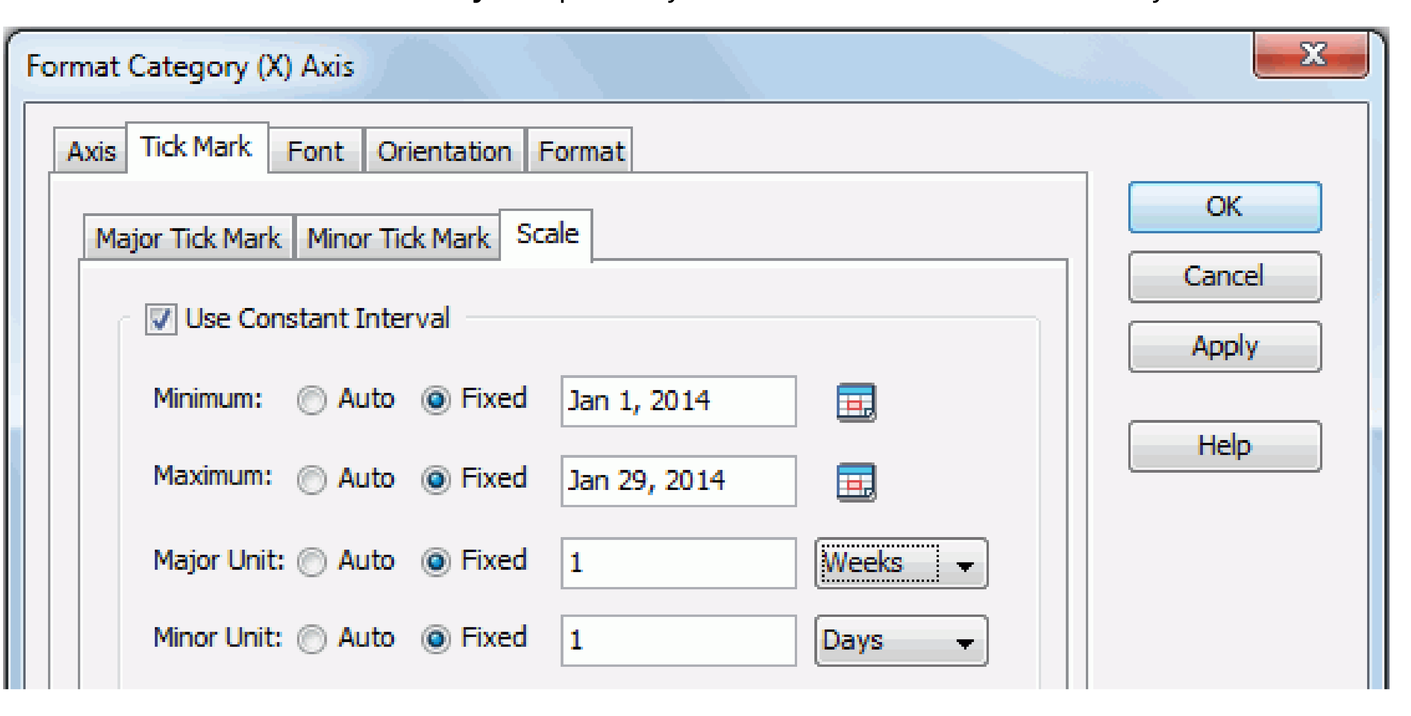
See also the Format Category (X) Axis dialog for report, or library component for detailed explanation about options in the dialog.

Time series data is data collected over time for a single or a group of variables. When using date series and time series values on the category (X) axis it is often beneficial to see the chart with constant ranges rather than just what is in the data values. Missing values for weekends and holidays leaves gaps which distorts the chart and might lead to incorrect decisions. The following example shows how to use a constant interval to label time series data on the category axis:

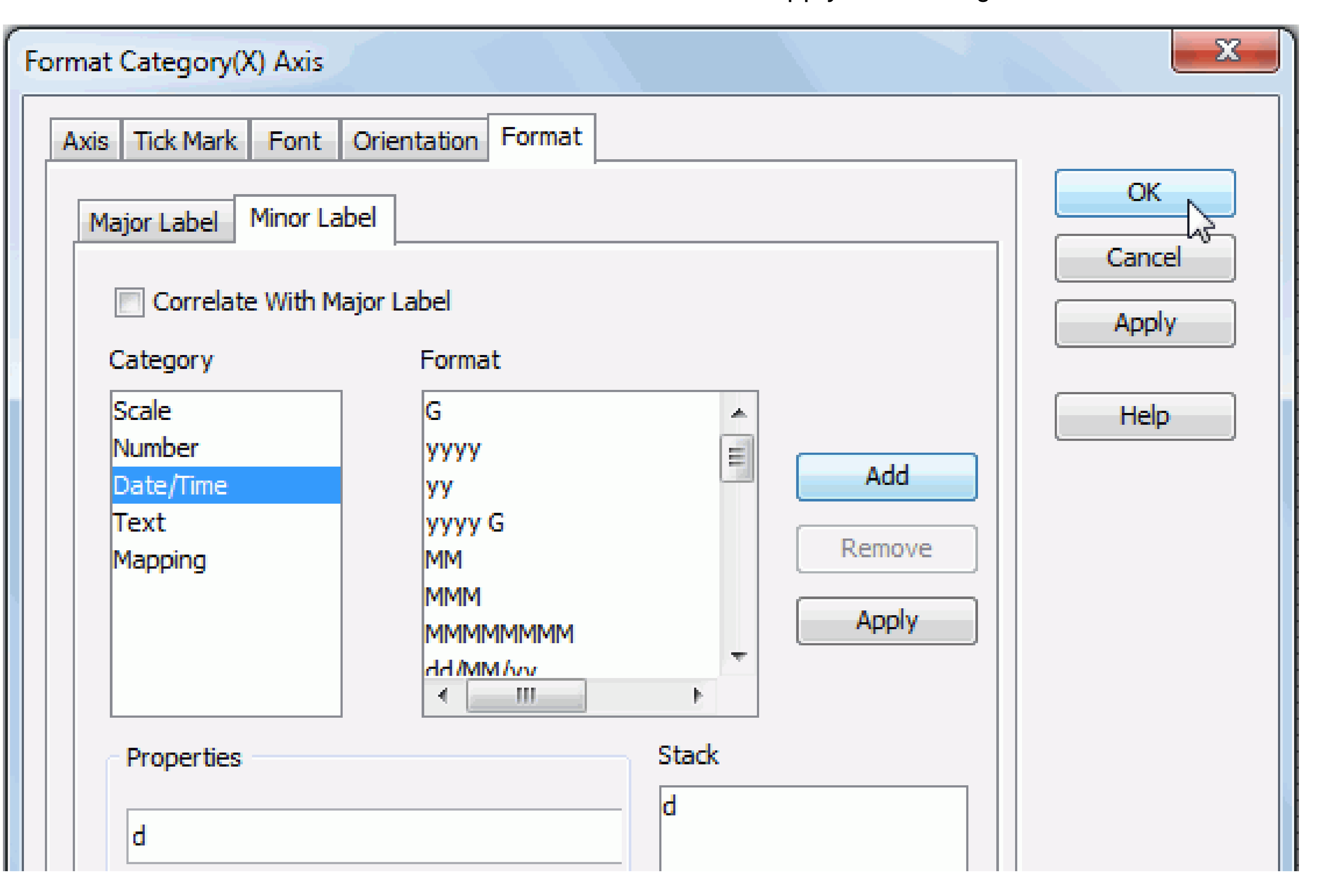
1. Open the catalog file SampleReports.cat.
2. Click **File** > **New** > **Page Report** on the menu bar.
3. Clear the text in the Report Title text box, select **Chart** from the layout box, then click **OK**.
4. In the Data screen of Chart Wizard, check the **New** radio button, select the query **WorldWideSales** in Data Source 1 and then click **Next**.
5. In the Type screen, check the **Single chart** radio button, select **Line** as the chart type and **Line 2-D** as the sub type, then click **Next**.
6. In the Display screen, click **<Add Summary...>** under the Summaries node in the Resources box.
7. In the New Summary dialog, specify the function as **Sum**, add the field **Quantity** from the Orders Detail table to the Summary On field, check the **Static Summary** radio button, specify the Group By field as **Order Date** from the Orders table, and then click **OK**.
8. Enter **Sum Quantity** in the Summary Name dialog and click **OK** to create the summary.
9. Add **Sum Quantity** to the Show Values box, the Order Date field will then be automatically added to the Category box. Click **Next**.
10. In the Filter screen, click **Add Condition** to add a condition line, then select **Order Date** from the field drop-down list, **>=** from the operator drop-down list, and **2014-01-06** from the value drop-down list.
11. Repeat the step above to add another condition line and set the the condition to **Order Date <= 2014-01-28**. Specify the relationship between the two conditions as **And**.
12. Switch to the Style screen and select **Basic** from the Style list. Click **Finish**.
13. Right-click the legend and select **Hide Legend** from the shortcut menu and resize the chart to make the data to be shown more clearly.
14. Click the **View** tab to view this chart, and it appears as follows:



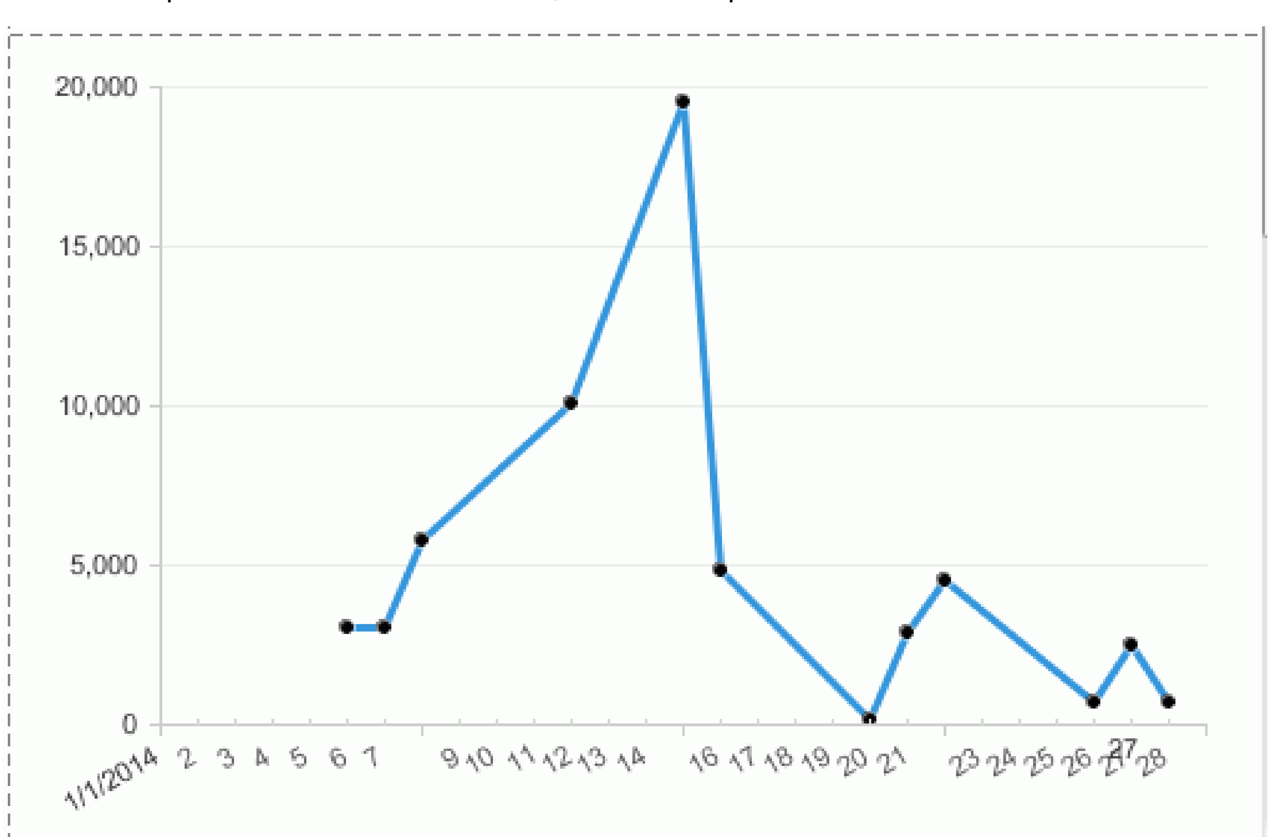
14. Click the **Design** tab to return to the design mode.
15. Right-click the chart and select **Format Axes** > **Format Category (X) Axis** from the shortcut menu. The Format Category (X) Axis dialog appears.
16. In the Tick Mark tab of the dialog, click the **Minor Tick Mark** sub tab, check **Inside** in the Type box to show the minor tick marks inside the axis, and then check **Show Minor Tick Mark Labels** in the Option box to show the minor tick mark labels.
17. Click the **Scale** sub tab, check the **Use Constant Interval** option, then set the Minimum, Maximum, Major Unit and Minor Unit options to **Fixed**. Specify the minimum value to **Jan 1, 2014** and the maximum value to **Jan 29, 2014**. Set the Major Unit and Minor Unit values to **1** and select the unit as **Weeks** and **Days** respectively which stand for 1 week and 1 day.



18. In the Minor Label sub tab of the Format tab, select the format in the Stack box and click **Remove** to delete it. Select **Date/Time** from the Category box, input **d** in the Properties text box and click **Add** to add the format to the Stack box. Then click **OK** to apply the settings.



19. Double-click the line to bring out the Format Line dialog. In the Node tab, select **True** from the Show Node drop-down list to show nodes on the line, and specify the node color to **#000000**. Click **OK** to accept the settings.
20. Save the report and click the **View** tab, then the report will be shown as follows:



Note: The tick marks for date vary based on the actual data values so each tick mark does not represent the same date range.

Formatting the value axis

To format the value axis of a chart, take the following steps:

1. Right-click any chart element and do either of the following:
   - Click **Format Axes** > **Format Value (Y) Axis** on the shortcut menu to bring up the Format Value (Y) Axis dialog.
   - Click **Format Axes** > **Format Value (Y2) Axis** on the shortcut menu to bring up the Format Value (Y2) Axis dialog for a combo chart.
2. In the Axis tab, specify settings for the axis.

   In the Option box, specify the maximum and minimum values to be displayed and the difference between two adjacent values on the axis, the number of tick marks to be shown on the axis, and whether or not to show the value labels on the axis in percent. If you want to show gridlines perpendicular to the axis, check **Show Gridlines**.

   In the Line box, set the line color, style, transparency, and thickness of the axis.

   In the Gap box, set the gap between the labels and the axis, and check **Best Effect** to draw the axis labels for the best result effect.
3. In the Tick Mark tab, specify properties of the major and minor tick marks respectively.

   In the Type box, specify whether to show the tick marks and their position which can be inside/outside the chart or across the axis.

   In the Line box, specify the color, style, transparency and thickness of the tick marks. If you want to use the same line setting as that of the axis for the tick marks, check **Correlate with Axis**. Then specify the length of the major/minor tick marks.

   In the Option box (only available for minor tick marks), specify whether to show the major tick mark labels. When the major tick mark labels are shown on the axis, you can control the frequency at which the major tick marks will be labeled, whether to display the complete label text when the mouse pointer points at a label on the axis, and the number of the major tick mark labels to be displayed on the axis.
4. In the Font tab, format the font for text in the major tick mark labels on the axis, including the font face, size, color, transparency, rotation angle, shearing angle, and effects such as style, strikeout, underline and so on.
5. In the Orientation tab, set an angle for the major tick mark labels on the axis so as to rotate them. You can either drag the Text hand in the Orientation box or type a value in the Angle spinner to perform the operation.
6. In the Format tab, specify the data format of the major tick mark labels on the axis.

   Select a category from the Category box, then select a format from the Format box and click **Add** to add it to the Stack box. If the formats listed in the Format box cannot meet your requirement, define the format in the Properties text field and click **Add** to add it as the format of the selected category. You can add more than one format, while for each category, only one format can be added. In the event that a format is not necessary, select it in the Stack box click **Remove** to clear it.
7. In the Behaviors tab (this tab is only available to library components), select a trigger event from the drop-down list in the Events column, then click in the Actions column and click [...] that appears in the text box to open the Web Action List dialog, where you can bind a web action to the axis which will be triggered when the specified event occurs on the axis.

   To add a web behavior line, click [+], and if a web behavior is not required, click [delete] to remove it.

   Click [up] or [down] to adjust the order of the behaviors. Then, when an event that has been bound with more than one action happens, the upper action will be triggered first.

   The web actions you can bind to the axis include Filter, Sort, Parameter, Property and SendMessage. For details about these web actions, refer to Applying web actions to a label.
8. When done, click **OK** to accept the changes.

Formatting the series axis

To format the series axis of a chart, take the following steps:

1. Right-click any chart element and click **Format Axes** > **Format Series (Z) Axis** on the shortcut menu to bring up the Format Series (Z) Axis dialog.
2. In the Axis tab, specify settings for the axis.

   In the Option box, specify whether or not to only draw the axis (hide the labels), and if not, specify the number of labels to be shown along the axis, the frequency at which the axis will be labeled, whether or not to display the label text when the mouse pointer points at a label on the axis. If you want to show gridlines perpendicular to the axis, check **Show Gridlines**.

   In the Line box, set the line color, style, transparency, and thickness of the axis.

   In the Labels box, set the gap between the labels and the axis, and check **Best Effect** to draw the axis labels for the best result effect.
3. In the Tick Mark tab, specify properties of the tick marks.

   In the Tick Mark box, specify the color, style, transparency and thickness of the tick marks. If you want to use the same line setting as that of the axis for the tick marks, check **Correlate with Axis**.

   In the Major Tick Mark and Minor Tick Mark boxes, set properties for the major and minor tick marks respectively, including whether to show these marks, the position, and length of the marks.
4. In the Font tab, format the font for text in the major tick mark labels on the axis, including the font face, size, color, transparency, rotation angle, shearing angle, and effects such as style, strikeout, underline and so on.
5. In the Orientation tab, set an angle for the major tick mark labels on the axis so as to rotate them. You can either drag the Text hand in the Orientation box or type a value in the Angle spinner to perform the operation.
6. In the Format tab, specify the data format of the major tick mark labels on the axis.

   Select a category from the Category box, then select a format from the Format box and click **Add** to add it to the Stack box. If the formats listed in the Format box cannot meet your requirement, define the format in the Properties text field and click **Add** to add it as the format of the selected category. You can add more than one format, while for each category, only one format can be added. In the event that a format is not necessary, select it in the Stack box click **Remove** to clear it.
7. In the Behaviors tab (this tab is only available to library components), select a trigger event from the drop-down list in the Events column, then click in the Actions column and click [...] that appears in the text box to open the Web Action List dialog, where you can bind a web action to the axis which will be triggered when the specified event occurs on the axis.

   To add a web behavior line, click [+], and if a web behavior is not required, click [delete] to remove it.

   Click [up] or [down] to adjust the order of the behaviors. Then, when an event that has been bound with more than one action happens, the upper action will be triggered first.

   The web actions you can bind to the axis include Filter, Sort, Parameter, Property and SendMessage. For details about these web actions, refer to Applying web actions to a label.
8. When done, click **OK** to accept the changes.

•

Formatting the wall/floor

You can format the wall and floor of a chart at any time. Note that 2-D charts do not have floor, and pie and org charts do not have floor or wall.

To format the wall of a chart:

1. Right-click any chart element and click **Format Walls** > **Format Wall** on the shortcut menu to bring up the Format Wall dialog.
2. Set the fill mode for the wall and bound wall, and the color, style, transparency and thickness of the wall border as required.
3. When done, click **OK** to apply the settings and close the dialog.

To format the floor of a chart:

1. Right-click any chart element and click **Format Walls** > **Format Floor** on the shortcut menu to bring up the Format Floor dialog.
2. Set the fill mode for the floor, and the color, style, transparency and thickness of the floor border as required.
3. When done, click **OK** to apply the settings and close the dialog.

•

Formatting the gridlines

The gridlines of a chart can be customized (pie, indicator, heat map and org charts do not have gridlines). To do this:

1. Right-click any chart element and select the desired entry from the Format Gridlines submenu to bring up the corresponding Format Gridline dialog.
2. Specify the line color, style, transparency, and thickness of the gridline.
3. When done, click **OK** to accept the settings and close the dialog.

See also the following documents for detailed explanation about options in each dialog:

- Format Category (X) Gridline dialog
- Format Value (Y) Gridline dialog
- Format Value (Y2) Gridline dialog
- Format Series (Z) Gridline dialog

•

Formatting the labels

In JReport Designer, you can add as many labels as required into a chart. To add a label to a chart, right-click any chart element and select **Add Label** from the shortcut menu. A label with the text Label will then appear at the upper left corner of the chart platform. To remove a label, right-click it and select **Remove**.

This section shows you how to format the labels (including titles) of a chart to make them more meaningful.

To format the label in a chart:

1. Double-click the label, or right-click the label and select **Format Label** from the shortcut menu to bring up the Format Label dialog.
2. In the General tab, specify the label text, whether or not to attach an icon to the label, the icon settings if applying an icon, the position of the text in the label box, and the icon position relative to the text.
3. In the Fill and Border tabs, specify the fill and border modes for the label (for details, refer to Formatting the platform).
4. In the Font tab, set the font settings for the label.
5. In the Orientation tab, set the rotation angle of the label text and the icon.
6. When done, click **OK** to apply the settings and close the dialog.

•

Formatting the bars

To format the bars in a bar or bench chart, follow the steps below:

1. Right-click any bar in a bar/bench chart and select **Format 2D Bar** or **Format 3D Bar** from the shortcut menu to bring up the Format Bar dialog.

2. In the General tab, specify the general settings of the bars.

   In the Size and Layout boxes, set the size and layout of the bars. If the chart is a clustered 2-D chart which contains the data series (Z axis data), you can further specify the gap between the bars. For a 2-D chart, you can also set the style of the bars to be normal or cylinder, and if you want to add a 3-dimensional effect to the bars, set the Use Depth option to **true** and specify the depth and direction as required.

3. In the Fill tab, specify the fill type: Use Single Color, Use Single Color with Condition or Use Multiple Color with Condition.

   If you select Use Single Color as the fill type,

   - If the chart has no series field, click the color image button and select a color swatch to fill all the bars (you can also directly type the hexadecimal value of the desired color in the text box). If the color swatches cannot meet your requirement, click **More Colors** on the color palette to open the Pick a Color dialog, with which you can set a full range of color, or click **More Fill Effects** to specify the fill effect in the Fill Effects dialog. If you want to make the bar colors vary, check **Vary Colors by Color List**, then click the **Color List** button to specify the color pattern for each bar in the Color List dialog.

   - If the chart has series field, click the color image button and select a color swatch to fill the selected bars in the same data series. You can also click the **Color List** button to specify the color pattern for bars in the same data series in the Color List dialog.

   If you select Use Single Color with Condition or Use Multiple Color with Condition as the fill type, specify the conditions and the color pattern bound with each condition respectively. For details, refer to Adding conditional color fill to a chart.

4. In the Border tab, set the border mode for the bars (for details, see Formatting the platform).

5. In the Data Label tab, specify whether or not to show static data labels for every bar, and if to show, the position of the label related to the bar. When the position of the static data label is set to one of the following: inside center, inside top or inside bottom, you can further specify whether or not to display the data labels inside the bars at the best position. Then in the Font and Effects boxes, set the font formats and effects of the data labels.

6. In the Behaviors tab (this tab is only available to library components), select a trigger event from the drop-down list in the Events column, then click in the Actions column and click that appears in the text box to open the Web Action List dialog, where you can bind a web action to the bars which will be triggered when the specified event occurs on the bars.

   To add a web behavior line, click , and if a web behavior is not required, click to remove it.

   Click or to adjust the order of the behaviors. Then, when an event that has been bound with more than one action happens, the upper action will be triggered first.

   The web actions you can bind to the bars include Filter, Sort, Parameter, Property and SendMessage. For details about these web actions, refer to Applying web actions to a label.

7. When done, click **OK** to accept the changes and close the dialog.

See also the Format Bar dialog for page report, web report, or library component for detailed explanation about options in the dialog.

Notes:

- If the chart is a combo chart composed by bars and areas/lines, when you set the depth properties for the bars, they will be applied to the areas/lines as well, and vice versa.
- For 3-D charts, the Show Static Data Label checkbox in the Data Label tab is grayed out. However, you can still see the data labels by moving the mouse pointer over the bars when viewing the report result in Designer view mode, in HTML format, or in Page Report Studio.
- If you set the style of the bars to be cylinder, the style can take effect only when the Use Depth option is set to true at the same time.

•

Formatting the lines

To format the lines in a line chart, follow the steps below:

1. Right-click any line in the line chart and click **Format 2D Line** or **Format 3D Line** on the shortcut menu to bring up the Format Line dialog.

2. In the General tab, check **Ignore Null Value** if you want the lines to be drawn from the previous data values to the next data values directly when null data values appear to avoid breaks on the lines.

   - For a 2-D line chart, set the layout style of the lines. If you want to add a 3-dimensional effect to the lines, set the Use Depth option to **true** and specify the depth and direction as required.

   - For a 3-D line chart, set the thickness of the lines.

3. In the Fill tab,

   - For a 2-D line chart

     Check **Use Single Color**, then in the Fill box, click the color image button and select a color swatch to fill the selected line, and specify the transparency of the color schema (you can also directly type the hexadecimal value of the desired color in the text box). If the color swatches cannot meet your requirement, click **More Colors** on the color palette to open the Pick a color dialog, with which you can set a full range of color, or click **More Fill Effects** to specify the fill effect in the Fill Effects dialog. You can also click the **Pattern List** button to specify the color pattern for each line in the Color List dialog

     In the Area box, set the fill and transparency settings of the area that are formed by the chart axes and the selected line. You can also click **Area Pattern List** to specify the color pattern for each area formed by each line and the chart axes in the Color List dialog.

     In the Line box, specify the line style and line thickness of the selected line. You can also click **Line Pattern List** to set the line pattern for each line in the Line Pattern List dialog.

   - For a 3-D line chart

     Click the color image button and select a color swatch to fill the selected line. You can also click the **Color List** button to specify the color pattern for each line in the Color List dialog.

4. In the Node tab (the tab is available to 2-D lines only), specify the fill type of the nodes: Use Single Color or Use Single Color with Condition.

   If you select Use Single Color as the fill type, in the Shape box, set Show Node to **true**, then specify the style, transparency, width, height and color of the nodes in the selected line, and specify whether or not to display the line nodes in the highest and lowest points with other colors and whether to show the highlight point in a distinguishing color when you hover the mouse over a node in the line. In the Border box, set the border properties of the nodes on the selected line, including border color, transparency, line style and thickness. If you set the node style to plus, multiplication, star1 or star2, the Line box is available instead of the Border box, in which you can specify the line style and thickness for the nodes with selected styles. You can also click the **Node Pattern List** button to specify the node pattern for each line in the Node Pattern List dialog.

   If you select Use Single Color with Condition as the fill type, specify the conditions and the node pattern bound with each condition respectively. For details, refer to Adding conditional color fill to a chart.

5. In the Border tab (the tab is available to 3-D lines only), set the border mode of the lines, including the border style, color, transparency, line style, thickness, end caps style, and line joint mode. You can also specify whether the border is to be outlined, and whether or not to automatically resize the border dashes if the border is drawn with dashes.

6. In the Data Label tab, specify whether or not to show static data labels for every line node, and if to show, the position of the label related to the node. In the Font and Effects boxes, set the font formats and effects of the data labels.

7. In the Behaviors tab (this tab is only available to library components), select a trigger event from the drop-down list in the Events column, then click in the Actions column and click ... that appears in the text box to open the Web Action List dialog, where you can bind a web action to the lines which will be triggered when the specified event occurs on the lines.

   To add a web behavior line, click +, and if a web behavior is not required, click 🗑 to remove it.

   Click ▲ or ▼ to adjust the order of the behaviors. Then, when an event that has been bound with more than one action happens, the upper action will be triggered first.

   The web actions you can bind to the lines include Filter, Sort, Parameter, Property and SendMessage. For details about these web actions, refer to Applying web actions to a label.

8. Click **OK** to apply the settings and close the dialog.

See also the Format Line dialog for page report, web report, or library component for detailed explanation about options in the dialog.

Notes:

- If the chart is a combo chart composed by lines and areas/bars, when you set the depth properties for the lines, they will be applied to the areas/bars as well, and vice versa.

- For 3-D charts, the Show Static Data Label checkbox in the Data Label tab is grayed out. However, you can still see the data labels by moving the mouse pointer over the lines when viewing the report result in Designer view mode, in HTML format, or in Page Report Studio.

•

Formatting the areas

To format the area of an area chart, follow the steps below:

1. Right-click any area in the area chart and click **Format 2D Area** or **Format 3D Area** on the shortcut menu to bring up the Format Area dialog.

2. In the General tab, set the layout style of the area chart. Check **Ignore Null Value** if you want the areas to be drawn from the previous data values to the next data values directly when null data values appear to avoid breaks on the areas.

   For a 2-D area chart, if you want to add a 3-dimensional effect to the areas, set the Use Depth option to **true** and specify the depth and direction as required. You can also set to show the lines that represent the data categories by checking High-Low Lines.

3. In the Fill tab, specify the fill type: Use Single Color or Use Multiple Colors with Condition.

   If you select Use Single Color as the fill type, click the color image button and select a color swatch to fill the selected area and specify the transparency of the color schema (you can also directly type the hexadecimal value of the desired color in the text box). If the color swatches cannot meet your requirement, click **More Colors** on the color palette to open the Pick a Color dialog, with which you can set a full range of color, or click **More Fill Effects** to specify the fill effect in the Fill Effects dialog. You can also click the **Color List** button to specify the color pattern for each area in the Color List dialog.

   If you select Use Multiple Colors with Condition as the fill type (only applied to Area 2-D and Area 3-D types), specify the conditions and the color pattern bound with each condition respectively. For details, refer to Adding conditional color fill to a chart.

4. In the Border tab, set the border mode for the areas, including the border style, color, transparency, line style, thickness, end caps style, and line joint mode. You can also specify whether the border is to be outlined, and whether or not to automatically resize the border dashes if the border is drawn with dashes.

5. In the Data Label tab, specify whether or not to show static data labels for every area node, and if to show, the position of the label related to the area node. In the Font and Effects boxes, set the font formats and effects of the data labels.

6. In the Behaviors tab (this tab is only available to library components), select a trigger event from the drop-down list in the Events column, then click in the Actions column and click ... that appears in the text box to open the Web Action List dialog, where you can bind a web action to the areas which will be triggered when the specified event occurs on the areas.

   To add a web behavior line, click +, and if a web behavior is not required, click 🗑 to remove it.

   Click ▲ or ▼ to adjust the order of the behaviors. Then, when an event that has been bound with more than one action happens, the upper action will be triggered first.

   The web actions you can bind to the areas include Filter, Sort, Parameter, Property and SendMessage. For details about these web actions, refer to Applying web actions to a label.

7. When done, click **OK** to apply the settings and close the dialog.

See also the Format Area dialog for page report, web report, or library component for detailed explanation about options in the dialog.

Notes:

- If the chart is a combo chart composed by areas and bars/lines, when you set the depth properties for the areas, they will be applied to the bars/lines as well, and vice versa.
- For 3-D charts, the Show Static Data Label checkbox in the Data Label tab is grayed out. However, you can still see the data labels by moving the mouse pointer over the areas when viewing the report result in Designer view mode, in HTML format, or in Page Report Studio.

•

Formatting the pies/donuts

To format the pies/donuts of a pie/donut chart, follow the steps below:

1. Right-click any portion of a pie/donut in the pie/donut chart, and then click **Format Pie/Donut** on the shortcut menu to bring up the Format Pie/Donut dialog.

2. In the General tab, set the general settings for the pies/donuts.

   In the Gap Amount field, enter a number to set the distance between every two adjacent pies/donuts. In the Explode Amount field, enter a number to specify the distance between the sections in each pie/donut and the pie/donut center. In the Data Label Type field, specify the type of the data labels for the pie/donut sections. For a donut chart, you can also specify the radius percentage of the donut hole to the total pie circle by setting the option Donut Hole.

3. In the Fill tab, specify the fill type: Use Single Color or Use Single Color with Condition.

   If you select Use Single Color as the fill type, click the color image button and select a color swatch to fill the selected sections in the same data series, and specify transparency of the color schema (you can also directly type the hexadecimal value of the desired color in the text box). If the color swatches cannot meet your requirement, click **More Colors** on the color palette to open the Pick a Color dialog, with which you can set a full range of color, or click **More Fill Effects** to specify the fill effect in the Fill Effects dialog. You can also click the **Color List** button to specify the color pattern for sections in the same data series respectively in the Color List dialog.

   If you select Use Single Color with Condition, specify the conditions and the color pattern bound with each condition respectively. For details, refer to Adding conditional color fill to a chart.

4. In the Border tab, set the border mode for the sections, including the border style, color, transparency, line style, thickness, end caps style, and line joint mode. You can also specify whether the border is to be outlined, and whether or not to automatically resize the border dashes if the border is drawn with dashes.

5. In the Data Label tab, specify whether or not to show static data labels for every section, and if to show, the position of the label related to the section and the color of the thin lines that are used to point to the static data labels. In the Font and Effects boxes, set the font formats and effects of the data labels.

6. In the Behaviors tab (this tab is only available to library components), select a trigger event from the drop-down list in the Events column, then click in the Actions column and click that appears in the text box to open the Web Action List dialog, where you can bind a web action to the pies/donuts which will be triggered when the specified event occurs on the pies/donuts.

   To add a web behavior line, click , and if a web behavior is not required, click to remove it.

   Click or to adjust the order of the behaviors. Then, when an event that has been bound with more than one action happens, the upper action will be triggered first.

   The web actions you can bind to the pies/donuts include Filter, Sort, Parameter, Property and SendMessage. For details about these web actions, refer to Applying web actions to a label.

7. When done, click **OK** to apply the settings and close the dialog.

See also the following dialogs for detailed explanation about options in each dialog:

- Format Pie dialog for report
- Format Pie dialog for library component
- Format Donut dialog for report
- Format Donut dialog for library component

•

Formatting the radar

To format a radar chart, follow the steps below:

1. Right-click any line which connects two value nodes in the radar chart, and then select **Format Radar** from the shortcut menu to bring up the Format Radar dialog.

2. In the General tab, specify the general settings of the radar.

   Set whether or not to show the category name in the radar chart, whether or not to fill the areas formed by value nodes of a same data series in the radar chart, and if to fill, the transparency of the fill. For the lines connecting value nodes, you can set the thickness, the node style, and the node width and height. For the numeric axes, you can set the style for their arrows in the Arrow Style box.

3. In the Fill tab, set the fill mode for the selected radar areas in the same data series, and the fill mode will also apply to its value node. Click the color image button and select a color swatch to fill the selected radar areas in the same data series, and specify transparency of the color schema (you can also directly type the hexadecimal value of the desired color in the text box). If the color swatches cannot meet your requirement, click **More Colors** on the color palette to open the Pick a Color dialog, with which you can set a full range of color, or click **More Fill Effects** to specify the fill effect in the Fill Effects dialog. You can also click the **Color List** button to specify the color pattern for radar areas in the same data series respectively in the Color List dialog.

4. In the Border tab, set the border mode for the radars, including the border style, color, transparency, line style, thickness, end caps style, and line joint mode. You can also specify whether the border is to be outlined, and whether or not to automatically resize the border dashes if the border is drawn with dashes.

5. In the Data Label tab, specify whether or not to show static data labels for every value node, and if to show, the position of the label related to the node. In the Font and Effects boxes, set the font formats and effects of the data labels.

6. When done, click **OK** to apply the settings and close the dialog.

•

Formatting the indicators

An indicator chart can be formatted to take some special appearances. To format the indicators in an indicator chart, follow the steps below:

1. Right-click any indicator in the indicator chart and select **Format Indicator** from the shortcut menu to bring up the Format Indicator dialog.
2. In the General tab, set the general options for the indicators in the chart.
   - For the indicators whose values are of Boolean type, set the left margin, top margin, and range radius of the indicators in the Option box. In the Color Range box, if you want to use colors to represent different values, uncheck **Show Image**, then specify the colors and names for the values; if you want to use images to represent the values, check **Show Image**, then specify the image and names for the values.
   - For the indicators whose values are of String/Numeric type, set the left margin, top margin, and range radius of the indicators in the Option box. In the Color Range box, specify the minimum and maximum values, colors, and names for the value ranges. Click or to add or delete a value range. In the Others box, specify the name and color for values that do not fall into any of the defined ranges. If you want to use images to represent the values, check **Show Image** and specify the image for each value range and the others value group respectively, the color setting of the values will then be ignored.
3. In the Border tab, set the style and border mode for the indicators, including the style, line style, thickness, color, and transparency. All the border options can only take effect when the indicators are represented by colors.
4. In the Data Label tab, set the font formats and effects of the data labels.
5. In the Behaviors tab (this tab is only available for library component), select a trigger event from the drop-down list in the Events column, then click in the Actions column and click that appears in the text box to open the Web Action List dialog, where you can bind a web action to the indicators which will be triggered when the specified event occurs on the indicators.

   To add a web behavior line, click , and if a web behavior is not required, click to remove it.

   Click or to adjust the order of the behaviors. Then, when an event that has been bound with more than one action happens, the upper action will be triggered first.

   The web actions you can bind to the indicators include Filter, Sort, Parameter, Property and SendMessage. For details about these web actions, refer to Applying web actions to a label.
6. When done, click **OK** to apply the settings.

See also the Format Indicator dialog for report or library component for detailed explanation about options in the dialog.

•

Formatting the gauge

A gauge chart can be formatted to take some special appearances. The formatting options vary with the subtypes: Gauge Dial 2-D, Gauge Bar 2-D, and Gauge Bubble 2-D.

To format a dial gauge chart:

1. Right-click any pointer in a dial of the dial gauge chart, and then click **Format Dial Gauge** on the shortcut menu to bring up the Format Dial Gauge dialog.
2. In the General tab, specify the general options for the dials in the chart.

   In the Circular box, set the start angle and stop angle of the dials, the dial curve style, the width of the dials, and the relative size of a dial. In the Pointer box, set the pointer style and arrow style. In the Color Range box, specify whether or not to fill the dial with colors, and if to fill, set the minimum and maximum values, color, and name for the three predefined ranges in each dial.
3. In the Fill tab, click the color image button and select a color swatch to fill the selected pointers in the same data series, and specify the transparency of the color schema (you can also directly type the hexadecimal value of the desired color in the text box). If the color swatches cannot meet your requirement, click **More Colors** on the color palette to open the Pick a Color dialog, with which you can set a full range of color, or click **More Fill Effects** to specify the fill effect in the Fill Effects dialog. You can also click the **Color List** button to specify the color pattern for pointers in the same data series respectively in the Color List dialog.
4. In the Border tab, set the border mode for the pointers, including the border style, color, transparency, line style, thickness, end caps style, and line joint mode. You can also specify whether the border is to be outlined, and whether or not to automatically resize the border dashes if the border is drawn with dashes.
5. In the Data Label tab, set the font formats and effects of the data labels.
6. In the Behaviors tab (this tab is only available to library components), select a trigger event from the drop-down list in the Events column, then click in the Actions column and click that appears in the text box to open the Web Action List dialog, where you can bind a web action to the dial gauges which will be triggered when the specified event occurs on the dial gauges.

   To add a web behavior line, click , and if a web behavior is not required, click to remove it.

   Click or to adjust the order of the behaviors. Then, when an event that has been bound with more than one action happens, the upper action will be triggered first.

   The web actions you can bind to the dial gauges include Filter, Sort, Parameter, Property and SendMessage. For details about these web actions, refer to Applying web actions to a label.
7. When done, click **OK** to apply the settings.

To format a bar gauge chart:

1. Right-click any data marks in a scale of the bar gauge chart, and then click **Format Bar Gauge** on the shortcut menu to bring up the Format Bar Gauge dialog.
2. In the General tab, set the general options for the bar gauge chart.

   In the Option box, decide the layout of the scales in the gauge chart. In the Range Border box, set the color, style, transparency, and thickness of the scale borders. In the Color Range box, specify the minimum and maximum values, color, and name for the three predefined ranges in each scale.
3. In the Fill tab, click the color image button and select a color swatch to fill the selected data marks in the same data series, and specify the transparency of the color schema (you can also directly type the hexadecimal value of the desired color in the text box). If the color swatches cannot meet your requirement, click **More Colors** to open the Pick a Color dialog, with which you can set a full range of color, or click **More Fill Effects** to specify the fill effect in the Fill Effects dialog. You can also click the **Color List** button to specify the color pattern for data marks in the same data series respectively in the Color List dialog.
4. In the Border tab, set the border mode for the data marks, including the border style, color, transparency, line style, thickness, end caps style, and line joint mode. You can also specify whether the border is to be outlined, and whether or not to automatically resize the border dashes if the border is drawn with dashes.
5. In the Data Label tab, set the font formats and effects of the data labels.
6. In the Behaviors tab (this tab is only available for library component), select a trigger event from the drop-down list in the Events column, then click in the Actions column and click that appears in the text box to open the Web Action List dialog, where you can bind a web action to the bar gauges which will be triggered when the specified event occurs on the bar gauges.

   To add a web behavior line, click , and if a web behavior is not required, click to remove it.

   Click or to adjust the order of the behaviors. Then, when an event that has been bound with more than one action happens, the upper action will be triggered first.

   The web actions you can bind to the bar gauges include Filter, Sort, Parameter, Property and SendMessage. For details about these web actions, refer to Applying web actions to a label.
7. When done, click **OK** to apply the settings.

To format a bubble gauge chart:

1. Right-click any bubble in the bubble gauge chart, and then click **Format Bubble Gauge** on the shortcut menu to bring up the Format Bubble Gauge dialog.
2. In the General tab, set the general options for the bubbles in the chart.

   In the Option box, set the left margin, top margin, relative radius of the bubbles, and whether or not to draw horizontal/vertical grids. In the Value box, specifies the minimum and maximum values for the color range. The values will be equally divided into three ranges, each of which will be filled with the color you specify in the Color Range box automatically. Values beyond the minimum and maximum will be filled with grey color by default. You can also specify the minimum and maximum values for each color range as you want.
3. In the Border tab, set the border mode for the bubbles, including the border style, color, transparency, line style, thickness, end caps style, and line joint mode. You can also specify whether the border is to be outlined, and whether or not to automatically resize the border dashes if the border is drawn with dashes.
4. In the Data Label tab, set the font formats and effects of the data labels.
5. In the Behaviors tab (this tab is only available for library component), select a trigger event from the drop-down list in the Events column, then click in the Actions column and click that appears in the text box to open the Web Action List dialog, where you can bind a web action to the bubble gauges which will be triggered when the specified event occurs on the bubble gauges.

   To add a web behavior line, click , and if a web behavior is not required, click to remove it.

   Click or to adjust the order of the behaviors. Then, when an event that has been bound with more than one action happens, the upper action will be triggered first.

   The web actions you can bind to the bubble gauges include Filter, Sort, Parameter, Property and SendMessage. For details about these web actions, refer to Applying web actions to a label.
6. When done, click **OK** to apply the settings.

See also the following dialogs for detailed explanation about options in each dialog:

- Format Dial Gauge dialog for report
- Format Dial Gauge dialog for library component
- Format Bar Gauge dialog for report
- Format Bar Gauge dialog for library component
- Format Bubble Gauge dialog for report
- Format Bubble Gauge dialog for library component

•

Formatting the surface

To format the surface of a surface chart, follow the steps below:

1. Right-click any node in the surface, and then click **Format Surface** on the shortcut menu to bring up the Format Surface dialog.

2. In the Fill tab, click the color image button and select a color swatch to fill the selected surface in the same data series, and specify transparency of the color schema (you can also directly type the hexadecimal value of the desired color in the text box). If the color swatches cannot meet your requirement, click **More Colors** on the color palette to open the Pick a Color dialog, with which you can set a full range of color, or click **More Fill Effects** to specify the fill effect in the Fill Effects dialog. You can also click the **Color List** button to specify the color pattern for surfaces in the same data series respectively in the Color List dialog.

3. In the Border tab, set the border mode for the surfaces, including the border style, color, transparency, line style, thickness, end caps style, and line joint mode. You can also specify whether the border is to be outlined, and whether or not to automatically resize the border dashes if the border is drawn with dashes.

4. In the Data Label tab, set the font formats and effects of the data labels.

5. When done, click **OK** to apply the settings.

•

Formatting the scatter

To format the lines of a scatter chart, follow the steps below:

1. Right-click any node in the scatter chart, and then click **Format Scatter** on the shortcut menu to bring up the Format Scatter dialog.

2. In the General tab, set the general options for the scatter chart.

   In the Layout box, specify the layout of the chart. In the Line box, specify the thickness of the lines that connect the nodes if you have set the layout to Straight Line or Curved Line. In the Node box, set the node style, width and height as required.

3. In the Fill tab, click the color image button and select a color swatch to fill the selected scatter markers in the same data series, and specify transparency of the color schema (you can also directly type the hexadecimal value of the desired color in the text box). If the color swatches cannot meet your requirement, click **More Colors** on the color palette to open the Pick a Color dialog, with which you can set a full range of color, or click **More Fill Effects** to specify the fill effect in the Fill Effects dialog. You can also click the **Color List** button to specify the color pattern for scatter markers in the same data series respectively in the Color List dialog.

4. In the Border tab, set the border mode for the scatter markers, including the border style, color, transparency, line style, thickness, end caps style, and line joint mode. You can also specify whether the border is to be outlined, and whether or not to automatically resize the border dashes if the border is drawn with dashes.

5. In the Data Label tab, set the font formats and effects of the data labels.

6. Click **OK** to apply the settings and close the dialog.

•

Formatting the bubbles

To format the bubbles in a bubble chart, follow the steps below:

1. Right-click any bubble in the bubble chart, and then click **Format Bubble** on the shortcut menu to bring up the Format Bubble dialog.

2. In the General tab, specify whether or not to apply a 3D visual effect to the bubbles, and you can use the option Cut Bubble Based on XY Area to control whether or not to cut the bubbles when they are beyond the area formed by the X axis and Y axis.

3. In the Fill tab, click the color image button and select a color swatch to fill all bubbles if the chart has no series field or the selected bubbles in the same data series, and specify the transparency of the color schema (you can also directly type the hexadecimal value of the desired color in the text box). If the color swatches cannot meet your requirement, click **More Colors** on the color palette to open the Pick a Color dialog, with which you can set a full range of color, or click **More Fill Effects** to specify the fill effect in the Fill Effects dialog. You can also click the **Color List** button to specify the color pattern for bubbles in the same data series respectively in the Color List dialog.

4. In the Border tab, set the border mode for the bubbles, including the border style, color, transparency, line style, thickness, end caps style, and line joint mode. You can also specify whether the border is to be outlined, and whether or not to automatically resize the border dashes if the border is drawn with dashes.

5. In the Data Label tab, set the font formats and effects of the data labels.

6. When done, click **OK** to apply the changes.

•

Formatting the stock

To format the lines of a stock chart, follow the steps below:

1. Right-click any stock line in the stock chart, and then click **Format Stock** on the shortcut menu to bring up the Format Stock dialog.

2. In the General tab, set the color and thickness of the stock lines. If the chart is a Open-High-Low-Close stock chart, you can specify the up color to indicate that the opening price is lower than the closing price, the down color to indicate that the opening price is higher than the closing price, and the width of the stock bars.

3. In the Border tab, set the border mode for the stock bars, including the border style, color, transparency, line style, thickness, end caps style, and line joint mode. You can also specify whether the border is to be outlined, and whether or not to automatically resize the border dashes if the border is drawn with dashes.

4. In the Data Label tab, set the font formats and effects of the data labels.

5. When done, click **OK** to apply the changes and return to the report.

Formatting the bullets

To format the bullets in a bullet chart, follow the steps below:

1. Right-click any bullet in the bullet chart, and then select **Format Bullet** from the shortcut menu to bring up the Format Bullet dialog.
2. In the General tab, specify the size of the bullets by setting the width of featured measure, comparative measure and qualitative ranges respectively.
3. In the Fill tab, specify the fill properties of the bullets.
   - In the Featured Measure Color List box, click the color image button and select a color swatch to fill all the featured measures, and specify transparency of the color schema (you can also directly type the hexadecimal value of the desired color in the text box). If the color swatches cannot meet your requirement, click **More Colors** on the color palette to open the Pick a Color dialog, with which you can set a full range of color, or click **More Fill Effects** to specify the fill effect in the Fill Effects dialog. If you want to make the featured measure colors vary, check **Vary Colors by Color List**, then click the **Color List** button to specify the color pattern for each featured measure in the Color List dialog.
   - In the Comparative Measure Color List, click the color image button and select a color swatch to fill the selected comparative measures in the same data series, and specify transparency of the color schema. You can also click the **Color List** button to specify the color pattern for comparative measures in the same data series respectively in the Color List dialog.
   - In the Qualitative Ranges Color List box, click the color image button and select a color swatch to fill the selected qualitative ranges in the same data series, and specify transparency of the color schema. You can also click the **Color List** button to specify the color pattern for qualitative ranges in the same data series respectively in the Color List dialog.
4. In the Border tab, set the border mode for the bullets, including the border style, color, transparency, line style, thickness, end caps style, and line joint mode. You can also specify whether the border is to be outlined, and whether or not to automatically resize the border dashes if the border is drawn with dashes.
5. In the Data Label tab, specify the font formats and effects of the data labels.
6. In the Behaviors tab (this tab is only available to library components), select a trigger event from the drop-down list in the Events column, then click in the Actions column and click ... that appears in the text box to open the Web Action List dialog, where you can bind a web action to the bullets which will be triggered when the specified event occurs on the bullets.

   To add a web behavior line, click +, and if a web behavior is not required, click 🗑 to remove it.

   Click ▲ or ▼ to adjust the order of the behaviors. Then, when an event that has been bound with more than one action happens, the upper action will be triggered first.

   The web actions you can bind to the bullets include Filter, Sort, Parameter, Property and SendMessage. For details about these web actions, refer to Applying web actions to a label.
7. When done, click **OK** to accept the changes and close the dialog.

See also the Format Bullet dialog for report or library component for detailed explanation about options in the dialog.

•

Formatting the rectangles

To format the rectangles in a heat map, follow the steps below:

1. When there is only one group in the heat map, right-click on the heat map and select **Format Rectangle** from the shortcut menu to bring up the Format Rectangle dialog.

   When there are two or more groups in the heat map, right-click on the heat map, then from the Format Rectangle submenu, select a desired group field to bring up the Format Rectangle dialog for the field.

2. In the Fill tab, specify the fill type: Use Single Color, or Use Single Color with Condition.

   If you select Use Single Color as the fill type,

   - For discrete colors when color by one of the following cases:
     - No field
     - One or more groups
     - One or more groups and one summary

     In the Color section, click the color box and select a color swatch to fill all the rectangles (you can also directly type the hexadecimal value of the desired color in the text box). If the color swatches cannot meet your requirement, click **More Colors** on the color palette to open the Pick a Color dialog, with which you can set a full range of color, or click **More Fill Effects** to specify the fill effect in the Fill Effects dialog. Then specify the transparency if required.

     If you want to make the rectangle colors vary, check **Vary Colors by Color List**, then click the **Color List** button to specify the color pattern for each rectangle in the Color List dialog.

   - For gradient colors when color by a summary, specify the start color for the minimum value and the end color for the maximum value, by clicking the corresponding color box to select a color or by directly typing the hexadecimal value of the color in the text box.

     To exchange the start color and the end color, check the **Reversed** option.

     To specify a color for the zero value, check the Color When Zero option and then click the color box to select a color or type a color string in the text box after #.

     To specify a color when the value is null, click the color box to select a color or type a color string in the text box after #.

   If you select Use Single Color with Condition as the fill type, specify the conditions and the color pattern bound with each condition respectively. For details, refer to Adding conditional color fill to a chart.

3. In the Separator tab, specify the color, transparency, line style, and thickness of the rectangle separators.

4. In the Behaviors tab (this tab is only available to library components), select a trigger event from the drop-down list in the Events column, then click in the Actions column and click ... that appears in the text box to open the Web Action List dialog, where you can bind a web action to the rectangles which will be triggered when the specified event occurs on the rectangles.

   To add a web behavior line, click +, and if a web behavior is not required, click 🗑 to remove it.

   Click ▲ or ▼ to adjust the order of the behaviors. Then, when an event that has been bound with more than one action happens, the upper action will be triggered first.

   The web actions you can bind to the rectangles include Filter, Sort, Parameter, Property and SendMessage. For details about these web actions, refer to Applying web actions to a label.

5. When done, click **OK** to apply the settings and close the dialog.

See also the Format Rectangle dialog for report or library component for detailed explanation about options in the dialog.

•

Formatting the rectangle titles

To format the rectangle titles in a heat map, follow the steps below:

1. When there is only one group in the heat map, right-click on the heat map and select **Format Rectangle Title** from the shortcut menu to bring up the Format Rectangle Title dialog.

   When there are two or more groups in the heat map, right-click on the heat map, then from the Format Rectangle Title submenu, select a desired group field to bring up the Format Rectangle Title dialog for the field.

2. In the General tab, specify the height, fill color, and the font color of the rectangle titles.

3. In the Font tab, set the font settings for the titles.

4. In the Border tab, set the border color, thickness, line styles, and whether the borders have shadow effect and the shadow color if yes.

5. When done, click **OK** to apply the settings and close the dialog.

•html>

Adding conditional color fill to a chart

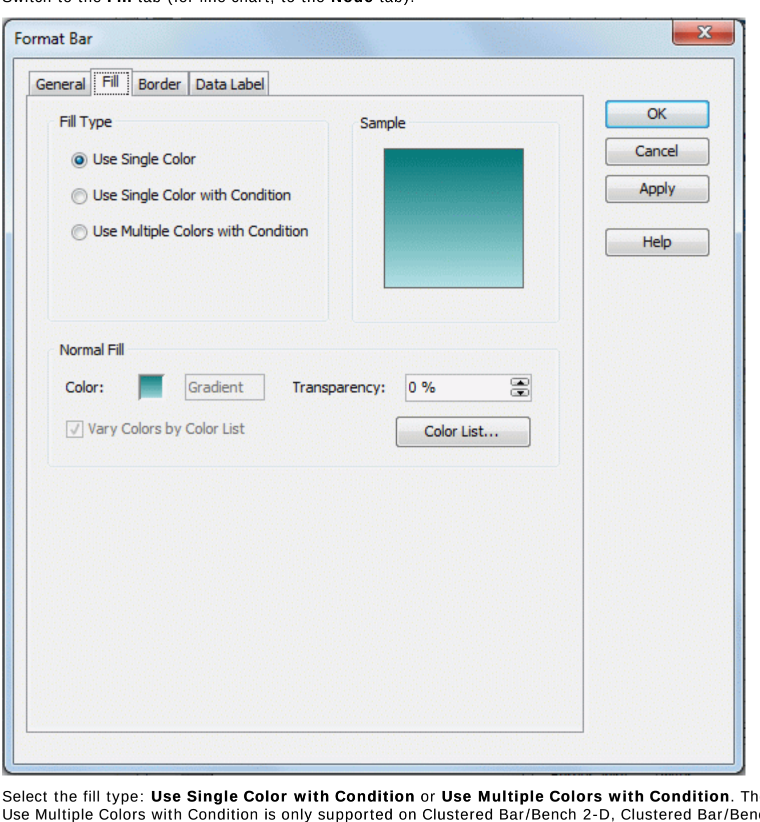
Usually, for a data marker in a chart, it uses a predefined color pattern assigned by JReport. However, by using the Conditional Color Fill feature, you can assign color patterns based on different value ranges. Thus, with the distinguishing colors, it is easy for you to locate the values you need to focus on in a chart.

There are two types of conditional color fill settings: Single Color with Condition and Multiple Colors with Condition. With a single color, you can make the data marker that meets the specified condition apply the color pattern bound with the condition. By using multiple colors, you can divide each data marker into different parts based on different value ranges along the direction of the value axis, and then specify different conditional colors to different value ranges.

Conditional color fill is only supported on bar, bench, pie, donut, area (Area 2-D and Area 3-D types), and 2-D line charts.

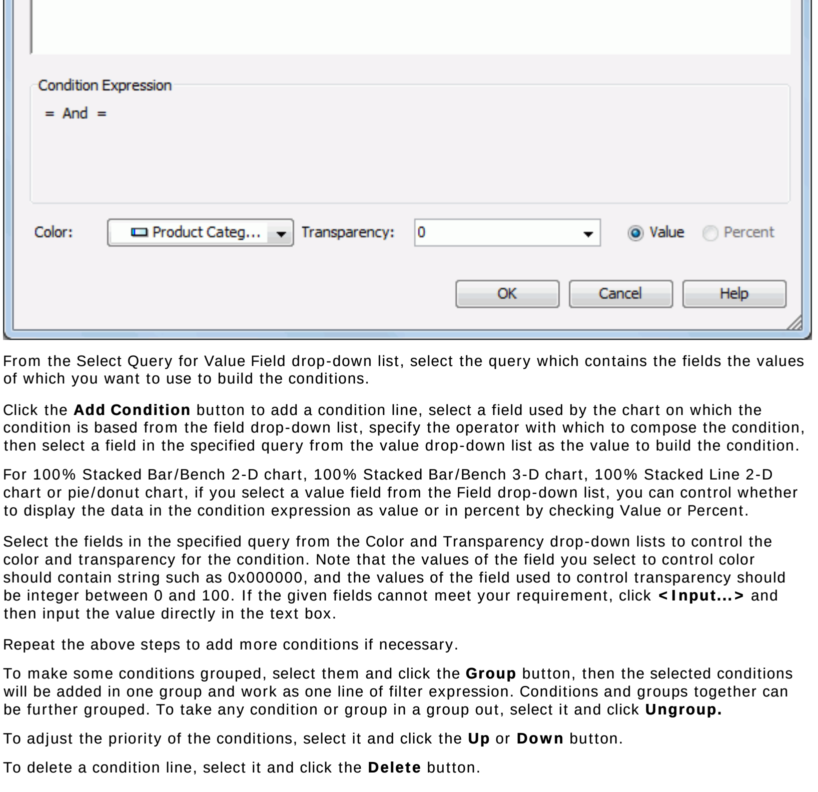
To add conditional color fill to a chart:

1. Double-click on any data marker (bar, bench, pie, donut, line, or heat map), and the corresponding format dialog is displayed (for heat map, it is the Format Rectangle dialog) .
2. Switch to the **Fill** tab (for line chart, to the **Node** tab).



3. Select the fill type: **Use Single Color with Condition** or **Use Multiple Colors with Condition**. The fill type Use Multiple Colors with Condition is only supported on Clustered Bar/Bench 2-D, Clustered Bar/Bench 3-D, Bar/Bench 3-D, Area 2-D and Area 3-D chart types.
4. If you select Use Single Color with Condition, you can edit the conditional fill with two modes (to switch between the two modes, click the **Advanced** or **Normal** button):
   - **Normal Mode**
     The procedure for editing conditional fill varies in normal mode according to the field on which the conditions will be based: category/series field or field whose return value is not a number; value field or field that returns a number.
     - If you want to define the conditions based on a category or series field or a field whose return value is not a number:
       a. Select the category/series field or a field whose return value is not a number from the Select Field drop-down list.
       b. Click to add a condition line, select a value from the Values drop-down list or input it directly in the text box to define the condition, then set the color and transparency for the value.
       c. Repeat the above step to add more conditional fill.
       d. The values beyond the ranges you have defined in all your conditions will automatically use the Other condition line. You can edit the color and transparency of the Other condition as well.
       e. By default, the condition expression of each condition line will be shown as the legend entry label. If you want to change it, select the condition line, check **Label**, and then change the label in the text box that follows.
     - If you want to define the conditions based on a value field or a field that returns a number:
       a. Select the value field or a field that returns a number from the Select Field drop-down list.
       b. Click to add a condition line, specify the start value and end value for the condition by inputting the values directly in the text boxes, then set the color and transparency for the value range.

          For 100% Stacked Bar/Bench 2-D chart, 100% Stacked Bar/Bench 3-D chart, 100% Stacked Line 2-D chart or pie/donut chart, you can control to display the data in the condition expression in percent. To achieve it, check **Percent**, and then input the percent value in decimal fraction in the Start Value and End Value column.
       c. Repeat the above step to add more conditional fill.
       d. The value beyond the ranges you have defined in all your conditions will automatically use the Other condition line. You can edit the color and transparency of the Other condition as well.
       e. By default, the condition expression of each condition line will be shown as the legend entry label. If you want to change it, select the condition line, check **Label**, and then change the label in the text box that follows.
   - **Advanced Mode**
     a. Click to open the Edit Conditions dialog.



     b. From the Select Query for Value Field drop-down list, select the query which contains the fields the values of which you want to use to build the conditions.
     c. Click the **Add Condition** button to add a condition line, select a field used by the chart on which the condition is based from the field drop-down list, specify the operator with which to compose the condition, then select a field in the specified query from the value drop-down list as the value to build the condition.

        For 100% Stacked Bar/Bench 2-D chart, 100% Stacked Bar/Bench 3-D chart, 100% Stacked Line 2-D chart or pie/donut chart, if you select a value field from the Field drop-down list, you can control whether to display the data in the condition expression as value or in percent by checking Value or Percent.
     d. Select the fields in the specified query from the Color and Transparency drop-down lists to control the color and transparency for the condition. Note that the values of the field you select to control color should contain string such as 0x000000, and the values of the field used to control transparency should be integer between 0 and 100. If the given fields cannot meet your requirement, click **<Input...>** and then input the value directly in the text box.
     e. Repeat the above steps to add more conditions if necessary.

        To make some conditions grouped, select them and click the **Group** button, then the selected conditions will be added in one group and work as one line of filter expression. Conditions and groups together can be further grouped. To take any condition or group in a group out, select it and click **Ungroup.**

        To adjust the priority of the conditions, select it and click the **Up** or **Down** button.

        To delete a condition line, select it and click the **Delete** button.
     f. When done, click **OK**. The condition specified in the Edit Conditions dialog will be listed as a new condition line in the Conditional Fill box in the chart formatting dialog. Click if you want to further edit the condition.
     g. The value beyond the ranges you have defined in all your conditions will automatically use the Other condition line. You can edit the color and transparency of the Other condition as well.
     h. By default, the condition expression of the condition line will be shown as the legend entry label. If you want to change it, select the condition line, check **Label**, and then change the label in the text box that follows.

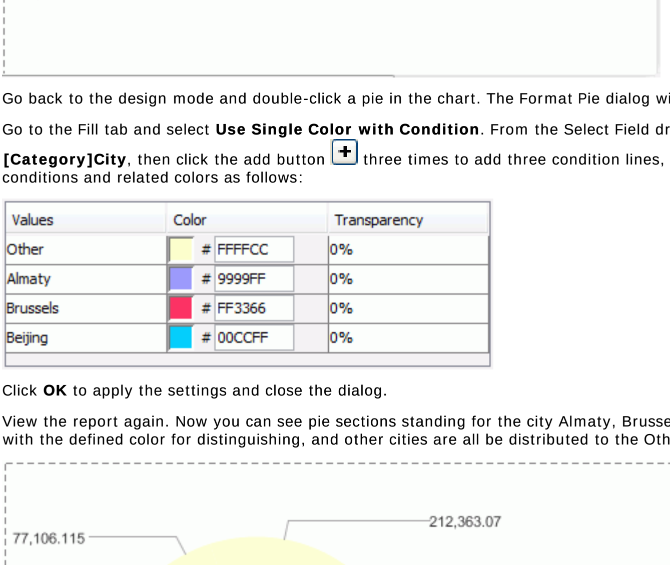
   If you select Use Multiple Colors with Condition, click button to add a condition line. Specify the start value and end value for the condition by inputting the values directly in the text boxes, then set the color and transparency for the value range. Click to add another condition and define the color pattern for it. The value beyond the ranges you have defined in all your conditions will automatically use color pattern defined in the Other condition line. By default, the condition expression of the condition line will be shown as the legend entry label. If you want to change it, select the condition line, check **Label**, and then change the label in the text box that follows.
5. When done, click **OK** to accept the settings and close the dialog.

Examples of adding conditional color fill to a chart

The following examples illustrate how to add conditional color fill to a chart.

Using the Use Single Color with Condition fill type

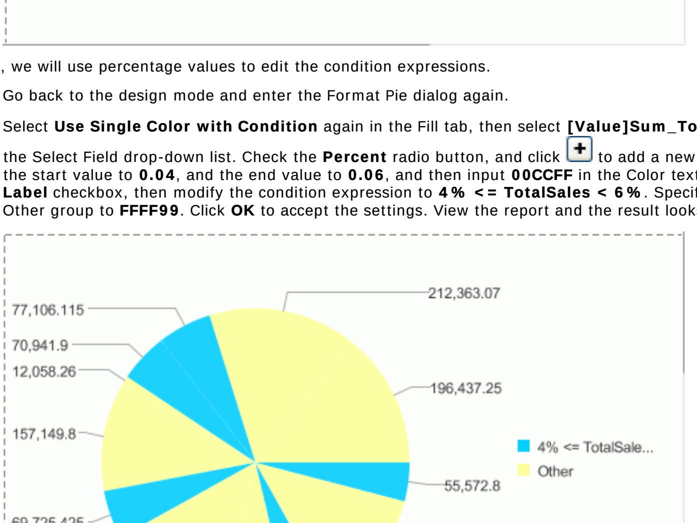
1. Open the catalog file SampleReports.cat.
2. Click **File** > **New** > **Page Report** on the menu bar. Select **Chart** in the New Page Report dialog and click **OK** to bring out the Chart Wizard.
3. In the Data screen of the wizard, select the query **WorldWideSales** from Data Source 1. Then click **Next**.
4. Select **Clustered Pie** type in the Type screen.
5. In the Display screen, add the field **City** to the Category box, then click the **Order/Select N** button. In the Category Options dialog, select **Top N** from the Select drop-down list, input **12** in the text field to the right, then click **OK** to return to the wizard. Add the field **Sum_TotalSalesbyCity** to the Show Values box.
6. Skip the Filter, Layout and Style screen, click **Finish** to create the chart.
7. Save the report and click the **View** tab, the report will show as follows:



8. Go back to the design mode and double-click a pie in the chart. The Format Pie dialog will be displayed.
9. Go to the Fill tab and select **Use Single Color with Condition**. From the Select Field drop-down list, select **[Category]City**, then click the add button three times to add three condition lines, and specify the conditions and related colors as follows:

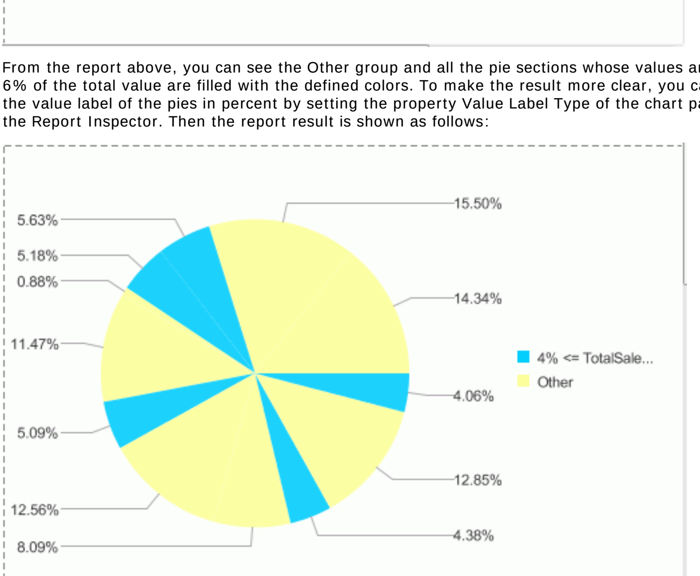
Values	Color	Transparency
Other	# FFFFCC	0%
Almaty	# 9999FF	0%
Brussels	# FF3366	0%
Beijing	# 00CCFF	0%

10. Click **OK** to apply the settings and close the dialog.
11. View the report again. Now you can see pie sections standing for the city Almaty, Brussels and Beijing are filled with the defined color for distinguishing, and other cities are all be distributed to the Other group.

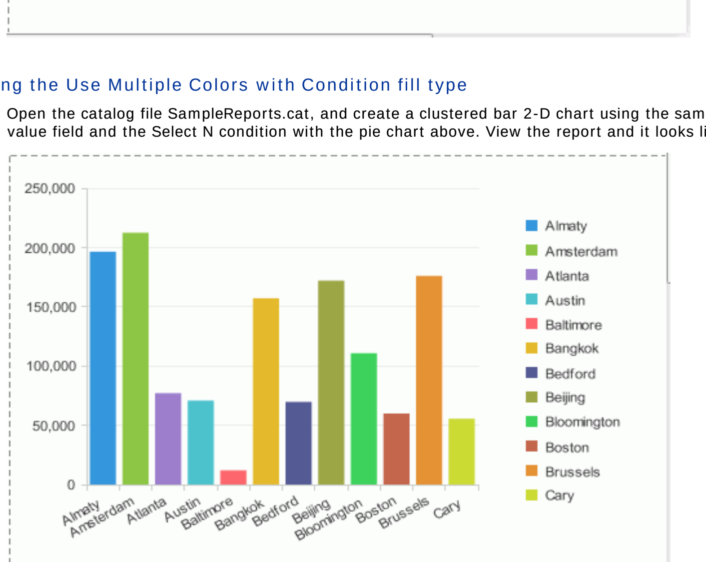


Next, we will use percentage values to edit the condition expressions.

12. Go back to the design mode and enter the Format Pie dialog again.
13. Select **Use Single Color with Condition** again in the Fill tab, then select **[Value]Sum_TotalSalesbyCity** from the Select Field drop-down list. Check the **Percent** radio button, and click to add a new condition line. Set the start value to **0.04**, and the end value to **0.06**, and then input **00CCFF** in the Color text field. Check the **Label** checkbox, then modify the condition expression to **4% <= TotalSales < 6%**. Specify the color for the Other group to **FFFF99**. Click **OK** to accept the settings. View the report and the result looks like:

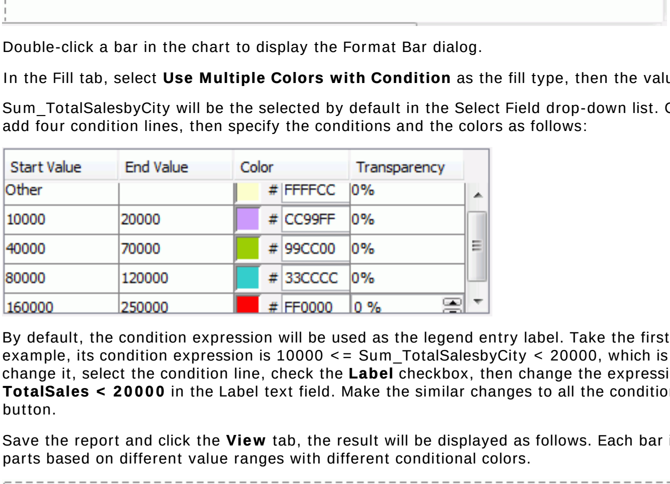


14. From the report above, you can see the Other group and all the pie sections whose values are between 4% and 6% of the total value are filled with the defined colors. To make the result more clear, you can specify to display the value label of the pies in percent by setting the property Value Label Type of the chart paper to percent in the Report Inspector. Then the report result is shown as follows:



Using the Use Multiple Colors with Condition fill type

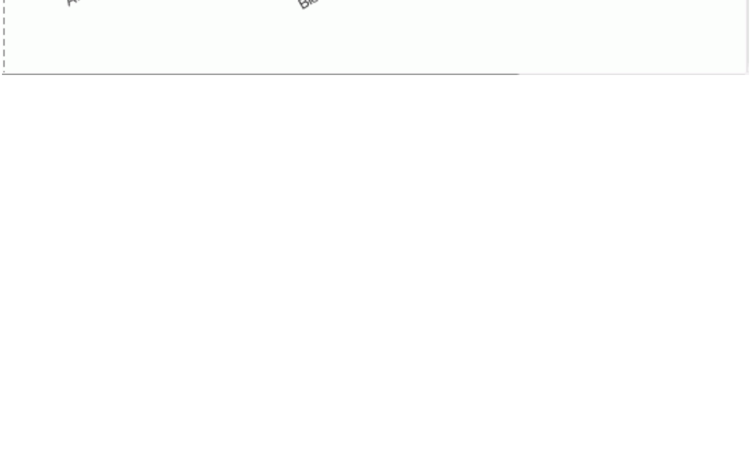
1. Open the catalog file SampleReports.cat, and create a clustered bar 2-D chart using the same query, category/value field and the Select N condition with the pie chart above. View the report and it looks like as follows:



2. Double-click a bar in the chart to display the Format Bar dialog.
3. In the Fill tab, select **Use Multiple Colors with Condition** as the fill type, then the value field Sum_TotalSalesbyCity will be the selected by default in the Select Field drop-down list. Click four times to add four condition lines, then specify the conditions and the colors as follows:

Start Value	End Value	Color	Transparency
Other		# FFFFCC	0%
10000	20000	# CC99FF	0%
40000	70000	# 99CC00	0%
80000	120000	# 33CCCC	0%
160000	250000	# FF0000	0 %

4. By default, the condition expression will be used as the legend entry label. Take the first condition line as example, its condition expression is 10000 <= Sum_TotalSalesbyCity < 20000, which is not very intuitive. To change it, select the condition line, check the **Label** checkbox, then change the expression to **10000 <= TotalSales < 20000** in the Label text field. Make the similar changes to all the condition lines and click the **OK** button.
5. Save the report and click the **View** tab, the result will be displayed as follows. Each bar is divided into several parts based on different value ranges with different conditional colors.



Binding a link to a chart

You can make a chart in a report linked with another report, a location specified by a URL, an e-mail address, or a Blob data type field. The link can either be a simple link or a conditional link. With conditional link, different targets can be loaded based on different conditions.

The links in a chart can be triggered from the following objects: the category (X) axis, the series (Z) axis, the legend, or the data markers.

To bind a link to a chart:

1. Right-click the trigger object from which to open the link, then click **Link** on the shortcut menu.
2. In the Insert Link dialog, specify whether it is a conditional link or a simple link, then select the link target and set the options for the link target as required.
3. When done, click **OK**.

For details about different link targets, see Binding a link to a label.

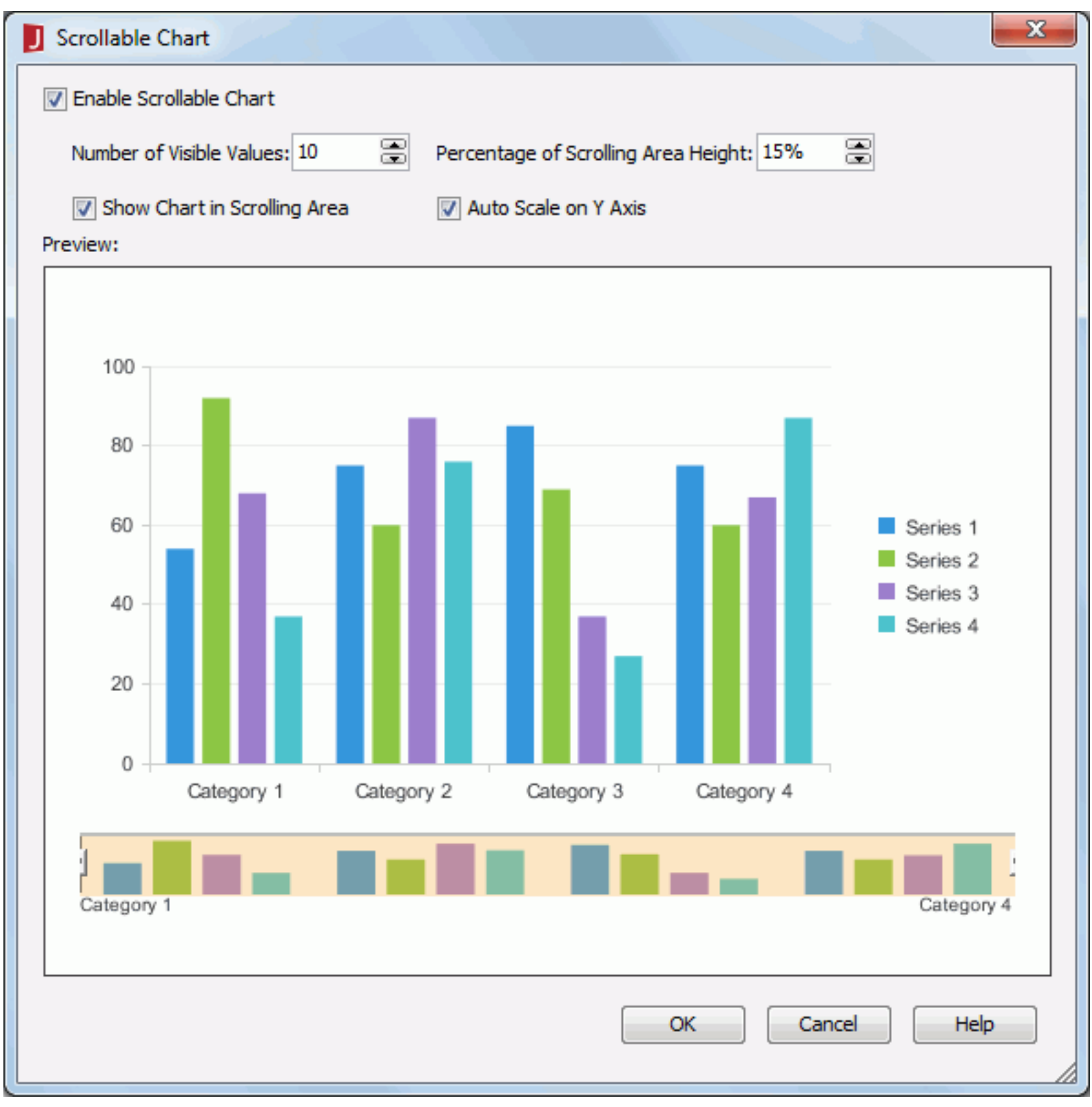
Creating a scrollable chart

In a scrollable chart, values displayed on the chart are not fixed. A scrollbar on the scrollable chart can help you to control the visible value range on the X axis of the chart. This is often used where the X axis is a date or a time such as stock quotes; however, it can be used with any type of value. As shown below, any time the number of categories may become too large to be easily identified on the screen, consider adding a scrollable option.

Currently, scrollable charts are supported on charts of bar, bench, line, and area types in web reports and library components only.

To make a chart of bar, bench, line, or area type scrollable:

1. Right-click anywhere of the chart except the chart object and select Scrollable Chart from the shortcut menu to display the Scrollable Chart dialog.

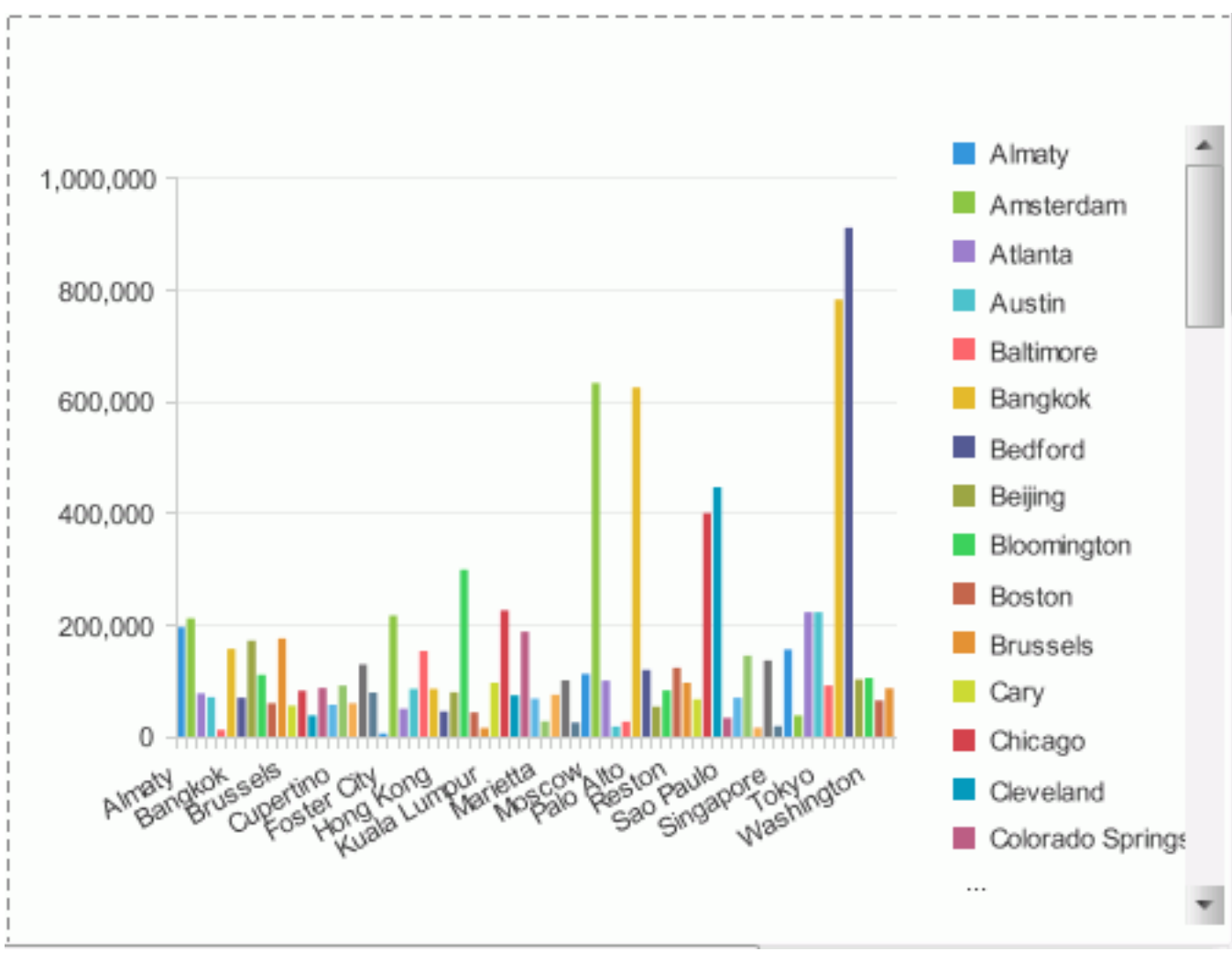


2. Check the **Enable Scrollable Chart** option to specify to use a scrollbar to control the number of visible values on the axis.
3. Specify how many data items will be selected on the scrollbar and displayed on the axis by default in the Number of Visible Values box, then specify the percentage the scrollbar occupies the whole size of the chart.
4. Specify whether to show the thumbnail chart in the scrollbar.
5. Specify whether to recalculate the scale of Value (Y) Axis when the the value of the option Number of Visible Values is changed.
6. Click **OK** to apply the settings for the chart.

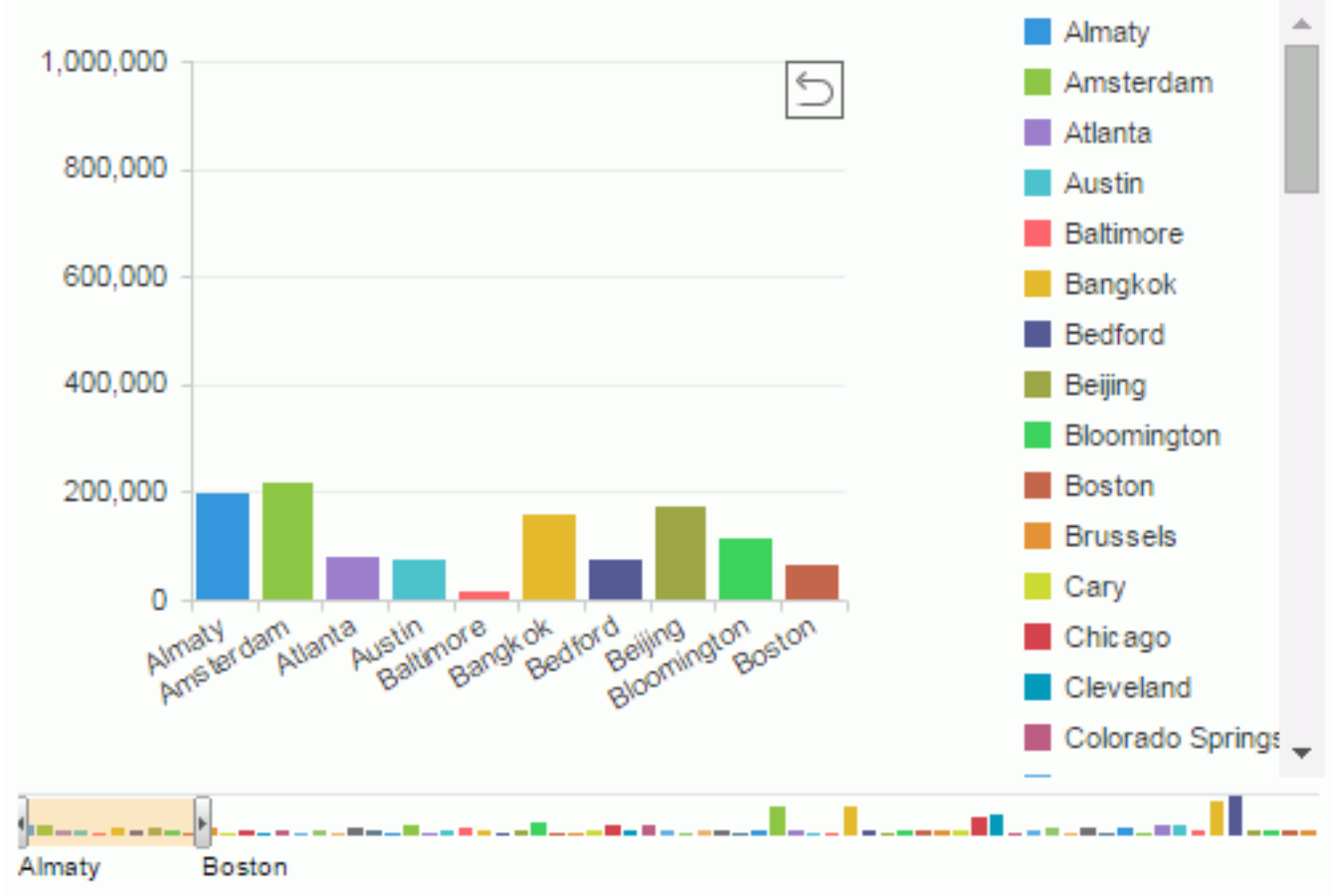
Example of creating a scrollable chart

The following example shows how to create a scrollable chart:

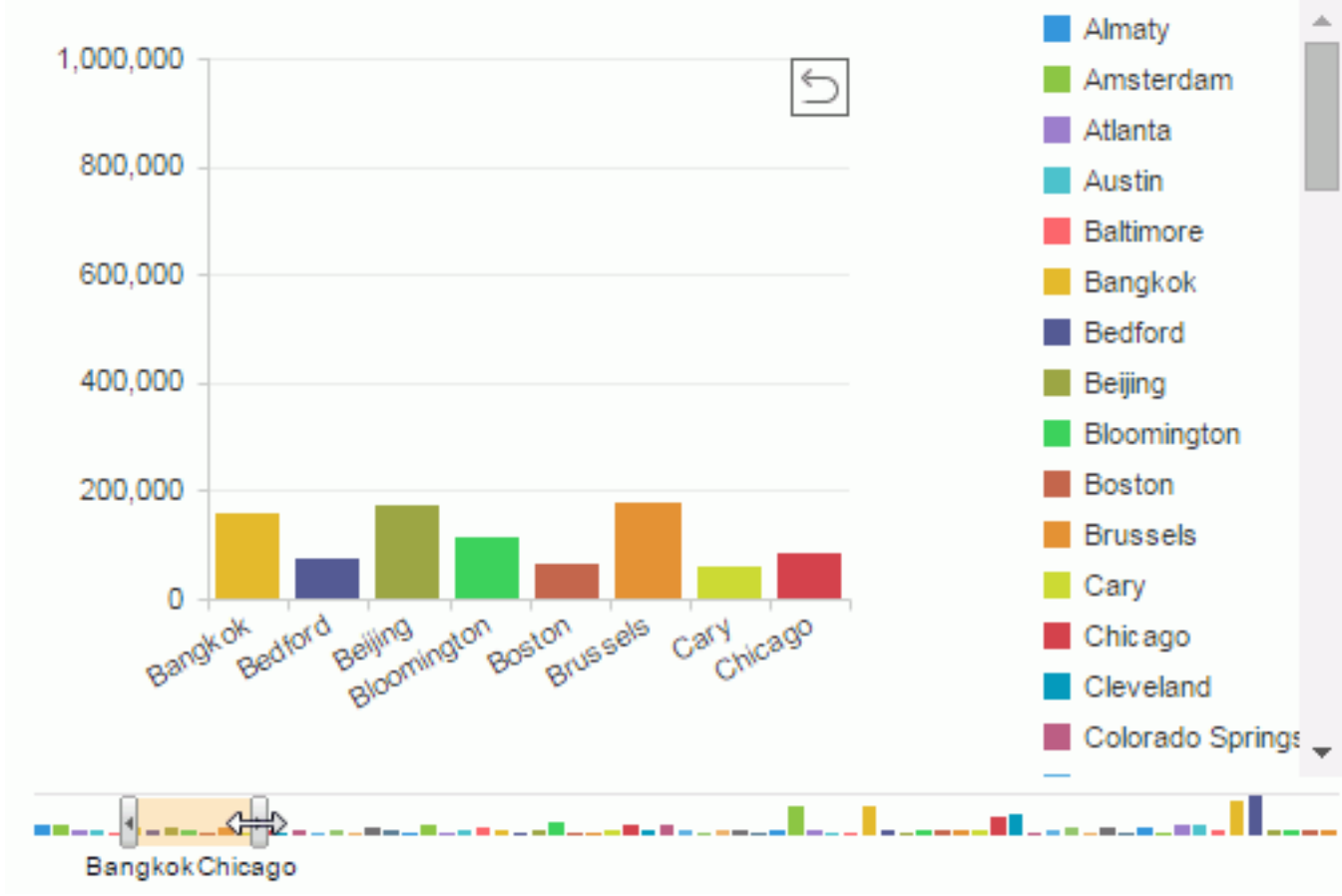
1. Open the catalog file JinfonetGourmetJava.cat in `<install_root>\Demo\Reports\TutorialReports`.
2. Click **File** > **New** > **Web Report** on the menu bar.
3. Drag **Chart** from the Toolbox panel to the web report to bring out the Create Chart wizard.
4. In the Data screen of the wizard, select **WorldWideSalesBV** from Data Source 1. Then click **Next**.
5. Select **Clustered Bar 2-D** type in the Type screen.
6. In the Display screen, add the field **City** to the Category box, and add **Total Sales** to the Show Values box.
7. Switch to the **Style** screen and select **Neutral** as the report style from the Style list.
8. Click **Finish** to create the scrollable chart.
9. Click the **View** tab to view this chart, and it appears as follows:



10. Click the **Design** tab to return to the design mode.
11. Right-click anywhere of the chart except the chart object and select **Scrollable Chart** from the shortcut menu. The Scrollable Chart dialog appears.
12. Check the **Enable Scrollable Chart** option, then set the option Number of Visible Values to **10** and Percentage of Scrolling Area Height to **10 %**. Then, click **OK** to apply the settings and leave the dialog.
13. The option Auto Scale on Y Axis is checked by default, which means the scale of Value (Y) axis will be recalculated when the the value of the option Number of Visible Values is changed.
14. Save the report and click **View** > **Preview As** > **Web Report Result** on the menu bar, the report will show as follows in Web Report Studio:



15. From the report result above, we can see only 10 total sales values from the city Almaty to Boston. To change the data items displayed on the X axis, drag the left or right handle or the area between the two handles on the scrollbar. See the result as follows:



Note: When you export a scrollable chart to any format, the scrolling function will not be available. In the exported results, only the values that are selected at the time when the report is exported will be displayed.

•

Creating a real time chart

A real time chart can be updated automatically at runtime by using data that changes rapidly with time such as stock market quotes. After you create a real time chart, when the data in the data source the chart uses is updated in real time, the chart will automatically fetch data from the data source to update itself with the interval you specify. Then, you will conveniently see the real time data information without updating the chart by yourself and without having to re-fetch all of the data. Only the new data since the last interval is selected by the query.

Before you create a real time chart, you have to make sure the data source you use contains real time data which can be updated periodically. Otherwise, the chart will lose its real time meaning.

Currently, real time charts are supported on web reports and library components only.

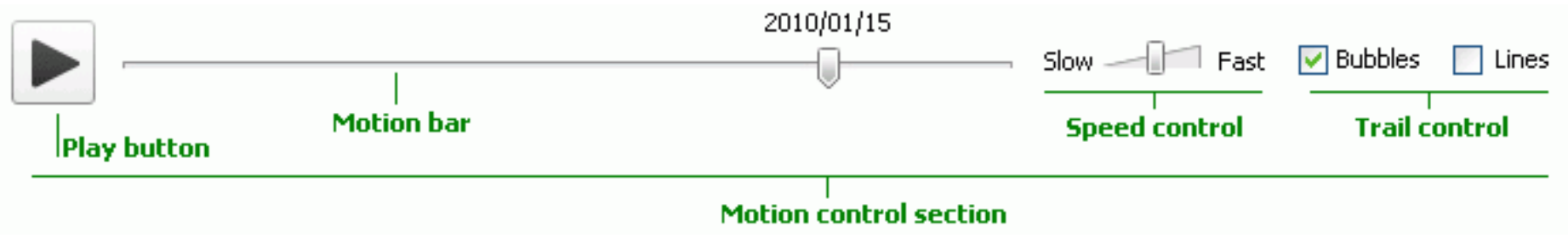
To insert a real time chart into a web report or a library component:

1. Position the mouse pointer at the destination where you want to insert the chart.
2. Do one of the following to open the Create Chart wizard:
   - Click **Insert** > **Chart** on the menu bar.
   - Click the **Insert Chart** button on the Insert toolbar.
   - Drag **Chart** from the Toolbox panel to the destination.
3. In the Data screen of the wizard, select a business view in the current catalog on which the chart will be created.
4. In the Type screen, specify the type of the chart which can only be single chart of bar, bench, line, or area type.
5. In the Display screen, specify a title for the chart in the Title text field. Check the **Real Time** checkbox.
6. By default, Use System Time for Category is checked and you can see the text Use System Refresh Time is displayed in the Category box, which means the time at which the chart refreshes itself will be used as the category value. You can uncheck the Use System Time for Category option and add another field to be displayed on the category (X) axis, and define the Order/Select N condition on the field if required. Then, add the fields of numeric type which cannot be summary fields to the Show Values box.
7. Specify the time interval at which the chart will get data and refresh itself automatically in the Refresh Interval text field.
8. Specify the most recent N records to be kept for the real time data on the chart in the Show Most Recent text field.
9. Click the **Incremental Fetch** button to add the fields you want to use as the unique key of the real time chart in the Unique Key dialog.

   Once a unique key is defined, each time when the real time chart automatically updates itself, duplicated data records will be filtered out based on the unique key. For instance, if you add the fields Country and Product ID as the unique key of a real time chart, when a record with the product ID 1 in USA has already been loaded into the chart, no more records of this product ID in USA will be added to the real time chart because they have the same unique key value.
10. To apply some filters to the chart so as to reduce the data displayed in the chart, go to the Filter screen and define the filter conditions.
11. In the Layout screen, specify the layout of the chart.
12. In the Style screen, select a style for the chart.
13. Click **Finish** to create the chart.

Creating a motion chart

You can make a chart move at runtime based on the value changes of a specified motion field. This kind of chart is called motion chart. For a motion chart, a motion control section will be added on its platform, which contains the following elements:



- A play button allowing you to make the chart play its dynamic trend or to be static.
- A motion bar which contains a defined motion field. The chart will update its dynamic trend based on the value changes of the motion field.
- A speed control. The moving speed of the chart can be controlled by dragging the slider on the speed control.
- For a bubble chart, there is also a trail control, which allows you to make the chart show trails of bubbles or lines (when null values appear, the trails of bubbles or lines will be unavailable).

Currently, motion charts are supported on web reports and library components only.

To insert a motion chart into a web report or a library component:

1. Position the mouse pointer at the destination where you want to insert the chart.
2. Do one of the following to open the Create Chart wizard:
   - Click **Insert** > **Chart** on the menu bar.
   - Click the **Insert Chart** button on the Insert toolbar.
   - Drag **Chart** from the Toolbox panel to the destination.
3. In the Data screen of the wizard, select a business view in the current catalog on which the chart will be created.
4. In the Type screen, specify the type of the chart which can only be single chart of bar, bench or bubble type.
5. In the Display screen, specify a title for the chart in the Title text field. Add the required fields from the Resources box to be displayed on the category axis, series axis and value axis of the chart, and define some special group, special function and Order/Select N condition on the category/series field if required. You can also add additional values including constant values and average values to be displayed on the value axis of the chart.

   If you select a bubble chart type, you need to specify the fields to be shown on its X axis, Y axis and the value you want to show as the bubble radius in the Show Values box. Note that when you specify a value for X axis, this value will be displayed on the category axis instead of the one specified in the Category box. However, the value defined in the Category box will be also be included in the data calculation.

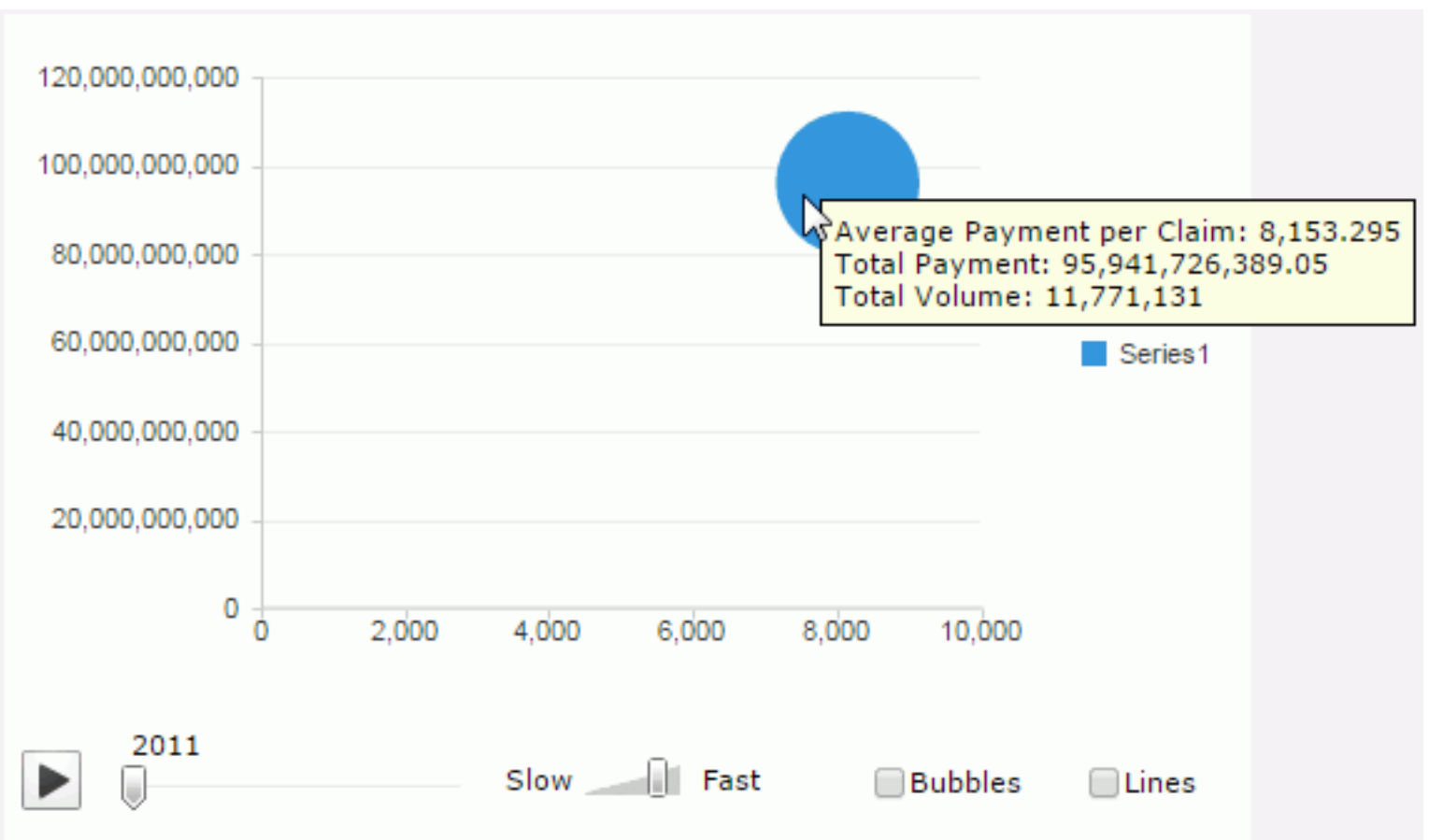
   Check the **Motion Bar for Playable Chart** checkbox, then from the Resources box add a field which is of Integer, Date or Time data type as the motion field. When the field is of the Date data type, you can define some special function on it.
6. To apply some filters to the chart so as to reduce the data displayed in the chart, go to the Filter screen and define the filter conditions.
7. In the Layout screen, specify the layout of the chart.
8. In the Style screen, select a style for the chart.
9. Click **Finish** to insert the chart.

When a motion chart is published to JReport Server, you can then run it in Web Report Studio or JDashboard. Then, you can use the motion control section to make the chart move. Click the play button and the chart will show its dynamic trend based on the value change of the motion field which is bound in the motion bar. To stop it, click the button again. You can also control its moving speed by dragging the slider between Slow and Fast on the speed control. For a bubble chart, you can control whether the chart will be moving in bubble or line trail.

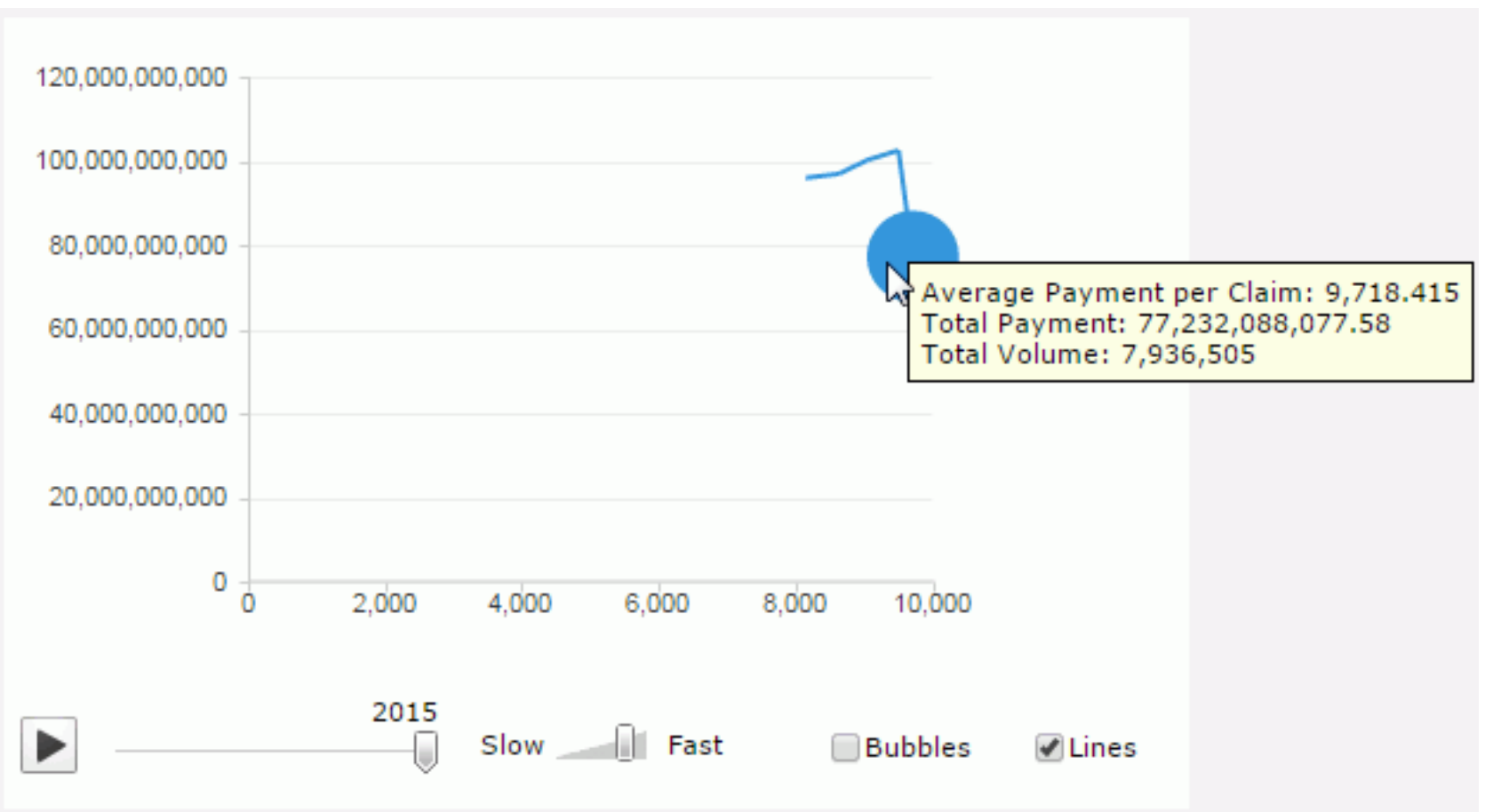
Example of creating a motion chart

This example illustrates how to create a motion chart in a web report and play it in Web Report Studio.

1. Open the catalog file SampleReports.cat.
2. Click **File** > **New** > **Web Report** on the menu bar.
3. Drag **Chart** from the Toolbox panel to the web report. The Create Chart wizard appears.
4. In the Data screen of the wizard, select **BubbleChartBV** from Data Source 1. Then click **Next**.
5. Select **Bubble 2-D** type in the Type screen.
6. In the Display screen, add the field **Category** to the Category box, then click the **Order/Select N** button. In the Category Options dialog, select **Top N** from the Select drop-down list, input **2** in the text field to the right, then click **OK** to return to the wizard.
7. In the Show Values box, add the field **Average Payment per Claim**, **Total Payment** and **Total Volume** respectively as the value of X Axis, Y Axis and Radius. Note that when you specify a value for X axis, this value is displayed as the category value instead of Category that is specified in the Category box. Although Category is displayed as the category value, it is also included in the data calculation.
8. Check **Motion Bar for Playable Chart**, add the field **Fiscal Year** as the motion field.
9. Click **Finish** to create the motion chart.
10. Save the report and click **View** > **Preview As** > **Web Report Result** on the menu bar, the report will show as follows:



11. Check the **Lines** checkbox, and click the play button, then the value of the motion field will change, and the bubbles in the chart will move based on the value change of the motion field in line trail. See the result below, when the value of the motion field is changed to 2015, the display value of the bubble also changes accordingly.



Note: To have a better and stable performance for motion charts, you are recommended to use the Google Chrome web browser.

Creating dynamic charts in Excel

You can make data in a chart map to a single cell in Excel, then when you change the data in a cell, the corresponding change will also be displayed in the chart. This kind of chart is called dynamic chart. To create a dynamic chart in Excel, the chart should use the same data as a banded object or table, which is exported to Excel together with the chart. That is to say, each data value in the chart must correspond to the one in the banded object or table. Moreover, the option Report Format should be selected when exporting the report to Excel.

To have the same data with a banded object or table, you must meet the following conditions for the chart before exporting the report to Excel:

- When the chart inherits data from the banded object or table:
  - The chart should have the same group structure as that of the banded object or table.

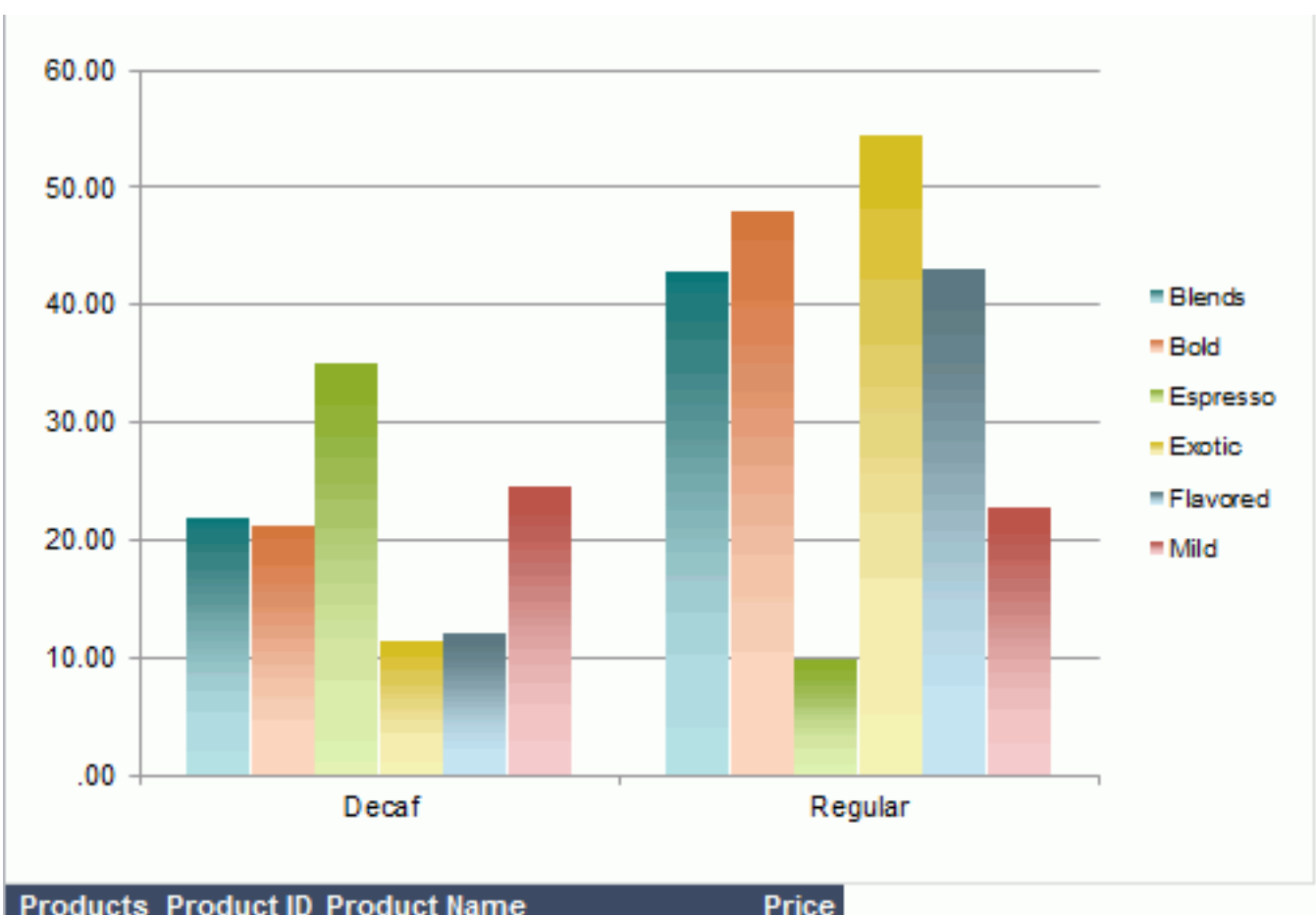
    The following example further explains what the same group structure means.

    A banded object has two groups: group 1 and group 2 (group 1 is the higher level)
    - When the chart is to be inserted into the banded header panel:
      - Add the summary 2 based on group 2 to the value axis, then group 2 will be added to the category axis automatically, and you need to add group 1 to the series axis manually.
      - Add the summary 1 based on group 1 to the value axis, then group 1 will be added to the category axis automatically, and no value need to be added to the series axis.
    - When the chart is to be inserted into the group header panel 1 which displays group 1, only summary 2 based on group 2 can be added to the value axis. Then group 2 will be added to the category axis, and no value will be added to the series axis.
    - When the chart is to be inserted into the group header panel 2 which displays group 2, the value axis cannot contain any summary value.
  - Different conditions apply to detail charts and summary charts:
    - A detail chart does not contain summary value on the value axis, but DBFields. The DBFields on the value axis and the value on the category axis must be in the detail panel of the banded object or table, and should not be hidden.
    - A summary chart contains summary value on the value axis, and the summary on the value axis must be in the banded object or table, and should not be hidden.
- When the chart does not inherit data from the banded object or table:
  - The chart should use the same dataset as the banded object or table.
  - The chart should have the same group structure as that of the banded object or table.
  - The chart should have the same filter/sort condition, Top N condition and special function as that of the banded object or table.
  - The chart can only be inserted into the banded header/footer panel or outside of the banded object.
  - Different conditions apply to detail charts and summary charts:
    - For a detail chart,
      - DBFields on the value axis must be in the detail panel of the banded object or table, and should not be hidden.
      - The category value must be in the detail panel of the banded object or table, and should not be hidden.
      - If the chart, banded object or table has a Top N condition, then the chart will not be mapped to the banded object or table.
    - For a summary chart, the summary on the value axis must be in the banded object or table, and should not be hidden.

The following example shows how to use the Dynamic Chart feature in Excel. Here the chart inherits data from its parent banded object.

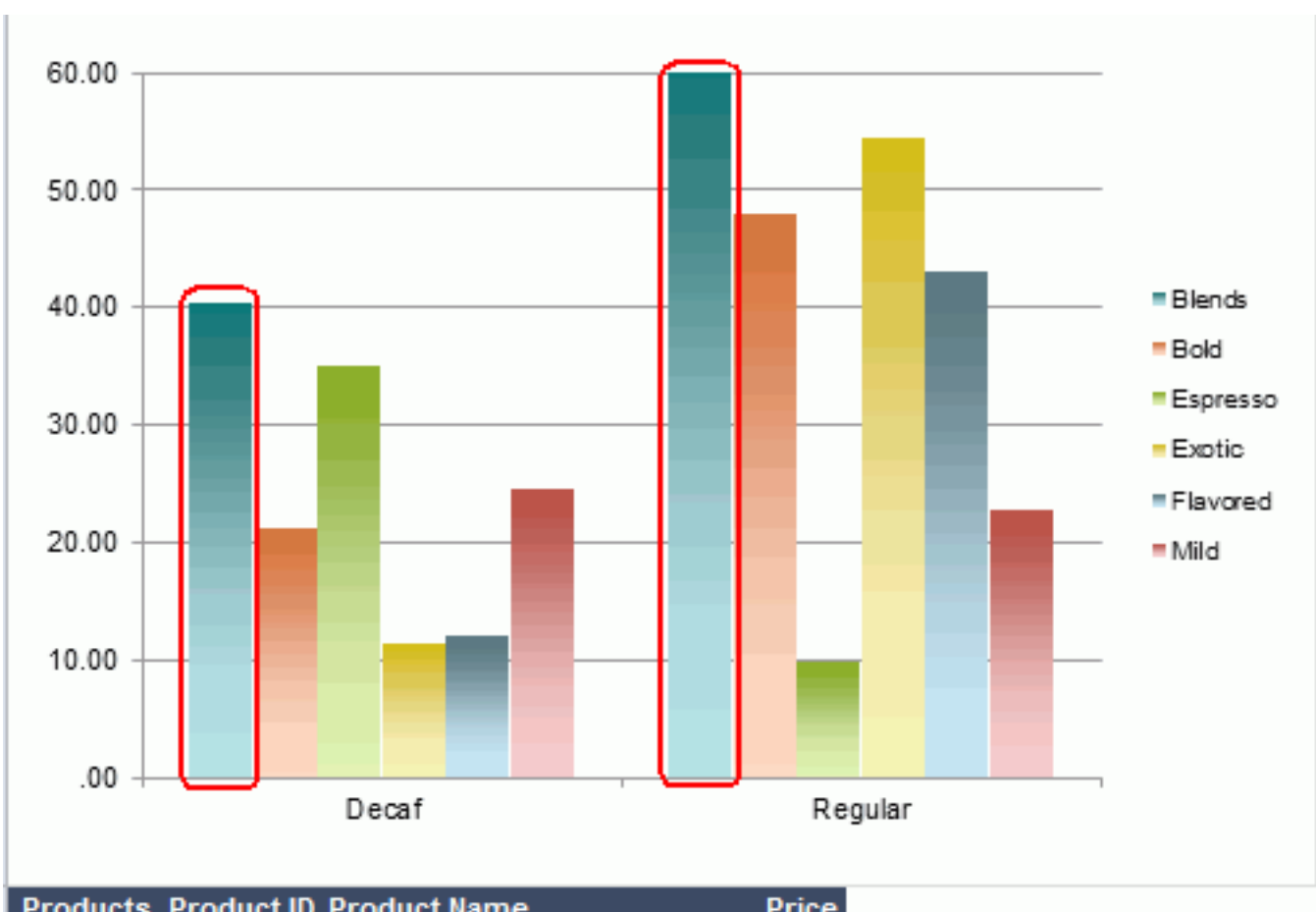
1. Open the catalog file JinfonetGourmetJava.cat in `<install_root>\Demo\Reports\TutorialReports`, and create a query Products in Data Source 1 of the catalog based on the table Products. Check all the columns in the table.
2. Create a standard banded report based on the query Products. The banded report displays the following fields: Products_Product ID, Product Name and Price, is grouped by Category and Product Type Name (Category is the higher level), and takes the Neutral report style.
3. In the Resource View panel, click **<Add Summary...>** from the Summaries node and create a summary named Sum_Price which uses the function Sum, sums on Price in the Products table, and applies to the Product Type Name group level.
4. Drag the summary to the footer panel of the Product Type Name group level in the banded object.
5. Click **Insert** > **Chart** to insert a Clustered Bar 2-D chart into the banded header panel. The chart inherits the dataset from the banded object (make sure Current is checked in the Data screen of the Create Chart wizard), displays Product Type Name on the category axis, Category on the series axis, and Sum_Price on the value axis.
6. Right-click the chart and select **Chart Wizard** from the shortcut menu.
7. In the Chart Wizard, switch to the **Layout** screen, select **Export** in the Options box and then select **BandedObject** from the Mapping Component drop-down list, then click **Finish**.
8. Click **File** > **Options**. In the Options dialog, check the option **Check the availability of dynamic chart for Excel** in the General category, then click **OK**.

   This option is provided to check if the chart can be correctly mapped to the banded object when you save the report or export it to Excel.
9. Click **File** > **Export To** > **Excel** to export the report to Excel (make sure Report Format is checked in the Export to Excel dialog).
10. Open the exported result file in `<install_root>\Demo\Reports\TutorialReports`.



Products_Product ID	Product Name	Price
Blends		
Decaf		
31	Gold Coast Blend	11.45
27	Kona Mountain	10.45
	Sum_Price	21.90
Regular		
29	Colombia El Tambo	12.00
28	Blue Mountain	9.95
24	Italian Roast	10.45
23	French Roast	10.45
	Sum_Price	42.85
Bold		
Decaf		
12	House Blend	9.95
8	Breakfast Blend	11.15

11. Change the price of Gold Coast Blend and Colombia El Tambo to **30.00**, and you will find that the value of the chart is changed accordingly.



Products_Product ID	Product Name	Price
Blends		
Decaf		
31	Gold Coast Blend	30.00
27	Kona Mountain	10.45
	Sum_Price	40.45
Regular		
29	Colombia El Tambo	30.00
28	Blue Mountain	9.95
24	Italian Roast	10.45
23	French Roast	10.45
	Sum_Price	60.85
Bold		
Decaf		
12	House Blend	9.95
8	Breakfast Blend	11.15

Notes:

- You cannot make charts of bubble, stock and scatter types to be dynamic charts if they have summary value on the value axis.
- Chart and the summary used in the chart cannot be inserted into the repeated panel of the banded object or table.
- For summary chart, only value axis can be mapped to each single cell in Excel. That is to say, when you change the data in Excel, the corresponding value axis will also be changed.

Tabulars

A tabular is a component designed to lay out other components. Unlike a table component, that is used to display a dataset, a tabular is a set of cells that can each display a dataset in a component.

Inserting a tabular in a report

A tabular can be inserted in the report areas listed in Component placement.

To insert a tabular into a report, use either of the following ways:

- **Using dialog**
  1. Do one of the following to bring out the Insert Tabular dialog:
     - Click **Insert** > **Tabular** on the menu bar.
     - Click the **Insert Tabular** button on the Insert toolbar.
  2. Specify the number of rows and columns, and the border width and tabular width as required.
  3. Click the **OK** button to close the dialog.

     If the mouse pointer is in the report body or a cell of another tabular, the tabular will be inserted there upon clicking OK. Otherwise, you need to click the mouse button in the desired location to place the tabular there.
- **By dragging and dropping**

  From the Toolbox panel, drag **Tabular** to the desired location in the report which allows the insertion of a tabular, and a tabular will then be created there.

After a tabular has been created in a report, you can then:

- Insert the required components into the tabular cells. The components that can be inserted into a tabular cell are listed in Component placement. If the component inserted is larger than the tabular cell, you can right-click the tabular cell and select **Auto Fit** from the shortcut menu to make the component fully displayed in the tabular cell.
- Double-click a tabular cell and then input and modify text in it.

You can also customize the tabular appearance and behavior by setting properties in the Report Inspector, which can be saved as a CSS style for future use (for details, see Creating a CSS style).

Merging and splitting tabular cells

In a tabular, adjacent cells that form a rectangle can be merged into one cell.

- To merge adjacent cells, select them, right-click and select **Merge** from the shortcut menu (or click the **Merge** button on the Standard toolbar). These cells will then be merged into one cell.
- To split a cell, follow the steps below:
  1. Right-click the cell and select **Split** from the shortcut menu.
  2. In the Split Cell dialog, check whether the cell will be split into rows or columns, and then specify how many rows/columns the cell will be split into.
  3. Click **OK**, and the cell will then be split according to your selection.

Resizing, inserting and deleting columns/rows in a tabular

- To resize a column/row of a tabular, drag the right/bottom boundary of the column/row until the column/row is the width/height you want. If the boundary is not the rightmost/last one, the column width/row height will change, but the total width/height of the tabular will not change. If you want to change both the column width/row height and the tabular width/height, press the Shift button on the keyboard while dragging.
- To insert a column/row into a tabular, select a cell and right-click it. On the shortcut menu, click the corresponding inserting column/row item and a column/row will then be inserted according to your selection.
- To delete a column/row from a tabular, select all the cells in this column/row, right-click, and then select **Delete Column**/**Delete Row** from the shortcut menu.

Formatting text and paragraphs in a tabular cell

When you enter text into a tabular cell, paragraphs are created. The text and paragraphs in a tabular cell can be easily formatted. For example, you can change the font face, font color of the text, the indentation of any paragraph. You can even find and replace a text string with another one easily as with a word processor. For detail, refer the specific topics in the Text boxes section.

See an example: The SampleComponents catalog, included with JReport Designer, contains reports that have examples of how each component type could be used in a report. For the tabular component example, open `<install_root>\Demo\Reports\SampleComponents\TabularReport.cls`.

Subreports

A report that is inserted into a report is referred to as the subreport. Subreports are inserted and managed just like a component in a primary report, but a subreport itself is created and managed just like a primary report. You can specify relationships between the primary report and the subreport as well as have JReport pass values to the primary report or the subreport. A primary report can have more than one subreport.

A subreport can be inserted in the report areas listed in Component placement.

To insert a subreport into a report:

1. Open the primary report.
2. Position the mouse pointer at the destination where you want to insert the subreport, then click **Insert** > **Subreport** on the menu bar, or the **Insert Subreport** button on the Insert toolbar. Then,
   - If the destination is the report body or a tabular cell, the Subreport dialog appears.
   - If the destination is a banded panel, click the mouse button in the panel to open the Subreport dialog.
3. In the Subreport dialog, click the **Browse** button to select the report which contains the report tab you want to use as the subreport. This report must belong to the same catalog as the primary report.
4. Select the required report tab as the subreport from the Report Tab drop-down list, where the names of all the report tabs in the specified report are displayed.
5. In the Field tab, click the **Add** button beside the Component in Report Tab box. In the Choose Component dialog, add the components in the subreport you want to be interlinked with the primary report with conditions. If the required components use different datasets, you need to add them respectively, however, as long as the components are based on the same dataset, you can add them all at a time.
6. Select a component from the Component in Report Tab box and set up links between the primary report and the component as required. Repeat this to define the link condition between other components in the subreport and the primary report.
7. If the subreport has parameters and you do not want to enter values for them at runtime, click the **Parameter** tab to assign values to the parameters.
8. If you want the subreport to return certain values to the primary report, click the **Return Value** tab and specify the settings according to your requirements.
9. Click **OK** to insert the subreport.

When a subreport is inserted into a report, you can further format the subreport settings if required. To do this, right-click the subreport and click **Format Subreport** on the shortcut menu. In the Subreport dialog, edit the settings according to your requirements. Moreover, when a report has been inserted into another report as a subreport, if you later modify this report, it is suggested that you go to the primary report and use the Format Subreport command to save the subreport settings again.

You can also customize the appearance and behavior of a subreport by setting properties in the Report Inspector, which can be saved as a CSS style for future use (for details, see Creating a CSS style).

Creating sublinks

A sublink is a way to connect the primary report and the subreport. When you define a sublink, JReport will match the records with equal relationships within the subreport and the primary report. You can apply one or more links to make the subreport and the primary report interlinked.

The following example explains the use of sublinks in detail.

Assume that you have two reports in the same catalog: one contains a customer report tab which has a banded object inside and displays information of customer ID, country, city and phone, and the other contains an order report tab which contains a crosstab showing information of customer ID, product name and quantity. Now, you want to insert the order report tab as subreport into the customer report tab, and set up a link between them, so that when the end users views the customer report tab, a subreport will be built for each customer based on the order report tab, showing order records for only the specified customer.

To achieve this, take the following steps:

1. Open the report that contains the customer report tab in JReport Designer.
2. Select the detail panel of the banded object and click the **Insert Subreport** button on the Insert toolbar. A box is then attached to the mouse pointer to indicate it is ready to place the subreport.
3. Click in the panel and the Subreport dialog is displayed.
4. Click the **Browse** button to select the report that contains the order report tab as the subreport.
5. In the Field tab, click the **Add** button.
6. Check **CTCrossTab** in the Choose Component dialog, then click **OK**. The component is listed in the Component in Report Tab box.
7. Add the **Customers_CustomerID** field in the customer report tab to the Field box, a link is then automatically set up as follows:

Fields(Primary)	OP	Fields(Subreport)
Customers_Customer ID	=	CUSTOMERS_CUSTOMERS ID

8. Click **OK** to apply the settings.
9. Preview the customer report tab. You will find that a crosstab is generated for each customer based on the customer ID.

mers_Customer ID	Customers_Countr	Customers_City	Customers_Phone
1	USA	NewYork City	(212) 555-3462

	1	Total
Blue Mountain	192	192
Chocolate Haze	506	506
Gold Coast Ble	213	213
Jamaican Rum	256	256
Java Dragon Bl	1207	1207
Sulawesi	420	420
Total	2794	2794

1	USA	NewYork City	(212) 555-3462

	1	Total
Blue Mountain	192	192
Chocolate Haze	506	506
Gold Coast Ble	213	213
Jamaican Rum	256	256
Java Dragon Bl	1207	1207
Sulawesi	420	420
Total	2794	2794

1	USA	NewYork City	(212) 555-3462

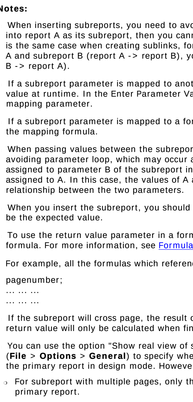
Setting up parameter links

If a subreport contains parameters, the values must be supplied at runtime. You can map the subreport parameters to DBFields, formulas or parameters which are contained in the dataset of the component where the subreport is inserted in the primary report, and are of the same data types as the subreport parameters.

The following example shows how to set up parameter links between a primary report and a subreport.

Assume that you have two reports in the same catalog: Report A (with Report Tab A in it) and Report B (with Report Tab B in it). Report Tab B has a parameter PToday (Date type). Then, you insert Report Tab B as subreport into Report Tab A. When you view Report Tab A, you will be prompted to enter value for its subreport parameter PToday. However, you do not want to enter the value each time you run Report Tab A and instead want to give it a fixed value. To achieve this, follow the steps below:

1. Right-click the inserted subreport in Report Tab A and select **Format Subreport**.
2. In the Subreport dialog, click the **Parameters** tab (this tab is disabled if the subreport does not contain any parameters).
3. In the Parameters column, you can see PToday is listed there and a same parameter is assigned to it by default in the Value column.
4. Click the value drop-down list where all the DBFields, formulas and parameters of Data type in the dataset of Report Tab A are listed. Here, we select **MYTODAY** which is a predefined formula and returns a value ToDate(2006,01,01).



5. Click **OK** to apply the settings.
6. View Report Tab A, and no parameter dialog will be displayed prompting you to enter the value since the formula value is passed to PToday.

Returning values to the primary report

You can make values of the subreport return to the primary report. To do this:

1. Right-click the subreport and select **Format Subreport** from the shortcut menu.
2. In the Subreport dialog, click the **Return Value** tab.
3. In the Component in Report Tab box, check the component in the subreport, the fields of which you want to use to return value to parameters of the primary report, then the Field in Subreport box will list all available fields used by the selected component.
4. Select the field that you want to assign to a parameter in the primary report, click , and a parameter in the primary report of the same data type will be displayed in the Parameters column of the Return Value box. If the automatically assigned parameter is not what you need, select the required one from the parameter drop-down list.
5. If required, select another component in the Component in Report Tab box and set more mapping fields for parameters in the primary report.
6. When done, click **OK** to apply the settings.

Notes:

- When inserting subreports, you need to avoid link loop. For example, if you have inserted report B into report A as its subreport, then you cannot insert report A into report B as its subreport again. It is the same case when creating sublinks, for example, if you set the links between the primary report A and subreport B (report A -> report B), you cannot set the link back to report A (report A -> report B -> report A).
- If a subreport parameter is mapped to another parameter, you will still be prompted to input the value at runtime. In the Enter Parameter Values dialog, the parameter name you see is that of the mapping parameter.
- If a subreport parameter is mapped to a formula, the parameter will obtain value automatically from the mapping formula.
- When passing values between the subreport and the primary report, you should pay attention to avoiding parameter loop, which may occur as this case: parameter A of the primary report is assigned to parameter B of the subreport in the Parameters tab, while in the Return Value tab, B is assigned to A. In this case, the values of A and B will depend on the default values and the position relationship between the two parameters.
- When you insert the subreport, you should insert it before the parameter, or the parameter will not be the expected value.
- To use the return value parameter in a formula, you need to add PageNumber in the first line in the formula. For more information, see Formula levels.

  For example, all the formulas which referenced the return value parameter should be like:

  pagenumber;
  ... ...
  ... ...
- If the subreport will cross page, the result of the formula may not be the expected value because the return value will only be calculated when finishing laying out the subreport.
- You can use the option "Show real view of subreport" in the General category of the Options dialog (**File** > **Options** > **General**) to specify whether or not to show the detailed subreport structure in the primary report in design mode. However,
  - For subreport with multiple pages, only the report body in the first page will be shown in the primary report.
  - For cascading subreport, only the first layer of the subreport will be shown in the primary report.

See an example: The SampleComponents catalog, included with JReport Designer, contains reports that have examples of how each component type could be used in a report. For the subreport examples, open `<install_root>\Demo\Reports\SampleComponents\Subreport_*.cls`.

Images

An image is a digital representation of a picture. JReport supports the following image formats: .bmp, .gif, .jpg, and .png.

This section describes how to use the image component.

Inserting an image into a report

An image can be inserted in the report areas listed in Component placement.

To insert an image into a report:

1. Position the mouse pointer at the destination where you want to insert the image.
2. Do one of the following to display the Select an Image dialog.
   - Click **Insert** > **Image** on the menu bar.
   - Click the **Insert Image** button on the Insert toolbar.
   - Drag **Image** from the Toolbox panel to the destination.
3. Specify the image you want to insert.
   - If you want to use an image from your local file system, use the Look in drop-down list to browse for the image, then click **Open**. If the image file is not in the current catalog directory, you will be prompted to copy it to the directory.
   - If you want to use an image via URL, enter the URL in the URL field and then click **OK**.
4. Click **Open** to insert the image into the destination.

Notes:

- If want to use an image via URL and you are in an intranet, to successfully access the image, you need to add the parameters `-DproxyHost=XXX -DproxyPort=XX` to the batch file JReport.bat, which is located in `<install_root>\bin`.
- If you insert an image via URL into a report, when the report is saved, only the URL information will be saved into the report template.

When an image has been inserted into a report, you can customize its appearance and behavior by setting properties in the Report Inspector, which can be saved as a CSS style for future use (for details, see Creating a CSS style).

Binding a link to an image

You can make an image linked with another report, a location specified by a URL, an e-mail address, or a Blob data type field. The link can either be a simple link or a conditional link. With conditional link, different targets can be loaded based on different conditions. However, conditional link is not supported on images that are not bound with a dataset.

To bind a link to an image:

1. Right-click the image and click **Link** on the shortcut menu.
2. In the Insert Link dialog, specify whether it is a conditional link or a simple link, then select the link target and set the options for the link target as required.
3. When done, click **OK**.

For details about different link targets, see Binding a link to a label.

See an example: The SampleComponents catalog, included with JReport Designer, contains reports that have examples of how each component type could be used in a report. For the image component example, open `<install_root>\Demo\Reports\SampleComponents\ForImage.cls`.

Maps

A map is a special kind of image in JReport. JReport now supports two kinds of maps: shape map and geographic map.

- Shape map contains a picture and certain areas that are bound with groups in a banded object. Shape map features only work when reports run in HTML format or in Page Report Studio.
- Geographic map is a kind of data component integrated with Google Maps, OpenStreetMap or OpenCycleMap. Geographic map features only work in Page Report Studio, Web Report Studio and JDashboard.

Shape maps

A shape map is a special kind of image in JReport. It contains a picture and certain areas that are bound with groups in a banded object. Shape map features only work when reports run in the HTML format or in Page Report Studio.

This section describes the following tasks related to shape maps:

- Inserting a shape map
- Binding data to a shape map
- Adding conditional formats to shape map areas
- Binding links to shape map areas
- Controlling labels and summary fields in each specific shape map area

See an example: The SampleComponents catalog, included with JReport Designer, contains reports that have examples of how each component type could be used in a report. For the shape map component example, open `<install_root>\Demo\Reports\SampleComponents\MapReport.cls`.

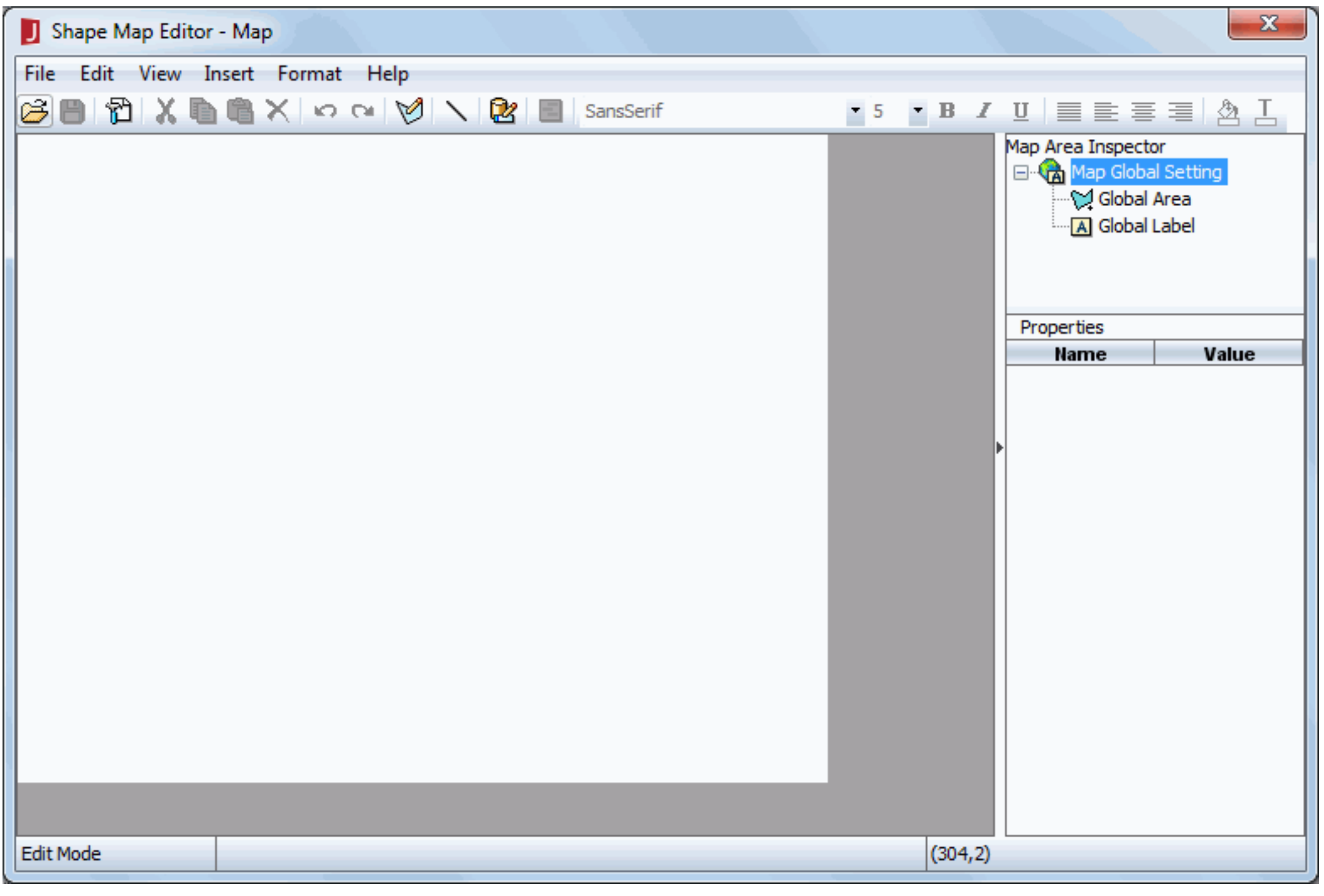
Inserting a shape map

Before you can insert a shape map, you need to create a tabular report or a banded object.

To insert a shape map into a report:

1. Select a tabular cell, or the banded header/footer, group header/footer, or banded page header/footer panel of the banded object.

2. Click **Insert** > **Map** > **Shape Map** on the menu bar, or the **Insert Shape Map** button on the Insert toolbar. The Shape Map Editor window appears.



3. If you want to use an image as background of the map, click **Insert** > **Background Image** on the menu bar of the Shape Map Editor window, then click ... in the Shape Map Background Image dialog to specify the image.

4. Next is to add areas to the map. You can do this either by importing from an previously saved XML file or an ESRI file which has been defined with some area information, or by drawing the areas manually, or using a combination of these two.
   - To import the predefined map areas, click **File** > **Open** in the Shape Map Editor window, then in the Import Map Area dialog, specify the .xml file or .shp file that contains the required area information and click **Open** to import the areas. If you choose to import the map areas from a .shp file, you need to further specify a field in the ESRI shape file to define area name of the map, and if required, you can also select fields from the ESRI shape file to define the area tooltips (for more information, see Select Area Name dialog).
   - To draw the areas manually, first check the **Insert Mode** item on the Edit menu to switch the Shape Map Editor to insert mode.

     To draw an area, click the left mouse button to pick the start point of the area. Move the mouse pointer and click the left mouse button to select the other points. The points will be connected automatically. When you move the mouse pointer to the start point, a square will appear indicating that the area will be closed. Click the left mouse button to finish drawing the area. If you want to undo your last line, click the right mouse button.

5. If you want to further change the shape or position of the areas, uncheck **Insert Mode** on the Edit menu to change the Shape Map Editor to edit mode.

   To change the shape of an area, move the mouse pointer over the border or the points of the area. When it becomes a cross, choose a point on the line and drag it to create a new point. You can also choose a point of the area and drag it to a new position. If there are three points on one line, the middle point will be deleted automatically.

   To move an area, move the mouse pointer close its border. When it changes to a four-headed arrow, drag the area to a new position.

   > **Tip:** If you have specified a background image for the map, and the size of it is not the same as that of the map object, you can click **Insert** > **Match Background Image** in the Shape Map Editor window to resize the map object so as to make it match the size of the background image.

6. To bind data to the map areas, click **Insert** > **Bind Data**, then specify the settings according to your requirements (for details, refer to Binding data to a shape map).

7. If you want to add some conditional formats to the map areas, click **Format** > **Conditional Formatting**, then define the conditions and formats as required (for details, refer to Adding conditional formats to shape map areas).

8. If required, you can make the map areas linked with different targets (for details, see Binding links to shape map areas).

9. In the Map Area Inspector panel, edit properties of any area, label, summary field and line in the map if necessary.

   For areas and labels, you can specify the properties globally and apply the global settings to individual areas and labels conveniently. To specify area or label properties globally, select the **Global Area** or **Global Label** node in the Map Area Inspector tree, then edit the property values accordingly.

   To make a specific area or label adopt the global settings, select the area or label in the inspector tree, then set its Use Global Setting property to **true**. If you want the global settings to be applied to all the areas or labels at a time, click **Edit** > **Reset All**, then in the Reset All dialog, specify the properties according to your requirements.

10. If you want to save the area information you have defined in a file for future use, click **File** > **Save As**. In the Save Map Area dialog, specify the directory where you want to save the file and a name for it, then click **Save**. Then, when you create a report with a map next time, you can choose to import areas from the saved XML file.

11. When done, click **File** > **Save** to save the changes to the map and close the Shape Map Editor window.

12. Click the mouse button in the tabular cell, or banded panel to insert the map.

When a map has been inserted into a report, you can customize its appearance and behavior by setting properties in the Report Inspector, which can be saved as a CSS style for future use (for details, see Creating a CSS style).

Binding data to a shape map

You can make a shape map bound with a dataset, so as to make values displayed on the shape map areas. To achieve it:

1. In the Shape Map Editor window, click **Insert** > **Bind Data**, or the **Bind Data** button on the toolbar. The Shape Map Data Binding Wizard appears.

2. In the Data screen of the wizard, define or select the dataset with which the map will be bound.

   - **New**
     If checked, the map will be bound with a new dataset created from the catalog resources.

   - **Existing**
     If checked, specify the dataset in the current report with which the map will be bound.

   - **Current**
     If checked, the map will be bound with the dataset used by its parent.

3. In the Group screen, add a field as the grouping criteria to group data in the specified dataset. You can add only one group by field (for details, see the same topic Grouping the data on table).

4. In the Summary screen, add the fields as the sum on fields, and specify the function as required. Data in each group will then be calculated based on the selected fields using the specified function, and the summary results will be displayed accordingly on map areas the names of which match with values of the group by field.

5. In the Filter screen, define some filter conditions to filter the data displayed on the map areas.

   To define a filter condition, click the **Add Condition** button to add a condition line, then from the field drop-down list, select the field on which the filter will be based, set the operator with which to compose the filter expression from the operator drop-down list, and in the value combo box, type the value of how to filter the field or click ... to specify the value in the Values window.

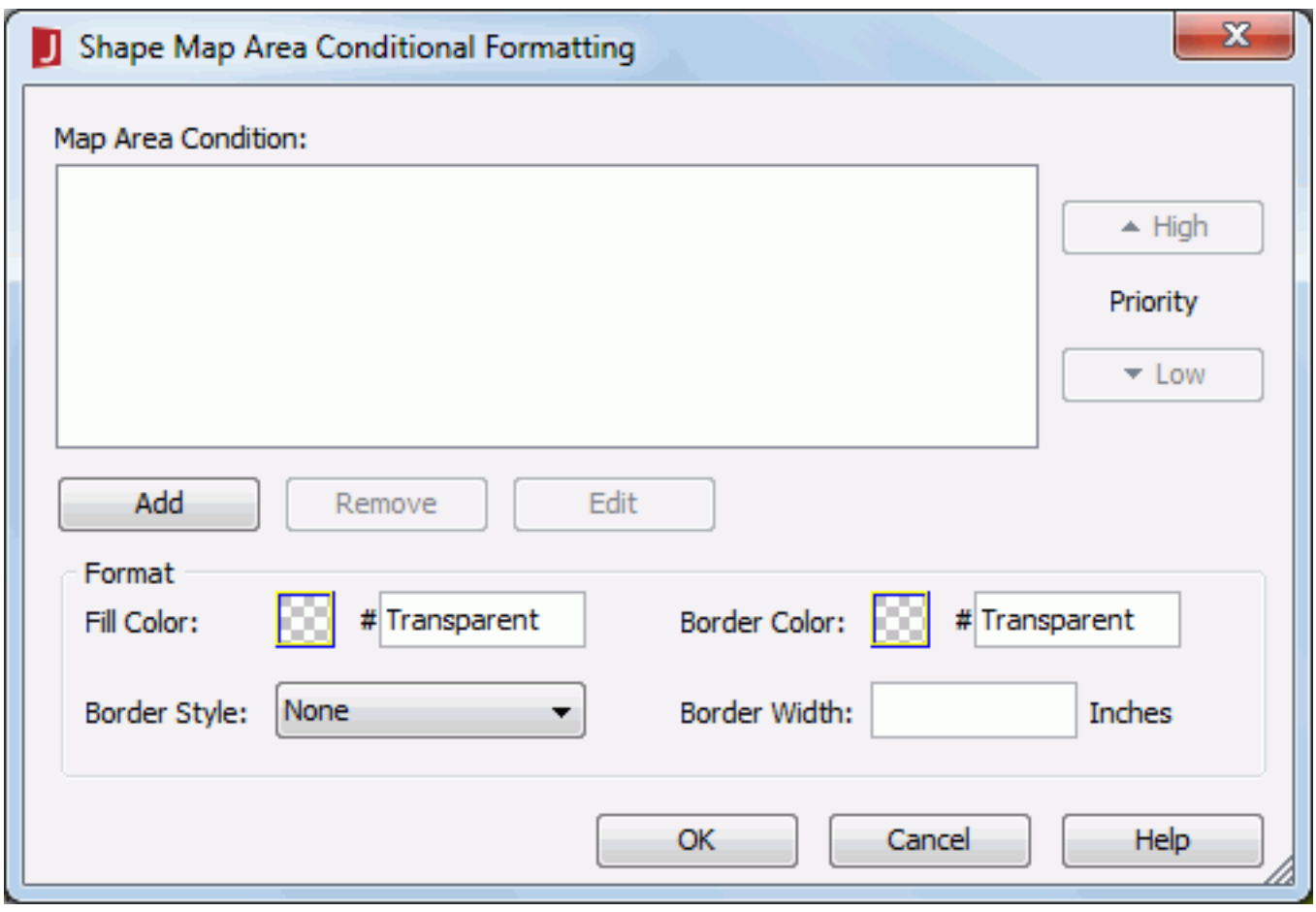
6. Click **Finish** button to apply setting and leave the dialog.

Adding conditional formats to shape map areas

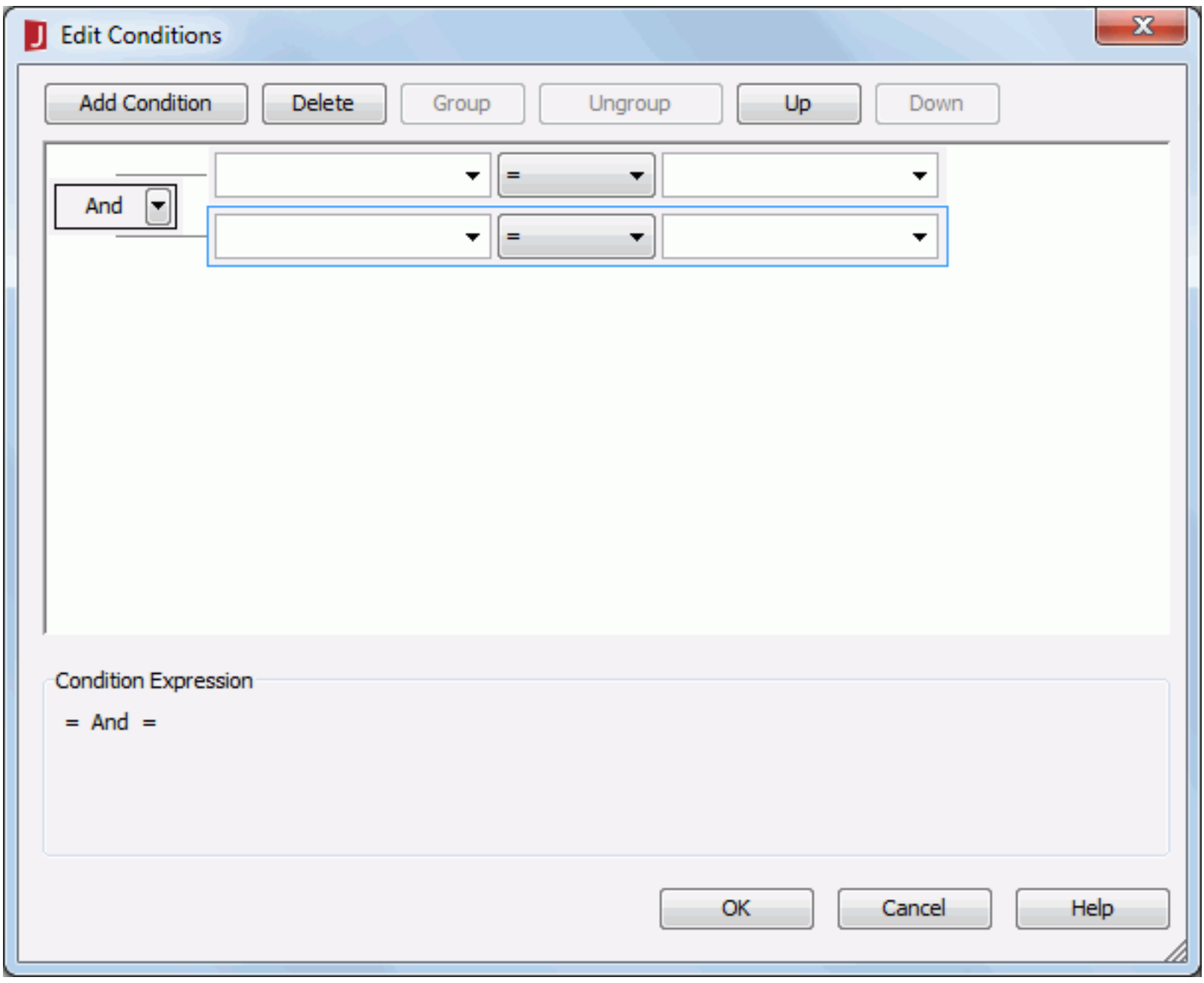
When the shape map areas are bound with data, you can add some conditional formats to them, then when a specified condition is fulfilled, the format bound with the condition will be applied to the shape map areas automatically for distinguishing.

To add conditional formats to the map areas:

1. In the Shape Map Editor window, click **Format** > **Conditional Formatting**, or the **Conditional Formatting** button on the toolbar. The Shape Map Area Conditional Formatting dialog appears.



2. Click the **Add** button below the Map Area Condition box. The Edit Conditions dialog appears.



3. Click the **Add Condition** button to add a condition line.
4. From the field drop-down list, select the field on which the condition will be based.
5. Choose the operator with which to compose the condition expression from the operator drop-down list.
6. Click ... next to the value box to specify the value of how to build the condition. You can also type in the value manually.
7. Click **Add Condition** to add more condition line and define the relationship between the condition lines.
8. When done, click **OK** to save the condition.

   The newly added condition will then be displayed in the Map Area Condition box in the Map Area Conditional Formatting dialog.
9. In the Format box of the Map Area Conditional Formatting dialog, set the format which will be applied to the specified map areas when the specified condition is fulfilled, including the fill color, border color, border style, and border width.
10. Repeat the above steps to add more conditions and define the format for each condition.

    To edit a condition, select the condition from the Map Area Condition box, click the **Edit** button, then edit the condition in the Edit Conditions dialog.

    To adjust the priority of the conditions, select a condition and click the **High** or **Low** button.

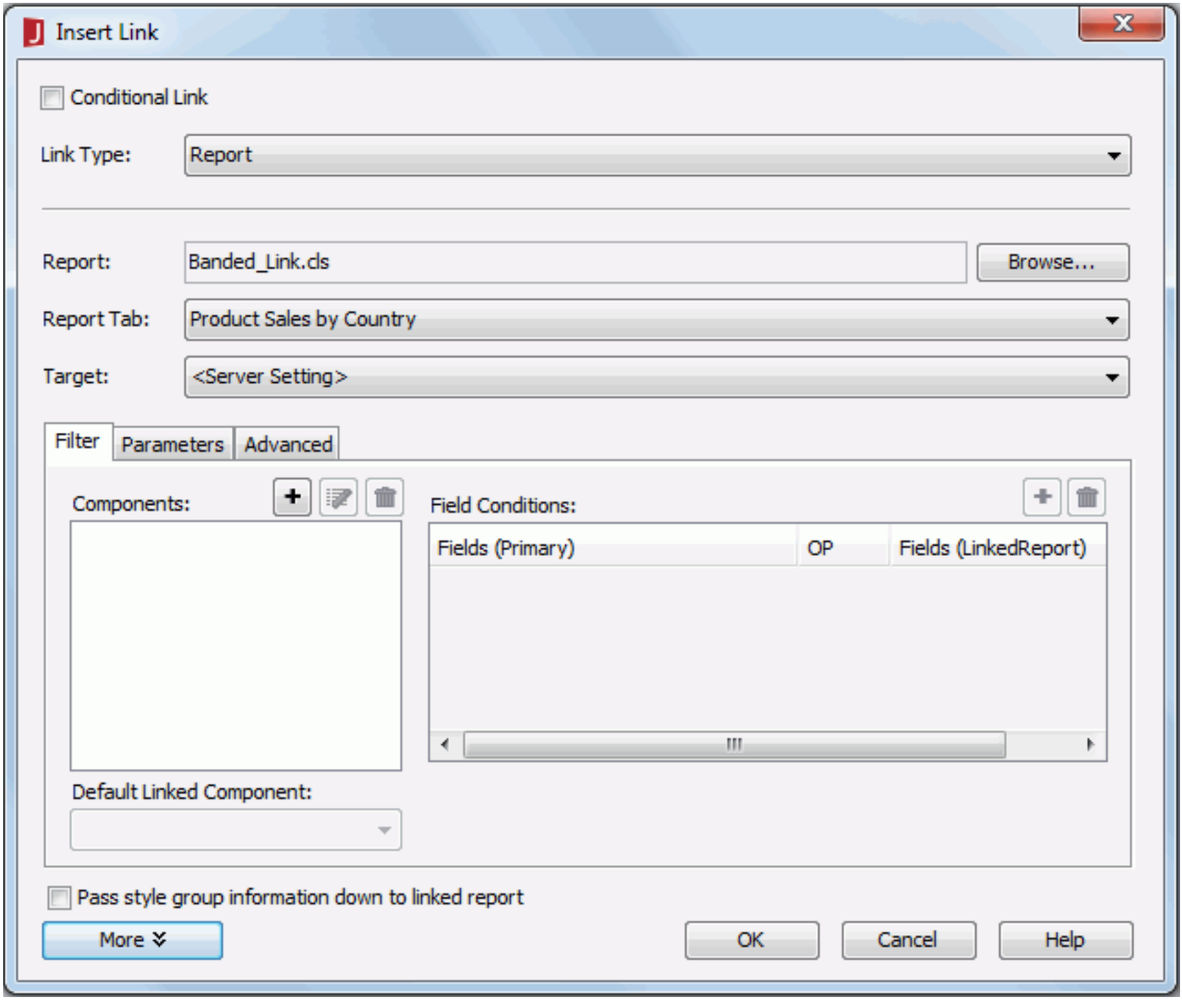
    To remove a condition and the corresponding format, select the condition from the Map Area Condition box and then click the **Remove** button.
11. When done, click **OK** to save and apply the settings.

Binding links to shape map areas

You can make the shape map areas linked with other reports, locations specified by URLs, e-mails, or Blob data type fields. The link can either be a simple link or a conditional link. With conditional link, different targets can be loaded based on different conditions.

To bind a link to a shape map areas:

1. Right-click a map area and select **Link** from the shortcut menu. The Insert Link dialog appears.



2. In the Insert Link dialog, specify whether it is a conditional link or a simple link, then select the link target and set the options for the link target as required.
3. When done, click **OK**.
4. Select another map area and repeat the above steps to add a link on it.

For details about different link targets, see Binding a link to a label.

Controlling labels and summary fields on shape map areas

You can control the labels and summary fields on the shape map areas by taking the following operations in the Shape Map Editor window:

- **Formatting labels/summary fields**
  To format a label/summary field on a map area, first select it, and then perform any of the following tasks:
  - To change the alignment, click the required alignment button on the Shape Map Editor toolbar: **Justify**, **Left**, **Center** or **Right**, or click the corresponding command from the Format > Align submenu. The content in the label/summary field will then be aligned according to your selection.
  - To change the font face, click the **Font Face Box** Arial and then select the required font from the drop-down list. The fonts introduced with * are True Type Fonts. Refer to the chapter True Type Fonts for more information.
  - To change the font size, click the **Font Size Box** 9 and then select the required size from the drop-down list (or enter the required value in the combo box and then press **Enter**).
  - To change the font style, click the corresponding font style buttons on the toolbar: **Bold** B, **Italic** I and **Underline** U, or the commands from the Format > Text Style submenu.
  - To change the background color, click the **Background Color** button and then specify the color as required.
  - To change the foreground color, click the **Foreground Color** button T and then select the desired color.
- **Setting properties for labels/summary fields on map areas**
  In the Map Area Inspector panel of the Shape Map Editor window, you can set properties of the labels/summary fields on the map areas. The label/summary field properties can be set either globally or respectively.
  - To set the properties globally, in the Map Area Inspector panel, expand the **Map Global Setting** node, select the sub node representing the label/summary field, then in the Properties panel, set the properties as required.
  - To set the properties for any label/summary field in specific, in the Map Area Inspector panel, select the node representing the label/summary field, edit the Use Default property of the label/summary field to **false**, then set its properties.

  For details about properties of labels and summary fields in a map, refer to Labels and Summary fields in the Property References chapter.
- **Making labels/summary fields automatically connected with map areas**
  You can drag the labels/summary fields out of the areas they describe and place them at other places, however, after you do this, you will not be able know clearly the connection between each label/summary field and the corresponding area. To keep you from being confused, JReport provides you with the Auto Connector property on labels and summary fields on map areas, by enabling which a line will automatically be drawn between the label/summary field and the area it describes when the label/summary field is dragged out of the area. This property can also be set globally for all the labels/summary fields in a map and for each label/summary field in specific.

  For the automatically drawn lines, you can set the properties if required.
- **Hiding labels/summary fields**
  To hide a label/summary field, right-click it and select **Hide** from the shortcut menu. To show a hidden label/summary field again, select the node representing it from the Map Area Inspector panel, then set its Invisible to **false**.

Geographic maps

The geographic map component is a data component integrated with Google Maps, OpenStreetMap or OpenCycleMap, which enables analyzing data geographically. Geo Analysis extends traditional data analysis on static maps, where zooming and panning are common, to more dynamic methods of interacting with geographic data. Drill down the hierarchy from country to state to county. Navigate through and analyze the data using a familiar map interface.

- **Google Maps**

  Using the Google Maps Javascript API, three types of maps can be displayed in JReport: roadmap, satellite and terrain. JReport only supports the Google Maps Javascript API Version 3.

  Users with higher permission level can access more resources about Google Maps. For Google Maps API documentation, it is suggested that you apply for one API key to Google Maps application. If you have one API key, you can add it directly in the configuration file mapConfiguration.xml in `<install_root>\bin` with the following format: `<property name="google.api.key">your key</property>`. If you are a business user with a client ID, you can add it in the configuration file mapConfiguration.xml with the following format: `<property name="google.api.client">your client id</property>`. Then you can access more resources about Google Maps.

  The interactive features of Google maps only work in Page Report Studio, Web Report Studio and JDashboard; however, static Google map images are supported in PDF and HTML formats with some limitations (for details, refer to Google Static Maps API).

- **OpenStreetMap and OpenCycleMap**

  Using OpenLayers Javascript API, OpenStreetMap and two types of OpenCycleMap, namely, Cycle Map and Transport Map, can be displayed in JReport.

  OpenStreetMap and OpenCycleMap cannot be displayed in the Designer View mode or be exported or printed in JReport Designer as the Static Maps API is not supported at present. Their interactive features only work in Page Report Studio, Web Report Studio and JDashboard.

Every geographic map component has its own data definition and can be manipulated by report developers. You can create a geographic map with a defined data source and group levels. Each group level has some markers with marker tips which show the defined tip information when you move the mouse over the markers at runtime. For geographic maps in web reports or library components, each group level can even have some areas with area tips, and each area can be shown with a polygon of distinct color.

Pick a task from the following:

- Inserting a geographic map
- Adding conditional formats to geographic map markers
- Example of using geographic map

Notes:

- To use the geographic map features, you have to make sure your JReport Server and browser have access to the internet. If you are in an intranet, you should specify the proxy for both of them. To specify the proxy for the server, you need to open the batch file JRServer.bat in `<install_root>\bin`, and then add the following parameters in the file::
  - **http.proxyHost**: The host name of the proxy server.
  - **http.proxyPort**: The port number, the default value being 80.
  - **http.nonProxyHosts**: A list of hosts that should be reached directly, bypassing the proxy. This is a list of patterns separated by '|'. The patterns may start or end with a '*' for wildcards. Any host matching one of these patterns will be reached through a direct connection instead of through a proxy.

  Besides HTTP protocol, HTTPS protocol is also supported by JReport. Loading the Maps API over HTTPS is necessary in SSL applications to avoid security warnings in most browsers, and is recommended for applications that include sensitive user data, such as a user's location, in requests.

  You can specify the required URL in the configuration file mapConfiguration.xml in `<install_root>\bin`.

- The Geographic Map feature cannot work on Internet Explorer 8.

Inserting a geography map

This document explains how to insert geographic map into a page report, web report and library, component respectively.

Inserting a geographic map to a page report

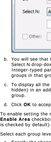
1. Position the mouse pointer at the destination where you want to insert the geographic map.
   A geographic map can be inserted into the page report areas listed in Component placement, or the blank area in a library component or web report.
2. Do one of the following to open the Create Geographic Map wizard:
   - Click **Insert > Geographic Map** on the menu bar.
   - Drag **Geographic Map** from the Toolbox panel.
   - Click the **Insert Geographic Map** button on the Insert toolbar.
3. In the Data screen of the wizard, define or select the dataset on which the geographic map will be created.
   - **New**
     If checked, the geographic map will be built on a new dataset created from the catalog objects.
   - **Existing**
     If checked, the geographic map will be built on the dataset which is used by its parent.
   - **Group Server**, add the fields as the grouping criteria, and then specify the sorting direction of each group level in the Sort Order section. Refer to the same topic Grouping the data for details.
4. In the Filter screen, you may define some optional filter conditions to filter the data displayed in the geographic map. Refer to the same topic Filtering the data on details.
5. In the Layout screen, specify the geographic data for the geographic map, which is GeoJSON, and the `<install_root>/gis` directory, which is the root directory of the GeoJSON files, or you will get the related `<install_root>/gis` directory. The geographic data XML file is used to map the geographic latitude and longitude coordinates from Google Maps, OpenStreetMap or OpenCycleMap based on the location information specified for each group in the Marker tab. When the location information key is not found in a group value matches the "id" field in the XML file, you will get the related latitude and longitude value for this group value.
   You can also map with the XML which contains predefined geographic data including latitude and longitude information. The field in geographic data should be relative to `<install_root>/gis` directory. In this case, the data is copied to this directory after it is imported. When you publish the report, the report will copy the related geographic data from the <install_root>/gis directory to the Geographic Information tab.
   If you do not specify the geographical information it will be created automatically when you first run the report.
   Then specify the map type from the Map Type drop-down list, which can be Google Maps - Roadmap, Google Maps - Satellite, Google Maps - Terrain, OpenStreetMap - Standard, OpenStreetMap - Cycle Map, OpenCycleMap - Transport Map, Google Maps - Terrain.
7. In the Marker screen, select each group level from the Geographic Map box and specify its marker properties.
   Specify the location of the markers by selecting a DBField or a formula from the Location link drop-down list which contains the values which matches the "id" field in the GeoJSON or Map XML file. The values seen in this Location link field will be used to create the "id=" field values when the XML does not already exist.
   Specify the tip information which displays when you hover the mouse on the markers at runtime by typing it in the Marker Tip text box directly. You can also use a formula to control the tip.
   To customize the image for the markers, uncheck the **Default Icon** option, then specify the image for the markers in the normal state and the image for the markers when mouse over them respectively. If you do not specify a mouse over image, the image in the normal state will change the same in a certain proportion, check **Constrain Proportion**.
   Specify whether to show the markers. You can simply select true or false from the Hide Markers drop-down list, or use a formula to control it.
   Click **Finish** to insert the geographic map.

   If you have used the menu or toolbar command to insert the geographic map into a page report, and specified to insert it in a page header/footer or header/footer area, and then clicking Finish in the wizard, you need to drag it to the destination using your mouse in order to insert the geographic map into the area.

Inserting a geographic map to a web report

1. Follow above steps 1 and 2 in inserting a geographic map to a page report to open the Create Geographic Map wizard.
2. In the Data screen of the wizard, select a business view in the current catalog on which the geographic map will be created.
   In the Display screen, specify the map type from the Map Type drop-down list, it could be OpenStreetMap - Standard, OpenStreetMap - Cycle Map, OpenCycleMap - Transport Map, Google Maps - Roadmap, Google Maps - Satellite, Google Maps - Terrain.
4. The Resources box lists all the elements in the selected business view, you can further edit the business view if you wish.
   To edit the business view, right-click the business view or any element in it and select **Edit** from the shortcut menu. The Advanced Business View Editor opens. Edit the business view as you want.
   To change the display name of the business view or an element, right-click the business view or the element and select **Rename** from the shortcut menu. Enter the new name in the name text box and click outside of the box to confirm the new name.
   To delete an element, right-click the element and select **Delete** from the shortcut menu.
   Add the fields from the Resources box to the Drill Path box as the grouping criteria, if needed, specify the Select N condition for each group level. To adjust the order of the added group levels, select the group and click the or above the Drill Path box. To remove a group level, select the group and click .

   To specify the Select N condition for a group:
   1. Select the group to field in the Drill Path box, then click the **Select N** button. The Select N dialog appears.

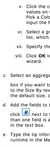


   b. You will see that the field you have just selected is displayed in the in text box. From the Select N drop-down list, select a number or an option from the drop-down list, such as a number or Integer-typed parameter. If the number or the parameter value is N, then the first or last N groups in the sorting order will be displayed.
   c. To display all the other groups that do not match the Select N condition (which are by default hidden) in an additional group, check the **Other** box and type a name for the additional group.
   d. Click **OK** to accept the condition.
6. To enable setting the marker and area properties for all groups, check the **Enable Marker** and **Enable Area** checkboxes (one of the two checkboxes must be checked at least and both are checked is checked by default).
   Select each group level from the Drill Path box and specify its marker properties:
   a. Specify the shape of the markers in any of the following ways.
      - If you want to use a shape as the markers, click next to the Shape box and select a shape from the shape list, then click outside of the shape list to apply the selected shape.
      - If you want to use an image to be the shape of the markers, click then select the checkbox ahead of the **Advanced** option in the shape list. A default image will then be displayed in the Advanced dialog box. You can specify the image for the markers in any of following the checkbox, click **Advanced** to open the Advanced dialog to specify the image.

      i. Uncheck the **Default Icon** option.
      ii. Input the URL to a web image in the Image text box, or click **Browse** button to specify a local image.
      iii. Specify the width and height of the image.
      iv. If you want to change the width and height for the image at the same time in a certain proportion, check **Constrain Proportion**.
      v. If the map is of any Google Maps type, you can further specify a shadow image for the markers.
      vi. When done, click **OK** to apply the images and return to the map wizard.
   - If you want to control the shapes of the markers by field values, select the group field that represents the currently selected group level, or a group field of a higher level than the current level in the Drill Path box, or an aggregate field from the Resources box and click to add it in the Shape by text box, then click to open the Edit Shapes dialog to edit the colors.

      i. From the shape pattern drop-down list, select the required pattern. A preview of the shapes in the selected pattern will be displayed in the box below.
      ii. The field values with the shapes in the selected pattern are listed in the Data Items box. You can change the shape for each value by selected it and then clicking a shape in the preview box. Note that if an aggregation field has been added to control the shapes, the box will be empty. You cannot customize the shapes and colors and the default shapes in the selected pattern will be applied at runtime.
      iii. Click **OK** to apply the selected shapes to the markers and return to the map wizard.
   b. Specify the color of the markers in any of the following ways.
      - Click next to the Color By text box and select a color from the color palette or click **More Colors** to open the Color dialog box to define a Pick a Color dialog, with which you can set a full range of colors.
      - If you want to control the colors of the markers by group values, select the group field that represents the currently selected group level, or a group field of a higher level than the current level in the Drill Path box from the Resources box and click to add it to the Color By text box. You can add more than one group fields and when several group fields are added, you can specify the color of the markers by dragging and dropping them in the text box. Click to open the Edit Color dialog to edit the colors.

      i. From the color pattern drop-down list, select the required pattern. A preview of the colors in the selected pattern will be displayed in the box below.
      ii. The field values with the colors in the selected pattern are listed in the Data Items box. You can change the color for each value by selected it and then clicking a color in the preview box.
      iii. Specify the opacity of the colors in the Opacity text box.
      iv. Click **OK** to apply the selected colors to the markers and return to the map wizard.
      - If you want to control the colors of the markers by aggregation values, select an aggregation field from the Resources box and click to add it to the Color by text box, then click to open the Edit Gradient Color dialog to edit the colors.



        i. Select a gradient color from the color drop-down list. A preview of the gradient color will be displayed in the box.
        ii. You can further customize the gradient start color, middle color and end color by selecting the corresponding color from the color palette or click **More Colors** in the color palette to open the Color dialog box to define a Pick a Color dialog, in which you can select a color within a wider range. The middle and start colors can be deleted, however, if the end color is deleted, a default start color will be applied automatically. To add the start and middle colors back, click + and specify the color in the Pick a Color dialog box. If the middle color is deleted, the start color will be applied as a customized gradient color and you will find the name changed to Customize in the color drop-down list.
        iii. Click **Reversed** if you want to reverse the direction of the gradient color.
        iv. Check the **Middle Value** checkbox and specify the middle data value mapping to the middle color, which will have effect only when the middle color is selected. Then the gradient color will be based on the start color to the middle color, and then change from the middle color to the end color. You can also set the middle value to 0 and check the **Auto Expand Start/End value** checkbox, in which case, the negative and positive values will be displayed in different color, the start color will be the start color and end color will be 0 and vice versa if the value is negative.
        v. Check the color image button beside Color of Null Values to specify a color for null values on the data set. Click **More Colors** on the color palette to access the Pick a Color dialog in which you can select a color within a wider range, or you can input the hexadecimal value of a color directly in the text box after it.
        vi. Select a gradient color algorithm between two colors from the Gradient By drop-down list, which can be Auto, RGB or HSL.
        vii. Specify the opacity of the gradient color in the Opacity text box.
        viii. Click **OK** to apply the specified gradient color to the markers and return to the map wizard.
   c. Select an aggregation field from the Resources box and click to add it to the Size by text box if you want to control the size of the markers by aggregation values, then click next to the Size By text box to open the Edit Range dialog box to define a range based on the default size. To restore the default size, click **Reset**.
   d. Add the fields to the Tip text box to specify the labels shown above the markers, then click next to the Label By text box to set the font properties of the labels. When more than one field is added, you can adjust the order of the fields by dragging and dropping them in the text box.
   e. Type the tip information which displays when you hover the mouse on the markers at runtime in the Marker Tip text box directly. You can also use a formula to control the tip.
   Select each group level from the Drill Path box and specify its area properties:
   a. Specifies the colors that are used to fill the areas in the Fill By text box using the same way for specifying the colors.
   b. Add the fields to the Label By text box to specify the labels shown above the areas, then click next to the Label By text box to set the font properties of the labels. When more than one field is added, you can adjust the order of the fields by dragging and dropping them in the text box.
   c. Type the tip information which displays when you hover the mouse on the areas at runtime in the Area Tip text box directly. You can also use a formula to control the tip.
9. Select each group level from the Drill Path box and specify the location of the markers and areas of the group by selecting a DBField or a formula from the Location link drop-down list which contains the values which matches the "id" field in the GeoJSON or Map XML file. For example, you may group by Country and then set the State, you would want to use a formula RlateCountry that concatenates state and country so Google Maps, OpenStreetMap or OpenCycleMap knows how to use such as "New York, USA" or "California, USA".
10. In the Filter screen, you may define some filter conditions to filter the data displayed in the geographic map. You can select a predefined filter from the specified business view if there is from the Filter drop-down list to apply, or select **User Defined** in the list to define a new filter.
11. Click **Finish** to insert the geographic map.

Inserting a geographic map to a library component

1. Follow above steps 1 to 9 inserting a geographic map to a web report to specify the geographic map information in the Data and Display screens of the Create Geographic Map wizard.
   Select each group level from the Drill Path box to bind web behaviors to the markers and areas of the group.
   a. Click the **Web Behaviors** button. The Web Behaviors dialog appears.

   b. In the Marker tab, select a trigger event from the drop-down list in the Events column, then click on the Actions column and click ellipsis button in the text box to open the Web Action List dialog, where you can bind a web action to the markers which will be triggered when the specified event occurs on the markers.
   To add a web behavior line, click and if a web behavior is not required, click to remove it.
   Click or to adjust the order of the behaviors. Then at runtime, when an event that has been bound with more than one action happens, the upper action will be triggered first.
   c. Switch to the Area tab and follow the same steps to bind web behaviors to the map areas.
   d. Click **OK** to apply the web behaviors to the group and return to the map wizard.
12. Web actions you can bind to the map markers and areas include Filter, Set Parameter, Property and SendMessage. For details about these web actions, refer to Applying web actions to a table.
13. In the Filter screen, you may define some filter conditions to filter the data displayed in the geographic map. You can select a predefined filter from the specified business view if there is from the Filter drop-down list to apply, or select **User Defined** in the list to define a new filter.
14. Click **Finish** to insert the geographic map.

Once a geographic map has been created, you can further modify it by accessing its shortcut menu which is composed by a set of commands that are similar to the edport screens used to create the geographic map. Moreover, you can customize its appearance and behavior by setting properties in the Report Inspector, which are in the Geographic Map section, and by adding conditional formatting (CFS styles).

After finishing the modification, you can also apply some filter conditions to narrow down the records displayed in the geographic map the same as you do for a table (for details, see Filtering the data). However, if the geographic map is based on a dataset, the conditions you define in the Format Filter dialog will not be applied when the geographic map runs due to the specialty of HDS. Therefore, if you want to filter the data displayed in such kind of geographic map, you need to make use of the Dataset filter.

Adding conditional formats to geographic map markers

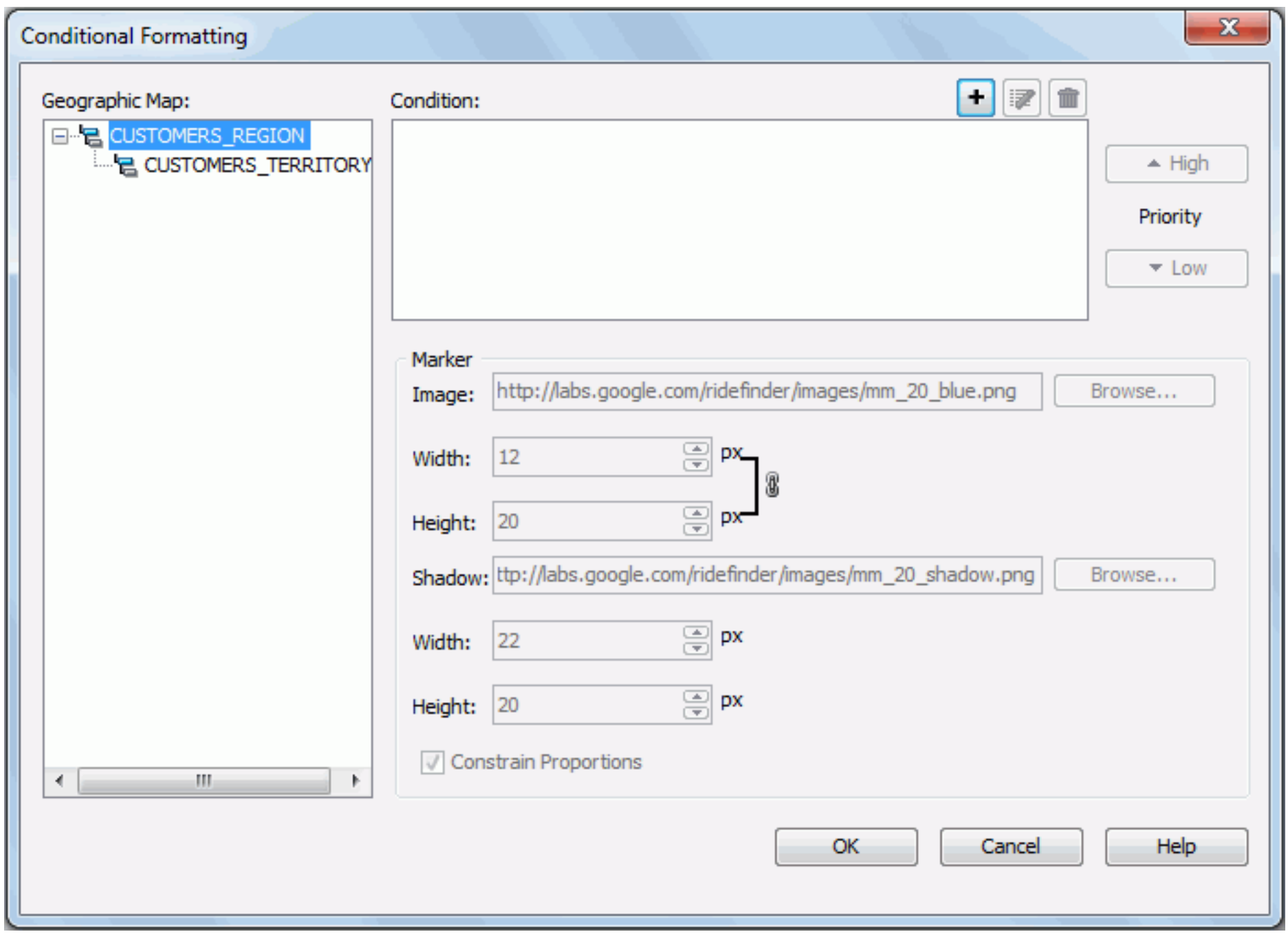
After a geographic map is created, each value in a group is bound with a marker. You can add some conditional formats to the markers, then when a specified condition is fulfilled, the format bound with the condition will be applied to the markers automatically for distinguishing.

However, for geographic maps in web reports and library components, the conditional formats are applied based on the advanced settings of the Shape By option, in other words, when the marker shape is controlled by image. If you apply conditional formats to a geographic map the shape of which is controlled by field or shape, the field or shape that you has specified to the Shape By option will be ignored after the conditional formats take effect.

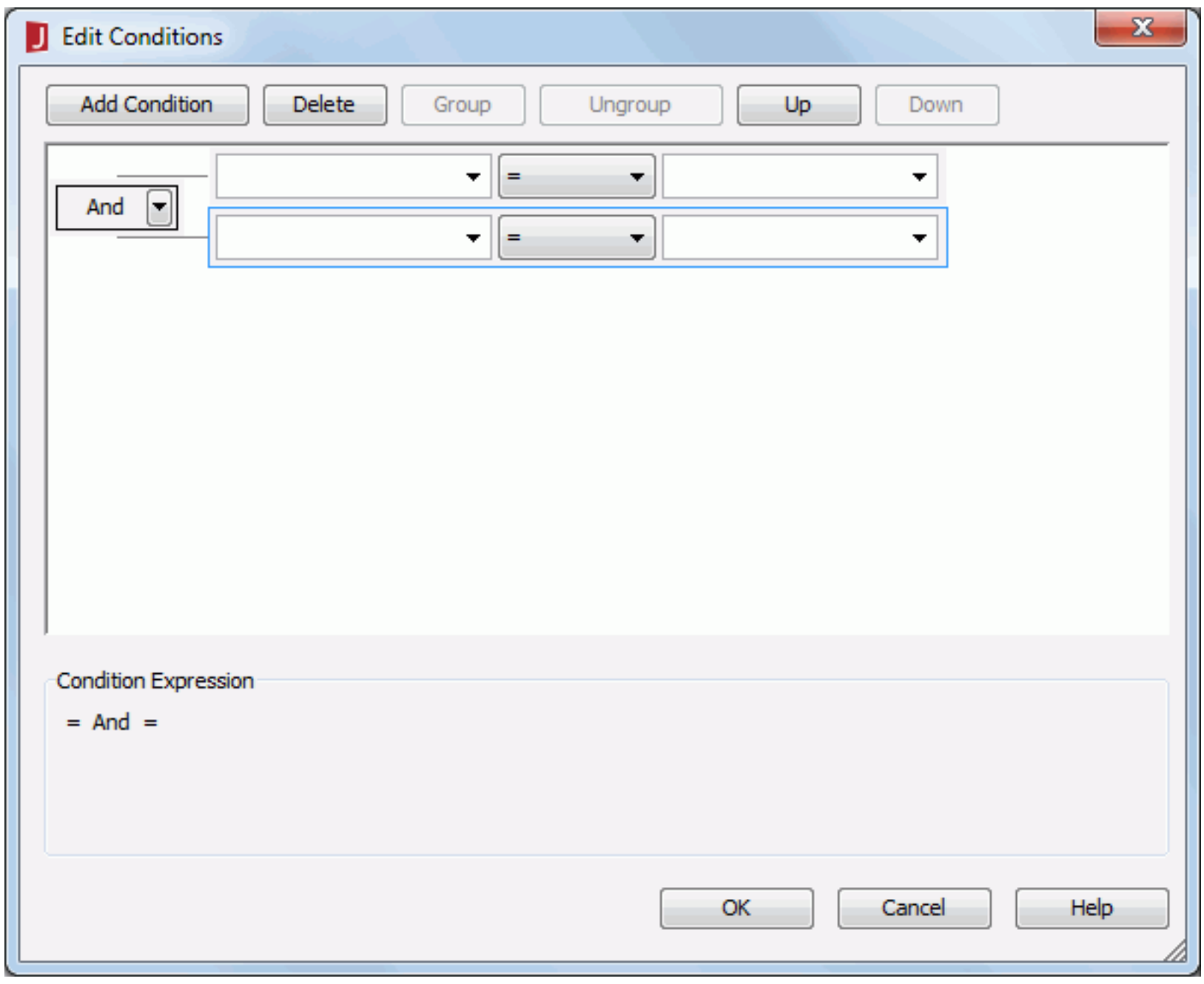
To add conditional formats to geographic map markers:

1. Select the geographic map and then do one of the following:
   - Click **Format** > **Conditional Formatting** on the menu bar.
   - Click the **Conditional Formatting** button on the Format toolbar.
   - Right-click the geographic map and select **Conditional Formatting** from the shortcut menu.

   The Conditional Formatting dialog appears.



2. Select a group level from the Geographic Map box, then click above the Condition box. The Edit Conditions dialog appears.



3. Click the **Add Condition** button to add a condition line.
4. From the field drop-down list, select the field on which the condition will be based.
5. Choose the operator with which to compose the condition expression from the operator drop-down list.
6. From the value drop-down list, select the value of how to build the condition. You can also type in the value manually.
7. Click **Add Condition** to add more condition lines and define the relationship between the condition lines.
8. When done, click **OK** to save the condition. The newly added condition will then be displayed in the Condition box of the Conditional Formatting dialog.
9. In the Marker box, specify the settings which will be applied to the marker of the selected group level when the condition is fulfilled, including the image, the shadow image, and their width and height. If you want to change the width and height for the image/shadow image at the same time in a certain proportion, check **Constrain Proportion**.
10. Repeat the above steps to add more conditions and define the marker properties for each condition.

    To edit a condition, select the condition from the Condition box, click , then edit the condition in the Edit Conditions dialog.

    To adjust the priority of the conditions, select a condition and click the **High** or **Low** button.

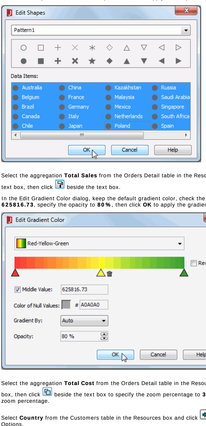
    To remove a condition and the corresponding marker properties, select the condition from the Condition box and then click .
11. When done, click **OK** to save and apply the settings.

Example of using geographic map

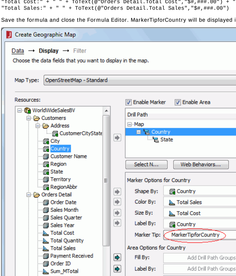
This document takes OpenStreetMap for example to describe how to add a geographic map in a library component and use it in JDashboard.

Creating a library component with a geographic map

1. Open the catalog file SampleReports.cat in the SampleReports folder.
2. Click **File** > **New** > **Library Component** on the menu bar.
3. In New Library Component dialog, clear the text in the Library Component Title box, select **Blank**, and then click **OK**.
4. Drag **Geographic Map** from the Toolbox panel to the library component. The Create Geographic Map wizard appears.
5. In the Data screen of the wizard, select **WorldWideSalesBV** from Data Source 1. Then click **Next**.
6. In the Display screen, add the fields Country and State in the Customers table as the group by fields one by one to the Drill Path box.
7. Check the **Enable Area** checkbox. The Enable Marker checkbox is checked by default.
   Next, we will define the marker and area properties for the country group level.
8. Click **Country** in the Drill Path box, select **Country** from the Customers table in the Resources box and click to add it to the Shape By text box, then click beside the text box.
9. In the Edit Shapes dialog, click anywhere in the Data Items box and click **Ctrl** + **A** on the keyboard to select all countries, then click in the preview box below the pattern name to apply the shape to all countries. Click **OK** to apply the shape.



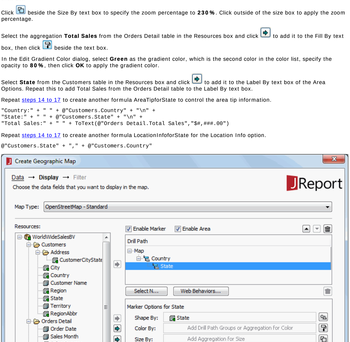
10. Select the aggregation **Total Sales** from the Orders Detail table in the Resources box and click to add it to the Color By text box, then click beside the text box.
11. In the Edit Gradient Color dialog, keep the default gradient color, check the **Middle Value** checkbox and set its value to **625816.73**, specify the opacity to **80 %** , then click **OK** to apply the gradient color.



12. Select the aggregation **Total Cost** from the Orders Detail table in the Resources box and click to add it to the Size By text box, then click beside the text box to specify the zoom percentage to **370 %** . Click outside of the size box to apply the zoom percentage.
13. Select **Country** from the Customers table in the Resources box and click to add it to the Label By text box of the Marker Options.
14. Click beside the Marker Tip text field, then select **<Create Formula>** from its drop-down list to create a formula to control the marker tip information.
15. In the Input Formula Name dialog, enter the name **MarkerTipforCountry** and click **OK**.
16. In the Formula Editor window, define the formula as follows.

```
"Country:" + " " + @"Customers.Country" + "\n" +
"Total Cost:" + " " + Totext(@"Orders Detail.Total Cost","$#,###.00") + "\n" +
"Total Sales:" + " " + Totext(@"Orders Detail.Total Sales","$#,###.00")
```

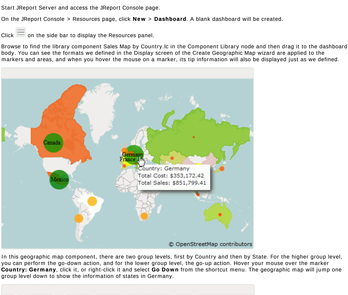
17. Save the formula and close the Formula Editor. MarkerTipforCountry will be displayed in the Marker Tip text field.



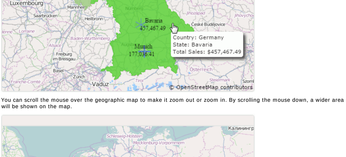
18. Select the aggregation **Total Sales** from the Orders Detail table in the Resources box and click to add it to the Fill By text box, then click beside the text box.
19. In the Edit Gradient Color dialog, select **Green-White-Orange** from the gradient color drop-down list, which is the last color at the second line in the list, specify the opacity to **80 %** , then click **OK** to apply the gradient color.
    Next, we will specify the marker and area properties for the state group level.
20. Click **State** in the Drill Path box, select **State** from the Customers table in the Resources box and click to add it to the Shape By text box, then click beside the text box.
21. In the Edit Shapes dialog, select **Pattern1** from the pattern drop-down list to apply the shapes in the pattern to all states, then click **OK**.
22. Click beside the Size By text box to specify the zoom percentage to **230 %** . Click outside of the size box to apply the zoom percentage.
23. Select the aggregation **Total Sales** from the Orders Detail table in the Resources box and click to add it to the Fill By text box, then click beside the text box.
24. In the Edit Gradient Color dialog, select **Green** as the gradient color, which is the second color in the color list, specify the opacity to **80 %** , then click **OK** to apply the gradient color.
25. Select **State** from the Customers table in the Resources box and click to add it to the Label By text box of the Area Options. Repeat this to add Total Sales from the Orders Detail table to the Label By text box.
26. Repeat steps 14 to 17 to create another formula AreaTipforState to control the area tip information.

```
"Country:" + " " + @"Customers.Country" + "\n" +
"State:" + " " + @"Customers.State" + "\n" +
"Total Sales:" + " " + Totext(@"Orders Detail.Total Sales","$#,###.00")
```

27. Repeat steps 14 to 17 to create another formula LocationInforforState for the Location Info option.
    `@"Customers.State" + "," + @"Customers.Country"`



28. Click **Next** to enter the Filter screen. Click the **Add Condition** button to add a condition line, and specify the filter condition as **@Country != USA** , then click **Finish** to create the map.



29. Select **GeoMarkers** and **GeoAreas** in the Report Inspector by pressing the Ctrl key on the keyboard and set their invisible property to **true** . This is to hide information about the state group level from displaying at the initial map to improve the map appearance. You can still get the state information by applying the go-down action on the countries.
30. Click **File** > **Save** to save the library component as **Sales Map by Country.lc**.

Displaying the geographic map in JDashboard

1. Publish the library component to the default location on JReport Server together with the resources used by it.
2. Start JReport Server and access the JReport Console page.
3. On the JReport Console > Resources page, click **New** > **Dashboard**. A blank dashboard will be created.
4. Click on the side bar to display the Resources panel.
5. Browse to find the library component Sales Map by Country.lc in the Component Library node and then drag it to the dashboard body. You can see the formats we defined in the Display screen of the Create Geographic Map wizard are applied to the markers and areas, and when you hover the mouse on a marker, its tip information will also be displayed just as we defined.

6. In this geographic map component, there are two group levels, first by Country and then by State. For the higher group level, you can perform the go-down action, and for the lower group level, the go-up action. Hover your mouse over the marker **Country: Germany**, click it, or right-click it and select **Go Down** from the shortcut menu. The geographic map will jump one group level down to show the information of states in Germany.

7. You can scroll the mouse over the geographic map to make it zoom out or zoom in. By scrolling the mouse down, a wider area will be shown on the map.

Web controls

Web controls are report components designed to be similar to the kinds of controls found on web pages. JReport provides some web controls such as text fields, drop-down lists, and buttons that are usually used in web pages. In most cases, web controls are not used independently. They are bound with some web actions, or put into a form. When a web control is bound with some web actions which will be triggered on certain events, any of the events applied to the web control will perform the web actions. You can also add some web controls into a form, and then define the form according to your requirements, so that the inputs at runtime can be submitted to JReport Server and thus be processed.

Based on the different functions and usages, web controls are divided into two types in JReport: basic web controls and advanced web controls. For the basic web controls, you need to bind web actions to them by yourself, however, advanced web controls are defined with specific web actions or are used for specific purposes so that you can use them directly.

This section explains how to use web controls in JReport as follows:

- Using basic web controls
- Using advanced web controls

See an example: The SampleComponents catalog, included with JReport Designer, contains reports that have examples of how each component type could be used in a report. For the web control example, open `<install_root>\Demo\Reports\SampleComponents\WebControls_*.cls`.

Using basic web controls

Basic web controls refer to the following: Text Field, Text Area, Checkbox, Radio Button, Image Button, Button, Drop-down List, and List. In most cases, these basic web controls are not used independently. They are bound with some web actions, or put into a form. When a basic web control is bound with some web actions which will be triggered on certain events, any of the events applied to the web control will perform the web actions. You can also add some basic web controls into a form, and then define the form according to your requirements, so that the inputs at runtime can be submitted to JReport Server and thus be processed.

At present, the basic web controls are not supported in web reports and their usages are limited in library components (for details about using basic web controls in library components, see Using the configuration panel).

This section describes how to use the basic web controls as follows:

- Inserting and defining a basic web control
- Binding a link to a basic web control
- Applying web actions to a basic web control
- Making basic web controls work in a form

Inserting and defining a basic web control

The basic web controls can be inserted into the report area hosted in Component placement. This document shows how to insert and define values of the web controls separately.

Text field

To insert a Text field and define its values:

1. Do one of the following to insert the text field:
   - Drag **Text Field** from the Toolbox panel to the destination.
   - On the menu bar, click **Insert > Web Controls > Text Field**, then click the mouse button in the destination.
   - On the Insert toolbar, click the down arrow on the **Insert Control Objects** button and select **Text Field** from the drop-down list, then click the mouse button in the destination.
2. Right-click the text field and select **Display Type** from the shortcut menu. The Display Type dialog appears.

3. In the Web Options panel, select the type of the text field: Standard or Password.
   - Standard means that the text field is a normal text field. Password means that the text field is a password box in which the typed characters are displayed as asterisks.
4. Enter the name, value for the text field in the Name and Value fields.
5. In the Tool Tip field, enter the tip you want to show for the text field. Then at server runtime or in HTML results, when you hover the mouse over the text field, the tooltip will be displayed.
6. Set the character width in the Display Width field, and the maximum number of characters the user can enter in the Length field.
7. Check the **Fixed Only** option if you would like to set this text field to be read only.
8. Check the **Disabled** checkbox if you want to make the text field disabled.
9. If needed, bind some web actions to the text field in the Web Behaviors box.
10. Upon finishing, click the **OK** button to close this dialog.

Hidden field

A hidden field is visible only in design mode.

To insert a hidden field and define its values:

1. Do one of the following to insert the hidden field:
   - Drag **Hidden Field** from the Toolbox panel to the destination.
   - On the menu bar, click **Insert > Web Controls > Hidden Field**, then click the mouse button in the destination.
   - On the Insert toolbar, click the down arrow on the **Insert Control Objects** button and select **Hidden Field** from the drop-down list, then click the mouse button in the destination.
2. Right-click the hidden field and select **Display Type** from the shortcut menu. The Display Type dialog appears.

3. In the Web Options panel, specify the name and value of the hidden field.
4. If needed, bind some web actions to the hidden field in the Web Behaviors box.
5. Upon finishing, click the **OK** button to close the dialog.

Text area

A text area allows you to input a multi-line string.

To insert a text area and define its values:

1. Do one of the following to insert the text area:
   - Drag **Text Area** from the Toolbox panel to the destination.
   - On the menu bar, click **Insert > Web Controls > Text Area**, then click the mouse button in the destination.
   - On the Insert toolbar, click the down arrow on the **Insert Control Objects** button and select **Text Area** from the drop-down list, then click the mouse button in the destination.
2. Right-click the text area and select **Display Type** from the shortcut menu. The Display Type dialog appears.

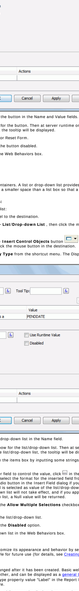
3. In the Web Options panel, enter the name, value of the text area in the Name and Value fields.
4. In the Tool Tip field, enter the tip you want to show for the text area. Then at server runtime or in HTML results, when you hover the mouse over the text area, the tooltip will be displayed.
5. Set the character width in the Display Width field.
6. Specify the number of lines that the text area displays in the Number of Lines field, and choose whether or not to wrap the text by selecting true or false from the Auto Wrap drop-down list.
7. Check the **Fixed Only** option if you would like to set this text area to be read only.
8. Check the **Disabled** checkbox if you want to make the text area in the Web Behaviors box.
9. If needed, bind some web actions to the text area in the Web Behaviors box.
10. Upon finishing, click the **OK** button to close this dialog.

Checkbox/Radio button

The way to insert and define the values of a checkbox and radio button is the same.

To insert a checkbox/radio button and define its values:

1. Do one of the following to insert the checkbox/radio button:
   - Drag **Checkbox/Radio Button** from the Toolbox panel to the destination.
   - On the menu bar, click **Insert > Web Controls > Checkbox/Radio Button**, then click the mouse button in the destination.
   - On the Insert toolbar, click the down arrow on the **Insert Control Objects** button and select **Checkbox/Radio Button** from the drop-down list, then click the mouse button in the destination.
2. Right-click the checkbox/radio button and select **Display Type** from the shortcut menu. The Display Type dialog appears.



3. In the Web Options panel, specify the name, value of the checkbox/radio button in the Name and Value fields.
4. In the Tool Tip field, enter the tip you want to show for the checkbox/radio button. Then at server runtime or in HTML results, when you hover the mouse over the checkbox/radio button, the tooltip will be displayed.
5. Set Initially Checked to **true** if you want the checkbox/radio button to be selected by default.
6. Check the **Disabled** checkbox if you want to make the checkbox/radio button disabled.
7. If needed, bind some web actions to the checkbox/radio button in the Web Behaviors box.
8. Upon finishing, click **OK** to close this dialog.

Image Button

To insert an image button and define its values:

1. Do one of the following to insert the image button:
   - Drag **Image Button** from the Toolbox panel to the destination.
   - On the menu bar, click **Insert > Web Controls > Image Button**, then click the mouse button in the destination.
   - On the Insert toolbar, click the down arrow on the **Insert Control Objects** button and select **Image Button** from the drop-down list, then click the mouse button in the destination.
2. Right-click the image button and select **Display Type** from the shortcut menu. The Display Type dialog appears.

3. In the Web Options panel, click the **Browse** button to specify the image source. For a DBField, formula, or a summary, you can also check the **From DBField** radio button to make the value of the DBField/formula/summary the image source. Then select the DBField from the Decode Type drop-down list to enabled fields which you want to display the image.
4. Enter the name, value of the image button in the Name and Value fields.
5. Type a string in the Alternate Text field to serve as content when the image cannot be rendered.
6. In the Tool Tip field, enter the tip you want to show for the image button. Then at server runtime or in HTML results, when you hover the mouse over the image button, the tooltip will be displayed.
7. Check the **Disabled** checkbox if you want to make the image button disabled.
8. Specify the scaling mode of the image button from the Scaling Mode drop-down list, and the maximum scaling ratio in the Max Ratio field. By default the scaling ratio of the image button is not limited. If it is set to a positive number, the actual scaling ratio is less than or equal to it.
9. Specify the size of the image button.
   - If you choose to use the original size of the image, check the **Original Size** option.
   - If you want to use customized image size, uncheck **Original Size**, and then set the display width and height in the Width and Height drop-down lists. Check the **Constrain Proportions** option if you want to keep the original proportion of the image when you specify the width and height.
10. If needed, bind some web actions to the image button in the Web Behaviors box.
11. Upon finishing, click the **OK** button to close the dialog.

Button

A button for submitting or resetting a form.

To insert a button and define its values:

1. Do one of the following to insert the button:
   - Drag **Button** from the Toolbox panel to the destination.
   - On the menu bar, click **Insert > Web Controls > Button**, then click the mouse button in the destination.
   - On the Insert toolbar, click the down arrow on the **Insert Control Objects** button and select **Button** from the drop-down list, then click the mouse button in the destination.
2. Right-click the button and select **Display Type** from the shortcut menu. The Display Type dialog appears.

3. In the Web Options panel, enter the name, value of the button in the Name and Value fields.
4. In the Tool Tip field, enter the tip you want to show for the button. Then at server runtime or in HTML results, when you hover the mouse over the button, the tooltip will be displayed.
5. Select the action of the button: None, Submit Form or Reset Form.
6. Check the **Disabled** checkbox if you want to make the button disabled.
7. If needed, bind some web actions to the button in the Web Behaviors box.
8. Upon finishing, click **OK** to close this dialog.

List/Drop-down list

List and drop-down list are generally called multivalue comments. A list or drop-down list provides a list of values for users to select from. While, a drop-down list also includes a smaller space than a list box so that report can be laid out more compactly.

To insert a list/drop-down list and define its values:

1. Do one of the following to insert the list/drop-down list:
   - Drag **List/Drop-down List** from the Toolbox panel to the destination.
   - On the menu bar, click **Insert > Web Controls > List/Drop-down List**, then click the mouse button in the destination.
   - On the Insert toolbar, click the down arrow on the **Insert Control Objects** button and select **List/Drop-down List** from the drop-down list, then click the mouse button in the destination.
2. Right-click the list/drop-down list and select **Display Type** from the shortcut menu. The Display Type dialog appears.

3. In the Web Options panel, enter a name for the list/drop-down list in the Name field.
4. In the Tool Tip field, enter the tooltip you want to show for the list/drop-down list. Then at server runtime or in HTML results, when you hover the mouse over the list/drop-down list, the tooltip will be displayed.
5. Specify the item labels and the value of each item in the items box by inputting some strings as the labels and values separately.
   - If you want to use a DBField/formula/summary field to control the value, click ☑ in the Value column to insert one with the Insert Fields dialog, and then select the format for the inserted field from the drop-down list in the Insert Fields dialog. Check the **Use Runtime Value** option if you want to use the value of the field to be the value of the item. If the field is a summary field or formula field, at design time or runtime, all fields defined on the list/drop-down list will not take effect, and if you applied some other web action that needs value from the list/drop-down list, a fault value will be returned.
6. If the display type is defined as List, you can check the **Allow Multiple Selections** checkbox if you want to allow multiple items to be selected.
7. In the Selected field, specify the selected value for the list/drop-down list.
8. If you want to disable the list/drop-down list, check the **Disabled** option.
9. If needed, bind some web actions to the list/drop-down list in the Web Behaviors box.
10. When done, click **OK** to apply the settings.

After a basic web control has been inserted, you can customize its appearance and behavior by setting properties in the Report Inspector, which can be saved as a CSS style for future use (for details, see Creating a CSS style).

Notes:

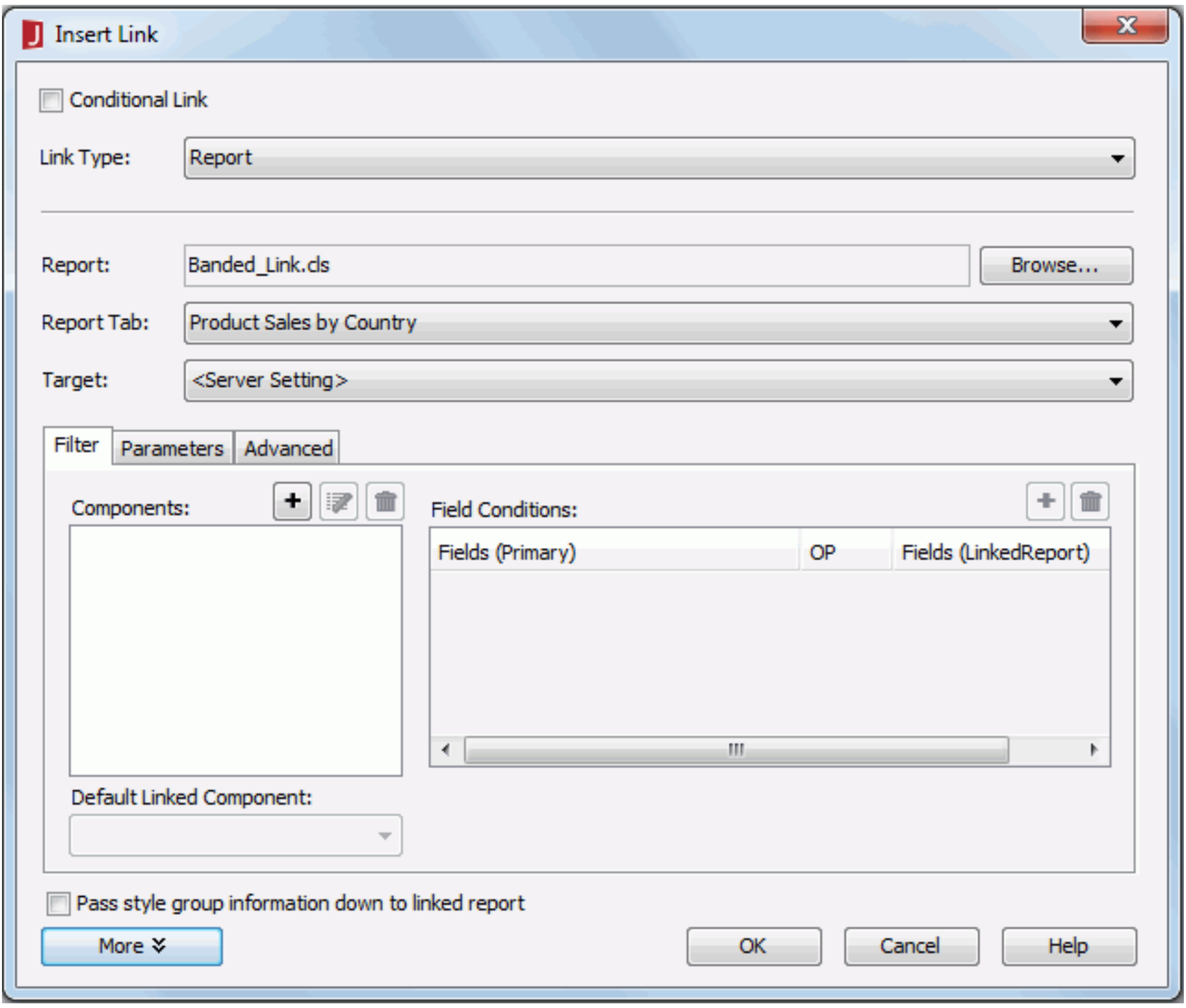
- The display type of a basic web control can also be changed after it has been created. Basic web controls other than list and drop-down list can be converted to each other, and can be displayed as a general type. You may need to change the value of the property "Value" to the property value "Label" in the Report Inspector, and vice versa. Some basic web controls are labels in nature.
- List and drop-down list can be converted to each other, which cannot be displayed as any other type.
- DBfields, parameter fields, formula fields, summary fields, special fields, and images can also be displayed as a text box, and can be set to read only. Text box can also be displayed as a read only text box.
- Some of the basic web control values can be controlled by formulas. To control a value by a formula, click the formula ☑ behind the text field of the value and then select a formula from the drop-down list. If the given formulas cannot meet your requirements, click the <CREATE FORMULA> item in the list to create one as required.
- If you have specified a tooltip for any basic web control, the tooltip is displayed differently on different browsers:
  - On Internet Explorer, the tip text is automatically wrapped if its length exceeds the maximum tip width the browser allows. In addition, for Internet Explorer, you can manually make some text displayed in a new line by adding &#10 or &#13 ahead of the text when editing it in the Tool Tip field.
  - On Firefox, the tip text is displayed in one line with the maximal tip width the browser allows, and the text that cannot be displayed within the width will be cut off and replaced by ellipsis.

Binding a link to a basic web control

You can make a basic web control link with another report, a location specified by a URL or an e-mail address, or a Blob data type field. The link can either be a simple link or a conditional link. With conditional link, different targets can be loaded based on different conditions.

To bind a link to a basic web control:

1. Right-click the web control and select **Link** from the shortcut menu. The Insert Link dialog appears.



2. In the Insert Link dialog, specify whether it is a conditional link or a simple link, then select the link target and set the options for the link target as required.

3. When done, click **OK**.

For details about different link targets, see Binding a link to a label.

Applying web actions to a basic web control

A web action can be bound to a basic web control, and then when the trigger event, such as click occurs, the web action will be performed. This allows you to customize a web control to make it respond to some events, and execute corresponding actions, such as sorting and filtering. For example, you can insert a button in JReport Designer and associate an action with the button's onclick event. Then, when you view the report at runtime and click the button, the defined operation will be executed.

To bind web actions to a basic web control:

1. Right-click the web control and select **Display Type** from the shortcut menu.
2. In the Web Behaviors box of the Display Type dialog, choose an event from the Events column, then click in the Actions column and click that appears in the text box.
3. In the Web Action List dialog, select the required action and then click **OK**.
4. Click in the Display Type dialog to add more web behavior lines and specify the events and actions accordingly. If a web behavior is not required, click to remove it.
5. Adjust the order of the added web actions by clicking or . Then, when an event that has been bound with more than one action happens, the upper action will be triggered first.
6. When done , click **OK** to accept the settings.

For more information about web actions, see Applying web actions to a label.

Example of applying web actions

In the following example, a sample report is built for guiding you through the process of using web controls and applying web actions step by step.

Step 1: Create the sample report

1. Design a standard banded report based on the WorldWideSales query in the SampleReports.cat catalog (make sure to remove the report title in the New Page Report dialog), which shows the following detail fields: Country, Product Type Name, Product Name, Unit Price, Quantity, Discount, and the formula Total (`@"Unit Price" * @Quantity - @"Unit Price" * @Quantity * @Discount/100`), applies the filter "COUNTRY = China AND PRODUCT TYPE NAME = Decaf", and uses the Classic style.
2. Insert a crosstab into the report and place it below the banded object. The crosstab shares the same dataset with the banded object, comprises the column field: Product Type Name, row field: Category, and aggregate field: Total (aggregate function: Sum) and applies the Classic style.
3. Click **View** > **Page Header** on the menu bar to have this panel visible, then set its Height property to **1.0** in the Report Inspector to hold all the web controls that will be inserted in the report later.
4. Save the report.

Step 2: Apply a web action without parameters

First, we want to allow the report user to click a button in the report to show the Search dialog of Page Report Studio. To achieve this:

1. Select the page header panel and click **Insert** > **Web Controls** > **Button** to insert a Button web control in it.
2. Right-click the button, and select **Display Type** from the shortcut menu.
3. In the Web Behaviors box of the Display Type dialog, specify event to **Click**, then click in the Actions column and click that appears in the text box.
4. In the Web Action List dialog, choose **user_showSearchDialog** to be web action of the button, then confirm the settings.
5. Go to the Report Inspector and set the Text property of this button to **Search**.
6. Save the report.

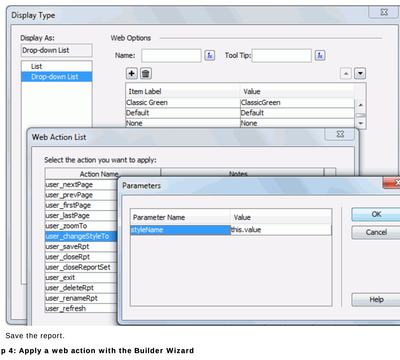
Step 3: Apply a web action with parameters

Now we want to allow the report user to rotate the crosstab in the report and to change the report style.

1. Insert another Button web control in the page header panel of the report and place it next to the Search button.
2. Specify the **Click** event and **user_rotateCrosstab** action to the button as explained above, then set **&CTCrossTab** to be the value of the action parameter instanceName (CTCrossTab is the name of the crosstab we can see in the report structure panel of the Report Inspector, and here the symbol & shall be prefixed to the crosstab name).
3. In the Report Inspector, edit the Text property of the button to **Rotate Crosstab**.
4. Insert a Label object in the page header panel and place it below the two buttons, then resize the label and edit its text to "**Change Report Style To:**".
5. Insert a Drop-down List web control next to the label.
6. Right-click the drop-down list and click **Display Type** shortcut menu.
7. In the Display Type dialog, add these items and corresponding values to the drop-down list (to add an item line, click):

Item Label	Value
Classic	Classic
Classic Blue	ClassicBlue
Classic Green	ClassicGreen
Default	Default
None	None

Then, in the Web Behaviors box of the dialog, set Event to **Data Change**, Action to **user_changeStyleTo**, and specify the value for the web action parameter styleName to **this.value**.



8. Save the report.

Step 4: Apply a web action with the Builder Wizard

The actions Filter, Sort and Parameter have a Builder Wizard. Now we would like to use the wizard to allow the report user to sort records in the banded object by one of the specified fields.

1. Insert another Drop-down List web control into the page header panel, and place it below the above web controls. The drop-down list is displayed as MultiValueContainer1 in the Report Inspector.
2. Right-click the drop-down list and click **Display Type** shortcut menu.
3. In the Display Type dialog, add these items and corresponding values to the drop-down list:

Item Label	Value
Unit Price	UNIT PRICE
Quantity	QUANTITY

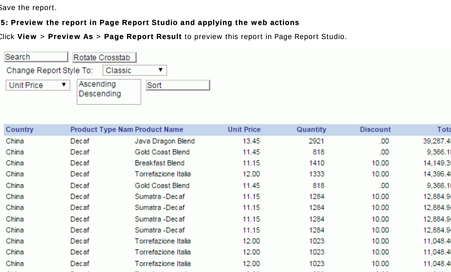
4. Insert a List web control next to the drop-down list. The list is displayed as MultiValueContainer2 in the Report Inspector.
5. In the Display Type dialog for this list, add these items and corresponding values:

Item Label	Value
Ascending	Ascending
Descending	Descending

6. Insert a Button web control and place it next to the list, then right-click the button and click **Display Type** on the shortcut menu.
7. In the Display Type dialog, specify Event to **Click** and Action to **Sort**, then in the Sort - Web Action Builder dialog, select **MultiValueContainer1** from the Sort On column, and **MultiValueContainer2** from the Sort Value column. Confirm the settings, and go to the Report Inspector to modify the Text property of the button to **Sort**.
8. Save the report.

Step 5: Preview the report in Page Report Studio and applying the web actions

1. Click **View** > **Preview As** > **Page Report Result** to preview this report in Page Report Studio.



Country	Product Type Nam	Product Name	Unit Price	Quantity	Discount	Total
China	Decaf	Java Dragon Blend	13.45	2921	.00	39,287.45
China	Decaf	Gold Coast Blend	11.45	818	.00	9,366.10
China	Decaf	Breakfast Blend	11.15	1410	10.00	14,149.35
China	Decaf	Torrefazione Italia	12.00	1333	10.00	14,396.40
China	Decaf	Gold Coast Blend	11.45	818	.00	9,366.10
China	Decaf	Sumatra -Decaf	11.15	1284	10.00	12,884.94
China	Decaf	Sumatra -Decaf	11.15	1284	10.00	12,884.94
China	Decaf	Sumatra -Decaf	11.15	1284	10.00	12,884.94
China	Decaf	Sumatra -Decaf	11.15	1284	10.00	12,884.94
China	Decaf	Torrefazione Italia	12.00	1023	10.00	11,048.40
China	Decaf	Torrefazione Italia	12.00	1023	10.00	11,048.40
China	Decaf	Torrefazione Italia	12.00	1023	10.00	11,048.40
China	Decaf	Torrefazione Italia	12.00	1023	10.00	11,048.40

	Decaf	Regular	Total
Blends	539,283.36	1,299,225.06	1,838,508.42
Bold	547,035.08	1,251,982.83	1,799,017.90
Espresso	317,954.32	142,730.28	460,684.61
Exotic	56,847.53	1,506,554.07	1,563,401.60
Flavored	726,826.80	1,359,545.65	2,086,372.45
Mild	1,844,101.35	480,128.25	2,324,229.60
Total	4,032,048.45	6,040,166.13	10,072,214.59

2. Click the **Search** button and the Search dialog appears.
3. Click the **Rotate Crosstab** button, the columns and rows of the crosstab are rotated.

	Blends	Bold	Espresso	Exotic	Flavored	Mild	Total
Decaf	539,283.36	547,035.08	317,954.32	56,847.53	726,826.80	1,844,101.35	4,032,048.45
Regular	1,299,225.06	1,251,982.83	142,730.28	1,506,554.07	1,359,545.65	480,128.25	6,040,166.13
Total	1,838,508.42	1,799,017.90	460,684.61	1,563,401.60	2,086,372.45	2,324,229.60	10,072,214.59

4. Select a style from the Change Report Style To drop-down list, for example, select **Classic Green**, and the report style now changes to the selected one. It affects both the banded object and the crosstab.

Country	Product Type Nam	Product Name	Unit Price	Quantity	Discount	Total
China	Decaf	Java Dragon Blend	13.45	2921	.00	39,287.45
China	Decaf	Gold Coast Blend	11.45	818	.00	9,366.10
China	Decaf	Breakfast Blend	11.15	1410	10.00	14,149.35
China	Decaf	Torrefazione Italia	12.00	1333	10.00	14,396.40
China	Decaf	Gold Coast Blend	11.45	818	.00	9,366.10
China	Decaf	Sumatra -Decaf	11.15	1284	10.00	12,884.94
China	Decaf	Sumatra -Decaf	11.15	1284	10.00	12,884.94
China	Decaf	Sumatra -Decaf	11.15	1284	10.00	12,884.94
China	Decaf	Torrefazione Italia	12.00	1023	10.00	11,048.40
China	Decaf	Torrefazione Italia	12.00	1023	10.00	11,048.40
China	Decaf	Torrefazione Italia	12.00	1023	10.00	11,048.40

	Blends	Bold	Espresso	Exotic	Flavored	Mild	Total
Decaf	539,283.36	547,035.08	317,954.32	56,847.53	726,826.80	1,844,101.35	4,032,048.45
Regular	1,299,225.06	1,251,982.83	142,730.28	1,506,554.07	1,359,545.65	480,128.25	6,040,166.13
Total	1,838,508.42	1,799,017.90	460,684.61	1,563,401.60	2,086,372.45	2,324,229.60	10,072,214.59

Note: You may notice that the value of the drop-down list does not change to the selected one after the report style is updated. To update the drop-down list value to the current report style, you need to append another action user_changeCompProperty to the event Data Change to the drop-down list.

5. Select a field name from the third drop-down list and then the sorting direction from the list. Here we select **Unit Price** and **Descending** respectively, then click the **Sort** button.

The records in the banded object now are sorted based on the Unit Price values descendingly.

Country	Product Type Nam	Product Name	Unit Price	Quantity	Discount	Total
China	Decaf	Java Dragon Blend	13.45	2921	.00	39,287.45
China	Decaf	Torrefazione Italia	12.00	1333	10.00	14,396.40
China	Decaf	Torrefazione Italia	12.00	1023	10.00	11,048.40
China	Decaf	Torrefazione Italia	12.00	1023	10.00	11,048.40
China	Decaf	Torrefazione Italia	12.00	1023	10.00	11,048.40
China	Decaf	Torrefazione Italia	12.00	1023	10.00	11,048.40
China	Decaf	Gold Coast Blend	11.45	818	.00	9,366.10
China	Decaf	Gold Coast Blend	11.45	818	.00	9,366.10
China	Decaf	Breakfast Blend	11.15	1410	10.00	14,149.35
China	Decaf	Sumatra -Decaf	11.15	1284	10.00	12,884.94
China	Decaf	Sumatra -Decaf	11.15	1284	10.00	12,884.94
China	Decaf	Sumatra -Decaf	11.15	1284	10.00	12,884.94
China	Decaf	Sumatra -Decaf	11.15	1284	10.00	12,884.94

Making basic web controls work in a form

In order to make web controls accept user inputs and submit requests to JReport Server, you must add the web controls to a form and then define some properties of the form.

Creating a form

To create a form in a report, you can do either of the following:

- Click **Insert** > **Web Controls** > **Form**.
- Drag **Form** from the Toolbox panel to the report.

The following are commands on the shortcut menu of a form:

- **Unmark Form:** Deselects this form.
- **Delete Form:** Deletes this form.
- **Remove All Controls:** Removes all the objects added to the form.
- **Properties:** Shows the properties of the form.

Adding an object to a form

To add an object to a form, right-click the object, and on the shortcut menu, select the required item from the Add to Form submenu. The object can be added to a new form or an existing one. In JReport Designer, the icon represents a form. If you click the icon, you will find that the form and all the objects added to it are enclosed by a red dashed outline.

After an object has been added to a form, if you want to remove it from the form, you can right-click the object and select **Remove from Form** from the shortcut menu.

Making web controls work in a form

After adding web controls to a form, you need to make the web controls cooperate, that is, you must set some properties of the form so as to organize the web controls and define their working mode. To do this, select the form, open the Report Inspector, and then edit the properties of the form.

Here is an example illustrating the use of form. In this example, a text field and a button in a report enable the report user to search for a particular string with Google.

1. Create a report which contains a blank flow report tab.
2. In the report tab, insert two web controls: a Text Field and a Button.
3. Right-click the text field, select **Display Type** from the shortcut menu, then in the Display Type dialog, set the following options (leave other options to the default values):

   Name: q
   Value: "" (null)
   Max Length: 30

4. Right-click the button, click **Display Type** on the shortcut menu and set the following options in the Display Type dialog:

   Value: Google
   Action: Submit Form

5. Select the two web controls, then right-click and select **Add to Form** > **Add to New Form** from the shortcut menu.
6. Set the properties of the form in the Report Inspector as follows:

   Form Action: http://www.google.com/search
   Form Method: get
   Form Target: blank

7. Save the report and preview it in Page Report Studio.

   Now you can input a string up to 30 characters in the text field, and then click the button **Google** to search for that string. The search result will be displayed in a new browser window.

Note: For a table, if you add a field to a new form, the form will be created in the first cell of the table row which contains the field, and fields in the previous table rows cannot be added to this newly-created form. If you add columns before the first cell, then fields in these columns of the same table row also cannot be added to the form.

Using advanced web controls

Advanced web controls refer to Expand/Collapse Group, Parameter Control, Parameter Form Control, Filter Control, Slider and Navigation Control. They are defined with specific web actions and are used for specific purposes.

The advanced web controls can be inserted into the report areas listed in Component placement (except the table cells in web reports).

This section describes each of the web controls and how to use them in detail.

- Expanding/Collapsing group records
- Using parameter control to specify a parameter to a report
- Using parameter form control to run reports
- Using slider or filter control to filter data
- Using navigation control to undo/redo value selection in filter controls
- Customizing dialogs to perform web actions

Note: After an advanced web control has been inserted, you can customize its appearance and behavior by setting properties in the Report Inspector, which can be saved as a CSS style for future use (for details, see Creating a CSS style).

Expanding/Collapsing group records

You can insert the web control, Expand/Collapse Group, to the group header panel of a banded object so that end users can use the web control to expand/collapse records in groups of the banded object. For details, refer to Managing the data of a banded object.

Using parameter control to specify a parameter to a report

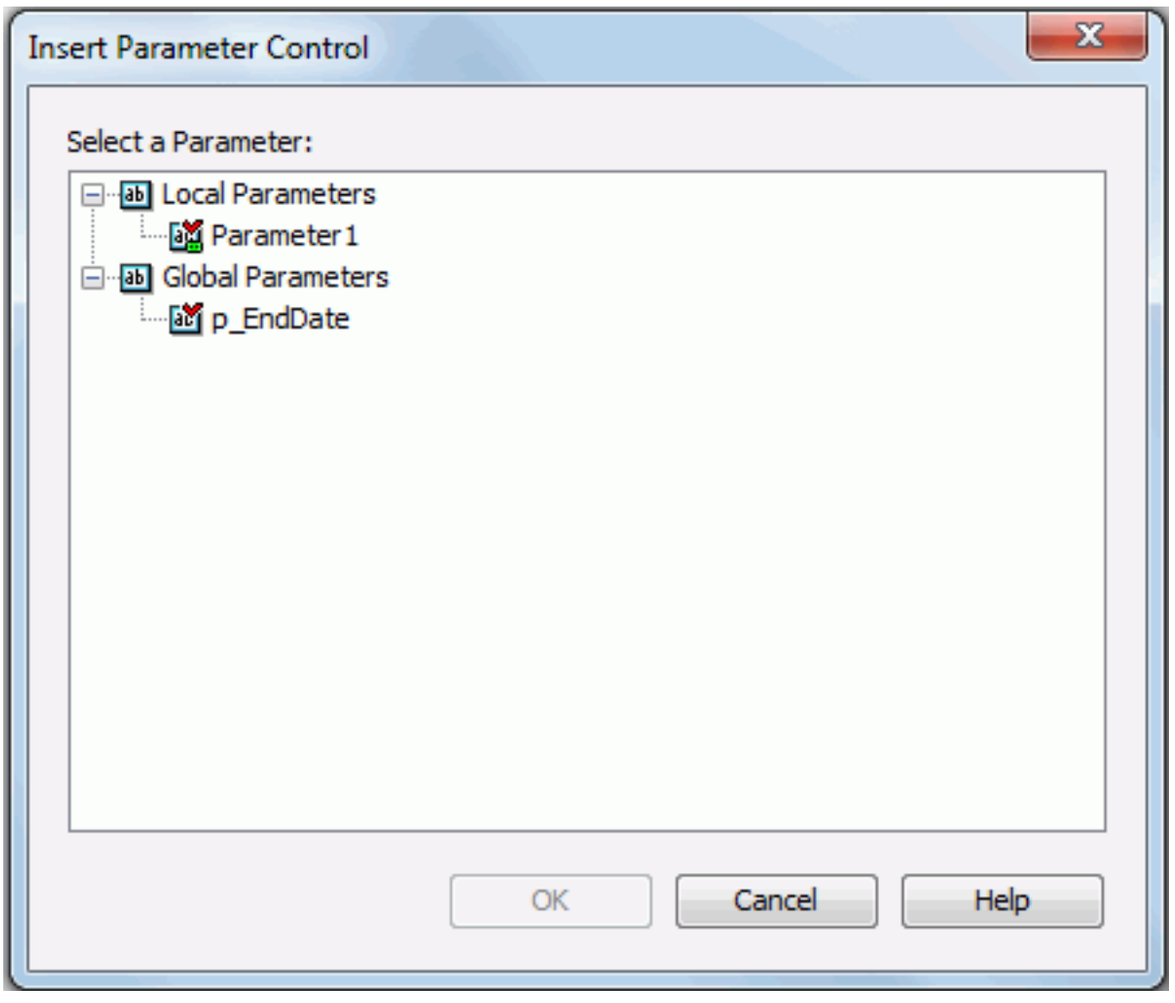
A parameter control is a web control that is bound with a parameter used by the current report. By specifying values to the parameter in a parameter control at runtime, end users can pass the parameter values to JReport and run the report with the specified values.

Parameter controls do not support inserting cascading parameters. If you want to do this, use parameter form controls instead.

To insert a parameter control into a report:

1. Do one of the following:
   - Drag **Parameter Control** from the Toolbox panel to the destination.
   - Click **Insert** > **Web Controls** > **Parameter Control** on the menu bar, then click the mouse button in the destination.
   - On the Insert toolbar, click the down arrow on the **Insert Control Objects** button and select **Parameter Control** from the drop-down list, then click the mouse button in the destination.

   The Insert Parameter Control dialog appears.



2. In the dialog, specify the parameter you want to add to the parameter control from the Select a Parameter box, which lists all the parameters except cascading parameters used in the current report.
3. When done, click **OK** to insert the parameter control into the desired destination.

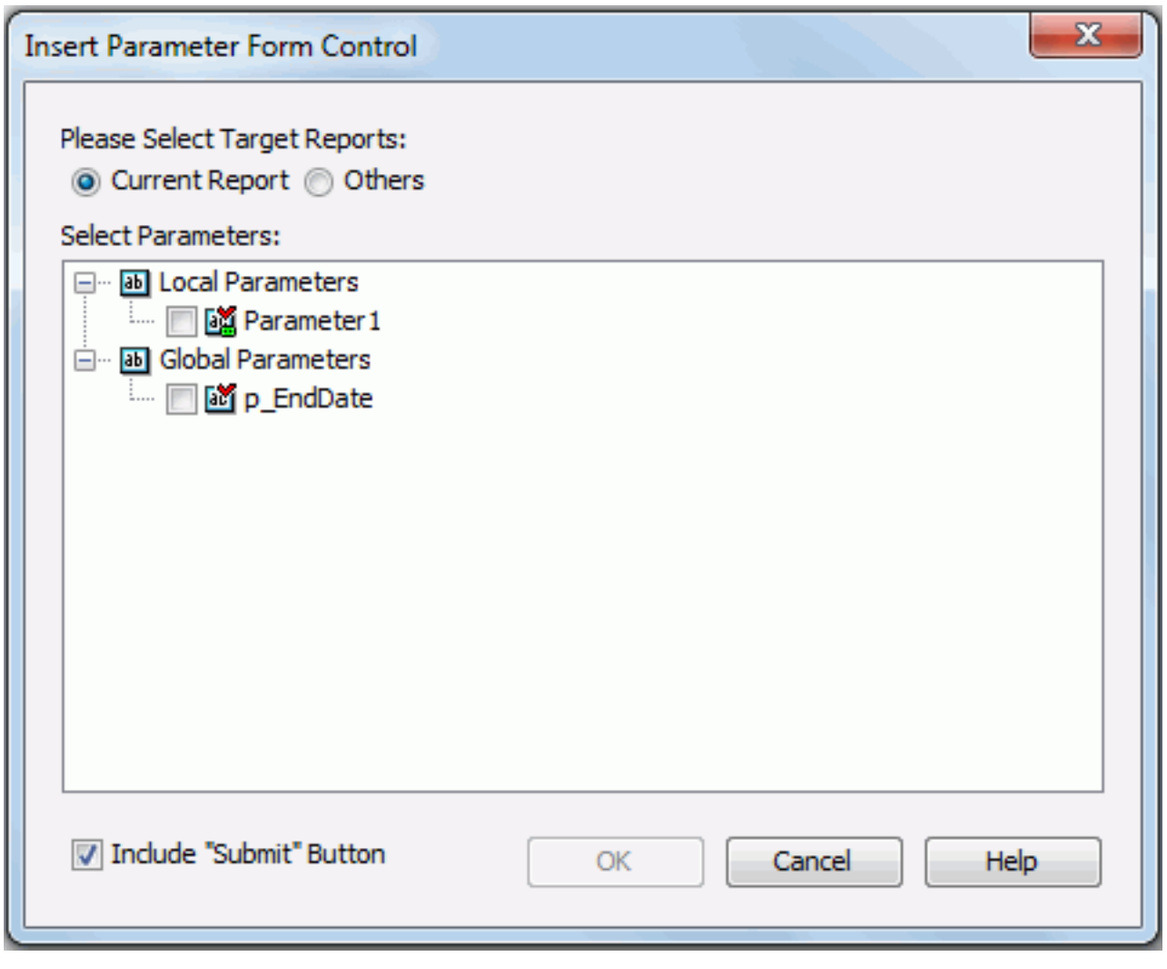
Using parameter form control to run reports

A parameter form control is a web control that is bound with the parameters used by the current report or other reports. By specifying values to the parameters in a parameter form control at runtime, end users can make the reports run with the specified parameter values.

To insert a parameter form control into a report:

1. Do one of the following:
   - Drag **Parameter Form Control** from the Toolbox panel to the destination.
   - Click **Insert** > **Web Controls** > **Parameter Form Control** on the menu bar, then click the mouse button in the destination.
   - On the Insert toolbar, click the down arrow on the **Insert Control Objects** button and select **Parameter Form Control** from the drop-down list, then click the mouse button in the destination.

   The Insert Parameter Form Control dialog appears.



2. Specify the target reports to run using the parameter form control.
   - To run the current report, select **Current Report**, then specify the parameters used to run the report from the Select Parameters box.
   - To run other reports, select **Others**, then select the reports you want to run. If all the selected reports contain no parameters, you cannot finish the dialog.
3. Specify whether to include the Submit button in the parameter form control. If Submit is included, it is used to submit the parameter values end users specify in the parameter form control. If Submit is not included, once end users change the value of a parameter in the parameter form control, the new value will be applied automatically.
4. When done, click **OK** to insert the parameter form control into the desired destination.

Note: If you save a report containing a parameter form to another directory or publish it to local directory or to JReport Server, the reports that you selected in the parameter form in order to run will not be saved or published along with the report.

Using slider or filter control to filter data

A slider or filter control is used to filter one or more data components in a report or library component. It can do filtering based on the fields in the data sources the data components are created from.

Filter controls allow you to pick one or more random values from a list and are used with categorical or nominal variables. You can choose one or more values from anywhere in the list and there is no mean or median value calculation possible such as States and Countries.

Sliders allow you to pick one or more sequential values from a list and are used for interval variables such as dates, times, quantity and currency variables where the slider represents the scale from lowest to highest value and the middle represents the median value.

Think about what you need and choose the proper filtering tool.

Filtering scenarios

Both sliders and filter controls have the same filtering mechanism.

Filtering based on one field is a common usage. Bind a field to a slider or filter control, and then based on the field to filter the data of the components created from the same data source as the field.

Another special usage is to filter components using different data sources. Choose a common field all the data sources contain and then bind a slider or filter control with the common field in all the data sources, that is, in the insertion dialog, select the common field under the nodes of all the data sources.

For example, there are two components containing data from different data sources, and you want to filter their data using one filter tool. The precondition is that both data sources have the field you want to filter. For example, you would like the two components to show the data of a specific country. In order to do this, insert a slider or a filter control according to your requirement into the report, then in the insertion dialog, select both the country fields from the two data sources (different data sources may use different names for the country field, for example, data source 1 uses "Country", data source 2 uses "P_Country", in this case, you need to select both "Country" and "P_Country"), then from the Apply To drop-down list, select the two components you want to filter, then click OK. The filter tool will be inserted in the report, and you can see it lists country names which come from the two data sources. In the filter tool select one or more countries, then the two target components will be filtered and only display the data of these selected countries.

When you bind a slider or filter control with multiple different fields, be sure the list of values in each field match so that when you select a value, such as Country, P_Country and S_Country, it will match the appropriate country field in each data source. The logic is the values are OR that is *Field1=SelectedValue1 or Field2=SelectedValue1 or Field3=SelectedValue1*. Therefore, when Field1, Field2 and Field3 are used in different components you will see the records correctly in each components.

When you bind a slider or filter control with multiple different fields which do not contain the same list of values such as Country, Region and Territory, at runtime after you select values in the slider or filter control, the filter condition will use OR logic to apply the selected values to all the fields of the slider or filter control, for example, *Field1=SelectedValue1 or Field2=SelectedValue1 or Field3=SelectedValue1*. In this case, when there are three fields but the list of values in each field do not match, the result will have no matching records for two of the components and therefore become blank components. We recommend that you use three different filter tools in cases like this.

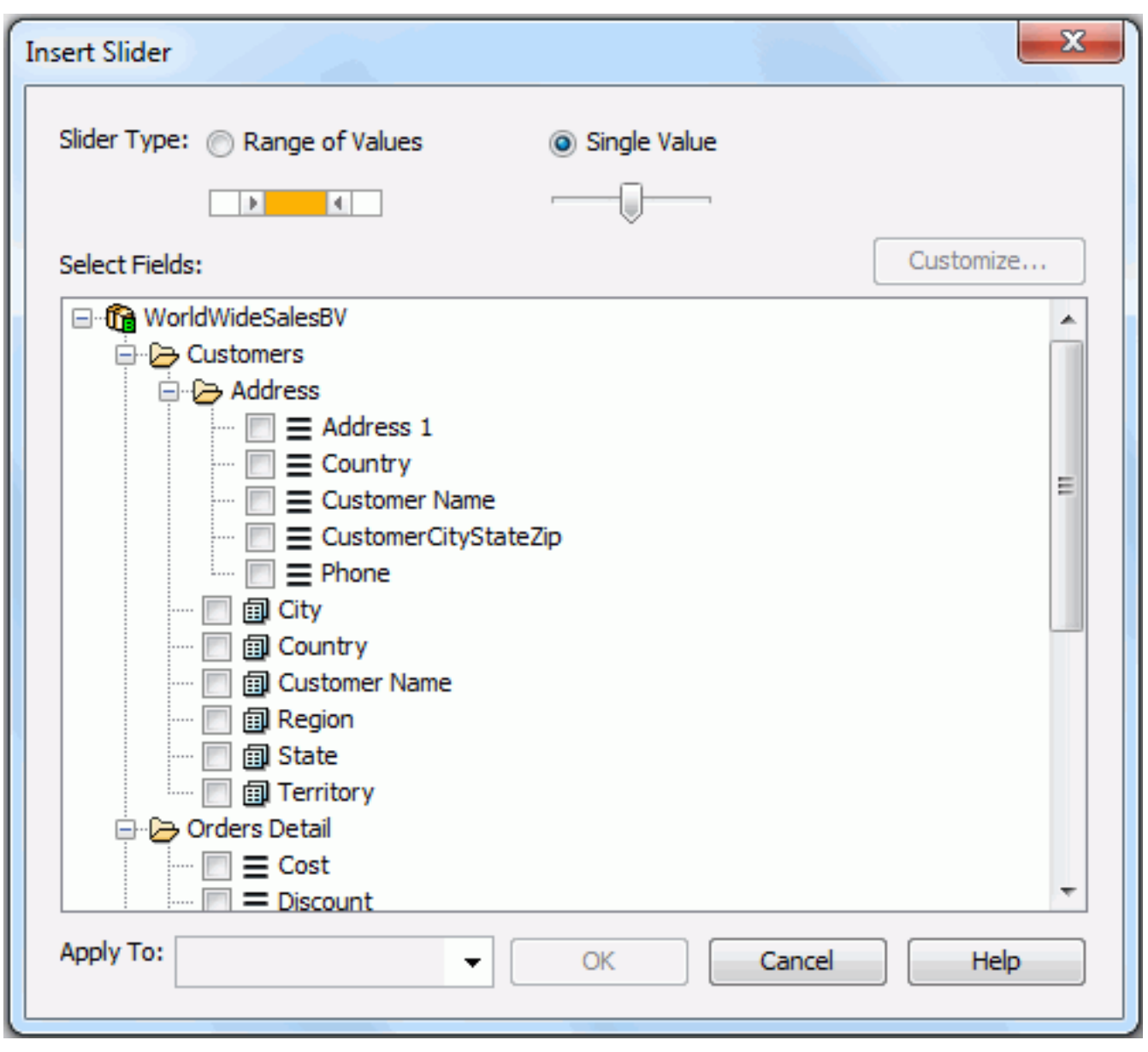
Using slider

Slider is only available for library component at present and can only be inserted into the body of a library component.

To insert a slider into a library component:

1. Do one of the following:
   - Drag **Slider** from the Toolbox panel to the destination.
   - Click **Insert** > **Web Controls** > **Slider** on the menu bar, then click the mouse button in the destination.
   - On the Insert toolbar, click the down arrow on the **Insert Control Objects** button and select **Slider** from the drop-down list, then click the mouse button in the destination.

   The Insert Slider dialog appears.



2. Choose whether the slider is used to specify a range of values or just a single value.
3. From the resource list in the Select Fields box, select the fields of the same data type to bind to the slider.

   To filter components created from the same data source, select a field in the data source.

   To filter components created from different data sources via one slider, find a common field these data sources contain, then select the field in each of the data sources.
4. By default, all the values of the selected fields will be available for the slider, which may be too many for a slider. However you can customize the values to show. To do this:
   a. Click the **Customize** button to display the Customize Value dialog.
   b. Uncheck the **Select All** option. You can then customize the values.
      - To specify values one by one, click to add a value line, then type a value or select a value from the drop-down list. Repeat the operation to add more values. To remove a value, select the line and click .
      - To specify the value range, specify a value in the From and To text fields respectively.
   c. For fields of Date/Time type, you can also specify a special function.
   d. Click **OK** to save the customized values. The customized values will be available on the slider for choosing.
5. The Apply To drop-down list provides the components involving the selected fields. Select the components which you want to filter.
6. When done, click **OK** to insert the slider. Then at runtime, end users can select the values on the slider to filter the specified data components.

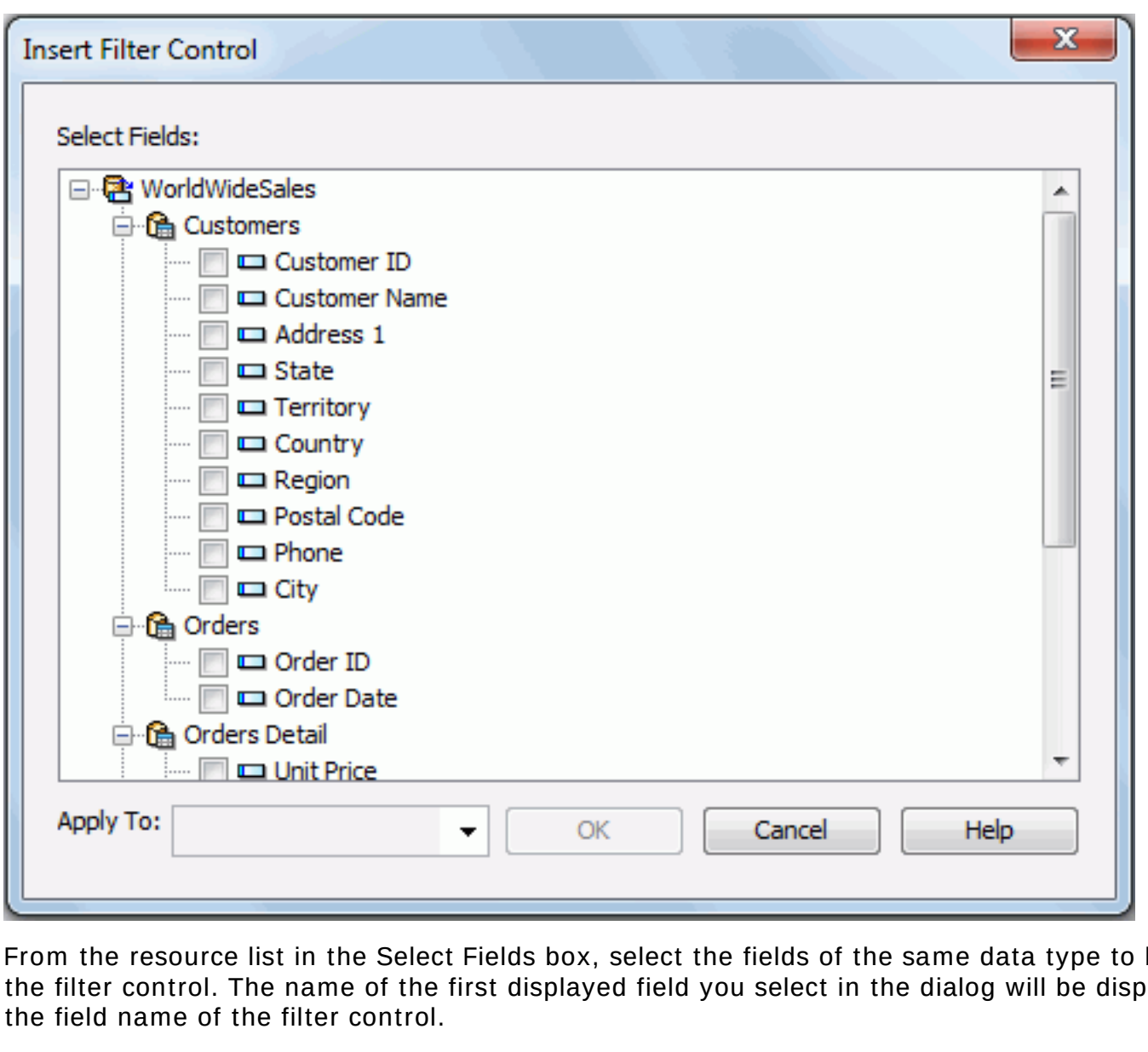
After a slider is inserted into a library component, you can further modify it if necessary. To do this, right-click the slider and select **Edit Slider** from the shortcut menu. In the Edit Slider dialog, edit the slider according to your requirements.

Using filter control

To insert a filter control into a report:

1. Do one of the following:
   - Drag **Filter Control** from the Toolbox panel to the destination.
   - Click **Insert** > **Web Controls** > **Filter Control** on the menu bar, then click the mouse button in the destination.
   - On the Insert toolbar, click the down arrow on the **Insert Control Objects** button and select **Filter Control** from the drop-down list, then click the mouse button in the destination.

   The Insert Filter Control dialog appears.



2. From the resource list in the Select Fields box, select the fields of the same data type to bind to the filter control. The name of the first displayed field you select in the dialog will be displayed as the field name of the filter control.

   To filter components created from the same data source, select a field in the data source.

   To filter components created from different data sources via one filter control, find a common field these data sources contain, then select the field in each of the data sources.
3. The Apply To drop-down list provides the components involving the selected fields. Select the components which you want to filter.
4. When done, click **OK** to insert the filter control. Then at runtime, all values of the specified fields will be listed in the filter control. End users can select one or more values to apply.

When a filter control is inserted into a report, you can further modify it if necessary. To do this, right-click the filter control and select **Edit Filter Control** from the shortcut menu. In the Edit Filter Control dialog, edit the filter control according to your requirements.

After inserting filter controls in a report, you can also insert a navigation control for undoing/redoing the value selection in the filter controls. For details about the usage of navigation control, see Using navigation control to undo/redo value selection in filter controls.

Using navigation control to undo/redo value selection in filter controls

A navigation control can be considered as an accessorial control for filter controls and is used to deal with the value selection operations in all the filter controls in the same report.

To insert a navigation control into a report, do either of the following:

- Drag **Navigation Control** from the Toolbox panel to the destination.
- Click **Insert** > **Web Controls** > **Filter Control** on the menu bar, then point to the destination where you want to add the navigation control and click the mouse button.
- On the Insert toolbar, click the down arrow on the **Insert Control Objects** button and select **Filter Control** from the drop-down list, then point to the destination and click the mouse button.

A navigation control is a combination of three buttons:

- **Back**
  Goes back to the previous value selection status and refreshes the report data accordingly.
- **Clear**
  Removes all the value selection histories and all the filter conditions based on the selections, and refreshes the report data accordingly.
- **Forward**
  Goes forward to the next value selection status and refreshes the report data accordingly.

By default, these three buttons are displayed as normal buttons, however, you can change their button type to be image button if necessary. To do this:

1. Select the target button, right-click it and select **Button Type** > **Image Button** from the shortcut menu.
2. In the Image Button dialog, click **Browse** to specify the path or input the URL of the image source directly in the text field.
3. Click **OK** to confirm.

If you want to get back the normal button type, you just need to right-click the target image button, and select **Normal Button** from the Button Type submenu.

Customizing dialogs to perform web actions

Currently JReport enables customizing a dialog to perform web actions such as sorting and filtering in a table of a page report at runtime, by using customized controls.

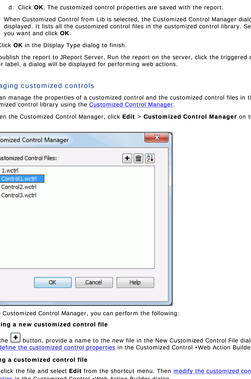
A customized control is a user defined web action dialog containing normal web controls that can process actions. The report designer first defines a customized control to be triggered from a component in a page report table in JReport Designer and then publish the report to JReport Server. Next the server end users can trigger the customized control which is shown as a dialog to perform web action as defined in the report.

In JReport Designer, you can store the properties of a customized control into a file with the extension .wctrl to the customized control library which is the Actions Customized folder under the installation root. JReport Server saves customized controls in DBMS.

Customized controls do not take part in the report layout and are not exported in the report results.

Defining customized control in JReport Designer

1. Right-click an object as a trigger in a table in a page report, for example, a column header label, and select **Display Type** from the shortcut menu.
2. In the Web Behaviors box of the Display Type dialog, choose an event from the Events column, for example, Click, then click in the Actions column and click that appears in the text box.
3. In the Web Action List dialog, select **Customized Control** to bind a customized control web action, or select **Customized Control from Lib** to reference a file from the library, and then click **OK**.
   - When Customized Control is selected, the Customized Control •Web Action Builder dialog is displayed.



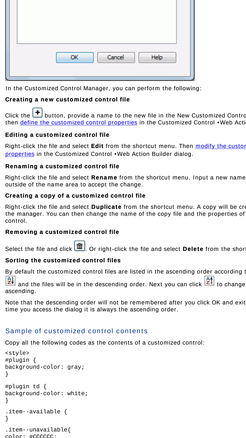
     a. Specify the size and position of the customized control shown in the browser window at runtime.
     b. In the Contents text box, input HTML fragment with JavaScript code. JavaScript in HTML fragment can use web action APIs. For details, refer to the index.html file in the <server_install_root>\public_html\webes\doc folder. There is a sample of the contents provided in the Sample of customized control contents section at the end of the doc.
     c. To save the customized control properties into a file to the customized control library, click **Save As** and then provide the file name.
     d. Click **OK**. The customized control properties are saved with the report.
   - When Customized Control from Lib is selected, the Customized Control Manager dialog is displayed. It lists all the customized control files in the customized control library. Select the one you want and click **OK**.
4. Click **OK** in the Display Type dialog to finish.

Next publish the report to JReport Server. Run the report on the server, click the triggered column header label, a dialog will be displayed for performing web actions.

Managing customized controls

You can manage the properties of a customized control and the customized control files in the customized control library using the Customized Control Manager.

To open the Customized Control Manager, click **Edit** > **Customized Control Manager** on the menu bar.



In the Customized Control Manager, you can perform the following:

Creating a new customized control file

Click the button, provide a name to the new file in the New Customized Control File dialog, and then define the customized control properties in the Customized Control •Web Action Builder dialog.

Editing a customized control file

Right-click the file and select **Edit** from the shortcut menu. Then modify the customized control properties in the Customized Control •Web Action Builder dialog.

Renaming a customized control file

Right-click the file and select **Rename** from the shortcut menu. Input a new name and then click outside of the name area to accept the change.

Creating a copy of a customized control file

Right-click the file and select **Duplicate** from the shortcut menu. A copy will be created and listed in the manager. You can then change the name of the copy file and the properties of the customized control.

Removing a customized control file

Select the file and click . Or right-click the file and select **Delete** from the shortcut menu.

Sorting the customized control files

By default the customized control files are listed in the ascending order according to their names. Click and the files will be in the descending order. Next you can click to change the order back to ascending.

Note that the descending order will not be remembered after you click OK and exit the dialog. The next time you access the dialog it is always the ascending order.

Sample of customized control contents

Copy all the following codes as the contents of a customized control:

```
<style>
#plugin {
background-color: gray;
}

#plugin td {
background-color: white;
}

.item--available {
}

.item--unavailable{
color: #CCCCC;
}

</style>
<table id="plugin" cellspacing="1px" style="margin:0px;width:100%;height:300px;
border:1px solid silver;background-color:;">
<tr style="height:30px;"><td>
<table style="text-align:center;width:100%;background-color:;">
<tr>
<td><button id="btnSortA" onclick="this.doSort('Ascending')">Ascending</button></td>
<td style="border-left:1px solid gray;"><button id="btnSortD" onclick="this.doSort
('Descending')">Descending</button></td>
</tr>
</table>
</td></tr>
<tr style="height:25px;"><td style="background-color:#cccccc;padding:3px;">
<input id="searchbox" style="width:85%;margin-right:3px;" type="text" onkeyup="this.
search()"/>
<div style="width:10px;height:10px;float:right;cursor:pointer;"
onclick="this.clearSearch()">x</div>
</td></tr>
<tr style="height:150px;"><td>
<div id="fc-values" onclick="this.selectValue(e)" style="width:100%;height:150px;
overflow:auto;"></div>
</td></tr>
<tr style="height:30px;">
<td style="background-color:#cccccc;"><button onclick="this.doFilter()">Apply</
button><button onclick="this.clear()">Clear</button></td>
</tr>
</table>

<script type="text/javascript">


var comp; // data container
var field; // db field
var filter;

var _onGetDataContainer = function(msg){
var result = msg.result;
if(result.err == 0){
comp = result.obj.comp;
if(!comp) return;
comp.getAssocDBField(context.dsid, _onGetDBField);
}else{
System.err.println(result.msg);
throw new Error(result.msg);
}
},$bind(this);

var _onGetDBField = function(msg){
var result = msg.result;
if(result.err == 0){
field = result.obj.field;
// TODO enable sort ?
Report.getFilterCtrl(field, comp, _onGetFilterCtrl);
}else{
System.err.println(result.msg);
throw new Error(result.msg);
}
},$bind(this);

var _onGetFilterCtrl = function(msg){
var result = msg.result;
if(result.err == 0){
filter = result.obj.ctrl;
if(!filter) return;
// TODO enable filter ?
_renderValues.call(this, filter);
}else{
System.err.println(result.msg);
throw new Error(result.msg);
}
},$bind(this);

var _canSort = function(){
var b = !!comp && !!field;
return b ? comp.isSortable() : false;
};

var _canFilter = function(){
var b = !!comp && !!field && !!filter
return b ? comp.isFiltered() : false;
};

var _renderValues = function(filter, values){
var td = document.getElementById("fc-values"),
i, len, val, item, buf, showAll;

showAll = !values;
values = values || filter.getAllValues();
td.innerHTML = "";
var isAll = filter.isAllSelected();
for(i=showAll ? -1:0, len=values.length; i<len; i++){
val = i < 0 ? {disp:"All", real:"__ALL__", selected:isAll, available:true} :
values[i];
item = document.createElement("LI");
item.className = val.available ? "item--available" : "item--unavailable";
buf = ["<input type='checkbox' id='fc-index-",i,"' ",
buf.push("value='",", val.real, "'",
if(val.selected){
buf.push("checked");
}
buf.push("/>", val.disp);
item.innerHTML = buf.join("");
td.appendChild(item);
}
};

this.doSort = function(type){
if(!_canSort()) return;
comp.sort(field, type);
};

this.selectValue = function(e){
e = e || window.event;
var ele = e.srcElement||e.target, key, checked;
if(ele.tagName != "INPUT") return;
key = ele.value;
checked = ele.checked;
if("__ALL__" === key){
filter.selectAll(checked);
_renderValues.call(this, filter);
}else{
filter.select(key, checked);

// The element for 'All' should be updated checked status
ele = document.getElementById("fc-index--1");
if(filter.isAllSelected()){
ele.checked = true;
}else{
ele.checked = false;
}
};

this.doFilter = function(){
if(!_canFilter()) return;
this.hide();
};

this.clear = function(){
if(!filter) return;
filter.clearSelects();
_renderValues.call(this, filter);
};

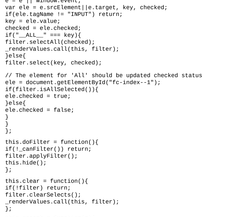
this.search = function(e){
if(!filter) return;
e = e || window.event;
var ele = e.srcElement||e.target, value = ele.value, values;
if(value){
values = filter.search(value);
}
_renderValues.call(this, filter, values);
};

this.clearSearch = function(){
document.getElementById("searchbox").value="";
if(!filter) return;
_renderValues.call(this, filter);
};

Report.getDataContainerBy(context.dsid, _onGetDataContainer);

</script>
```

The dialog looks like as follows:



Related topics:

- Publishing additional resources for how to publish customized control files to JReport Server.
- Downloading customized controls from JReport Server

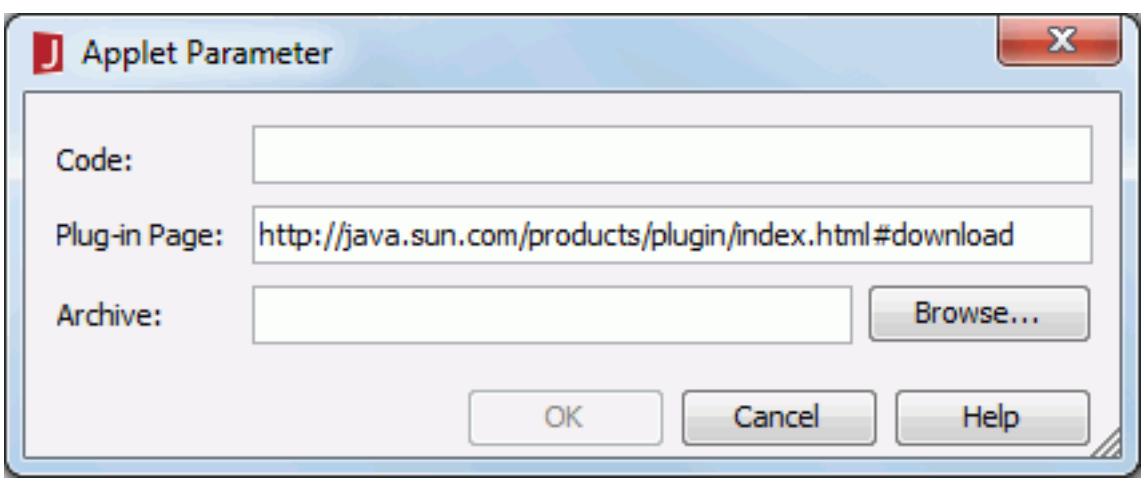
Multimedia objects

Multimedia objects, including Applet, Flash, RealMedia, and Windows Media objects can be inserted in the report areas listed in Component placement.

The following explains how to insert different multimedia object into a report:

Inserting an applet object

1. Click **Insert** > **Multimedia Objects** > **Applet**, or drag **Applet** from the Toolbox panel to the report. The Applet Parameter dialog appears.

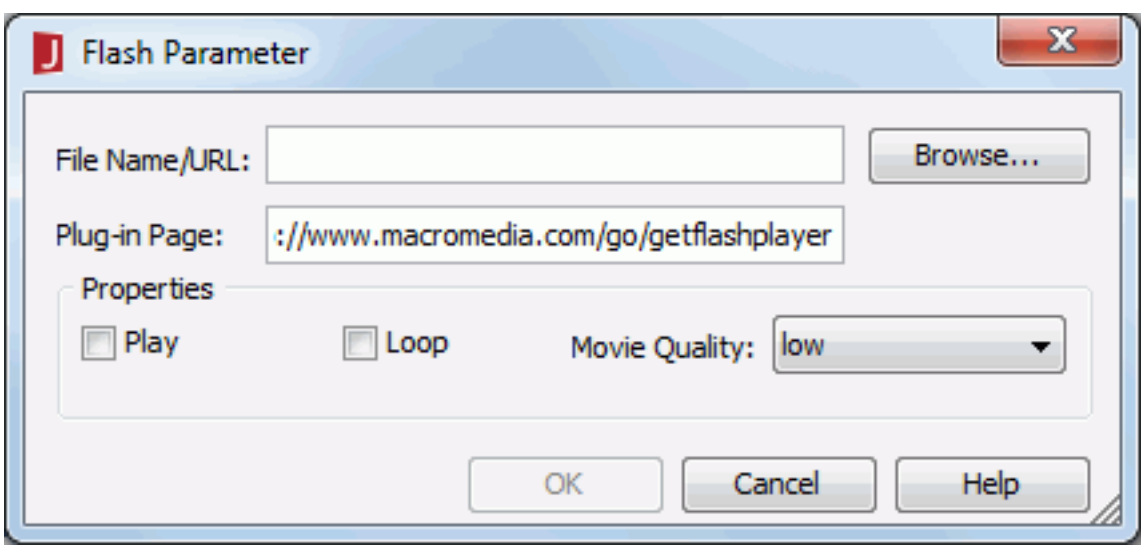


2. Type the code for this applet in the Code field.
3. In the Plug-in Page field, set the URL of the plug-in page. By default, it is `http://java.sun.com/products/plugin/index.html#download`.
4. Click the **Browse** button next to the Archive field to specify the archive file (.zip or .jar).
5. Click **OK** in the dialog to insert the object in the required location.

When you use a Java technology-enabled web browser to view a report that contains an applet, the applet's code will be transferred to your system and executed by the web browser's Java Virtual Machine (JVM).

Inserting a flash object

1. Click **Insert** > **Multimedia Objects** > **Flash**, or drag **Flash** from the Toolbox panel to the report. The Flash Parameter dialog appears.

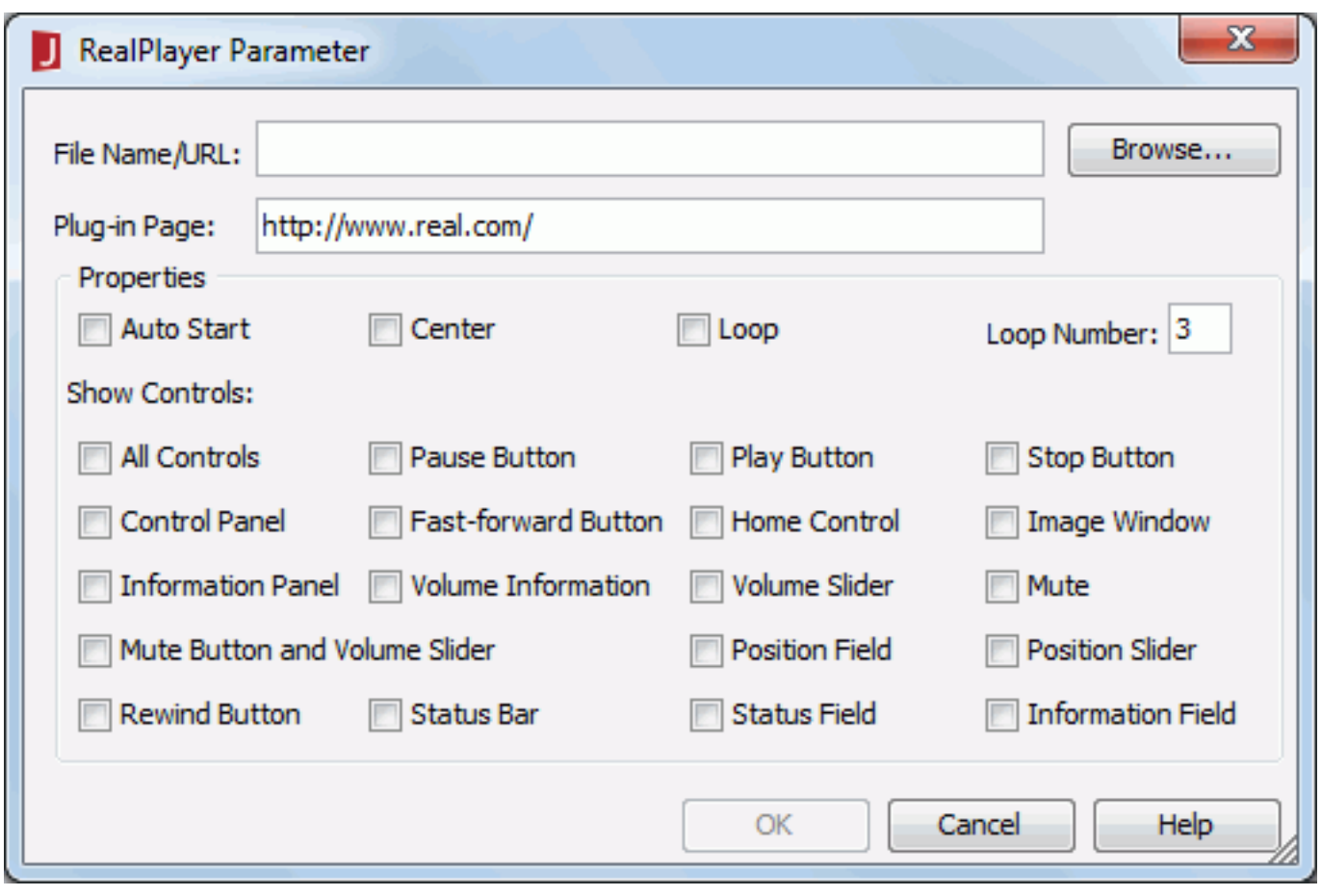


2. Click the **Browse** button to select a flash file, or type the URL of a flash file in the File Name/URL field.
3. In the Plug-in Page field, set the URL of the plug-in page. By default, it is `http://www.macromedia.com/go/getflashplayer`.
4. Specify the properties so as to control the options when playing the flash file in Page Report Studio or Web Report Studio.
5. Click **OK** in the dialog to insert the object in the required location.

When you view the report result with a web browser in which the Flash Player plug-in has been installed, the media file will be played according to your settings.

Inserting a RealMedia object

1. Click **Insert** > **Multimedia Objects** > **RealMedia File**, or drag **RealMedia File** from the Toolbox panel to the report. The RealPlayer Parameter dialog appears.

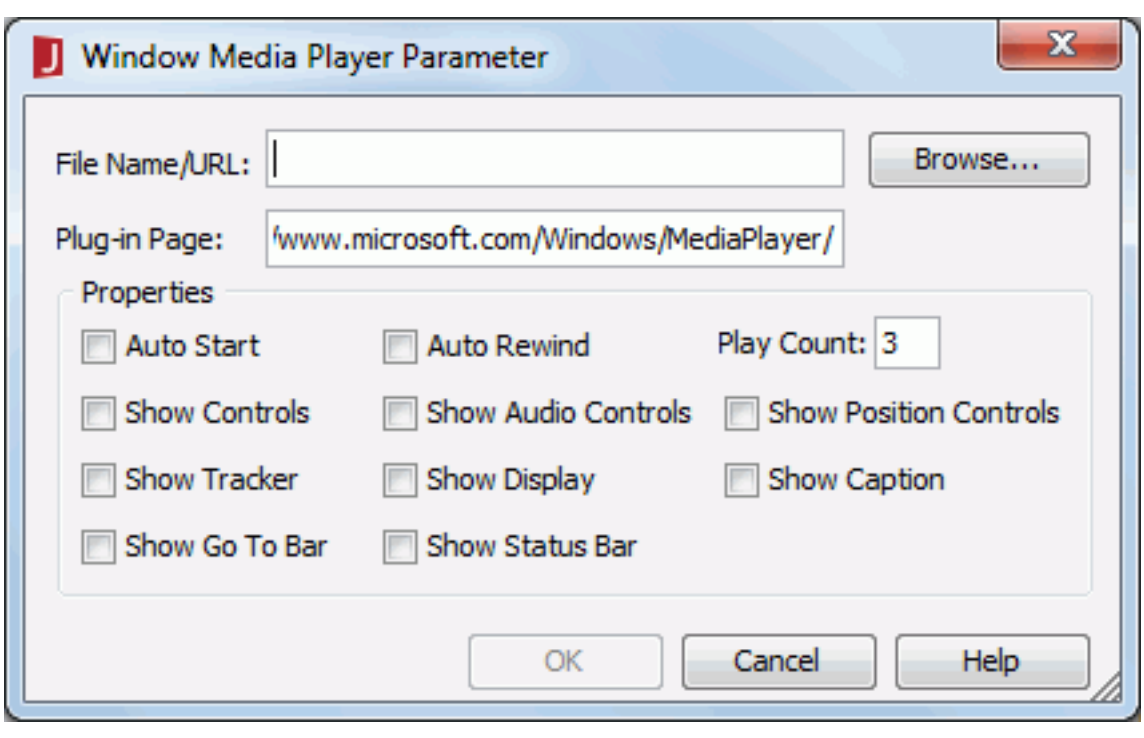


2. Click the **Browse** button to select a RealMedia file, or type the URL of a RealMedia file in the File Name/URL field.
3. In the Plug-in Page field, set the URL of the plug-in page. By default, it is `http://www.real.com/`.
4. Specify the properties so as to control the options when playing the media file in Page Report Studio or Web Report Studio.
5. Click **OK** in the dialog to insert the object in the required location.

When you view the report result with a web browser in which the RealPlayer plug-in has been installed, the media file will be played according to your settings.

Inserting a Windows Media object

1. Click **Insert** > **Multimedia Objects** > **Windows Media File**, or drag **Windows Media File** from the Toolbox panel to the report. The Windows Media Player Parameter dialog appears.



2. Click the **Browse** button to select a Windows Media file, or type the URL of a Windows Media file in the File Name/URL field.
3. In the Plug-in Page field, set the URL of the plug-in page. By default, it is `http://www.microsoft.com/Windows/MediaPlayer/`.
4. Specify the properties so as to control the options when playing the media file in Page Report Studio or Web Report Studio.
5. Click **OK** in the dialog to insert the object in the required location.

When you view the report result with a web browser in which the Windows Media Player plug-in has been installed, the media file will be played according to your settings.

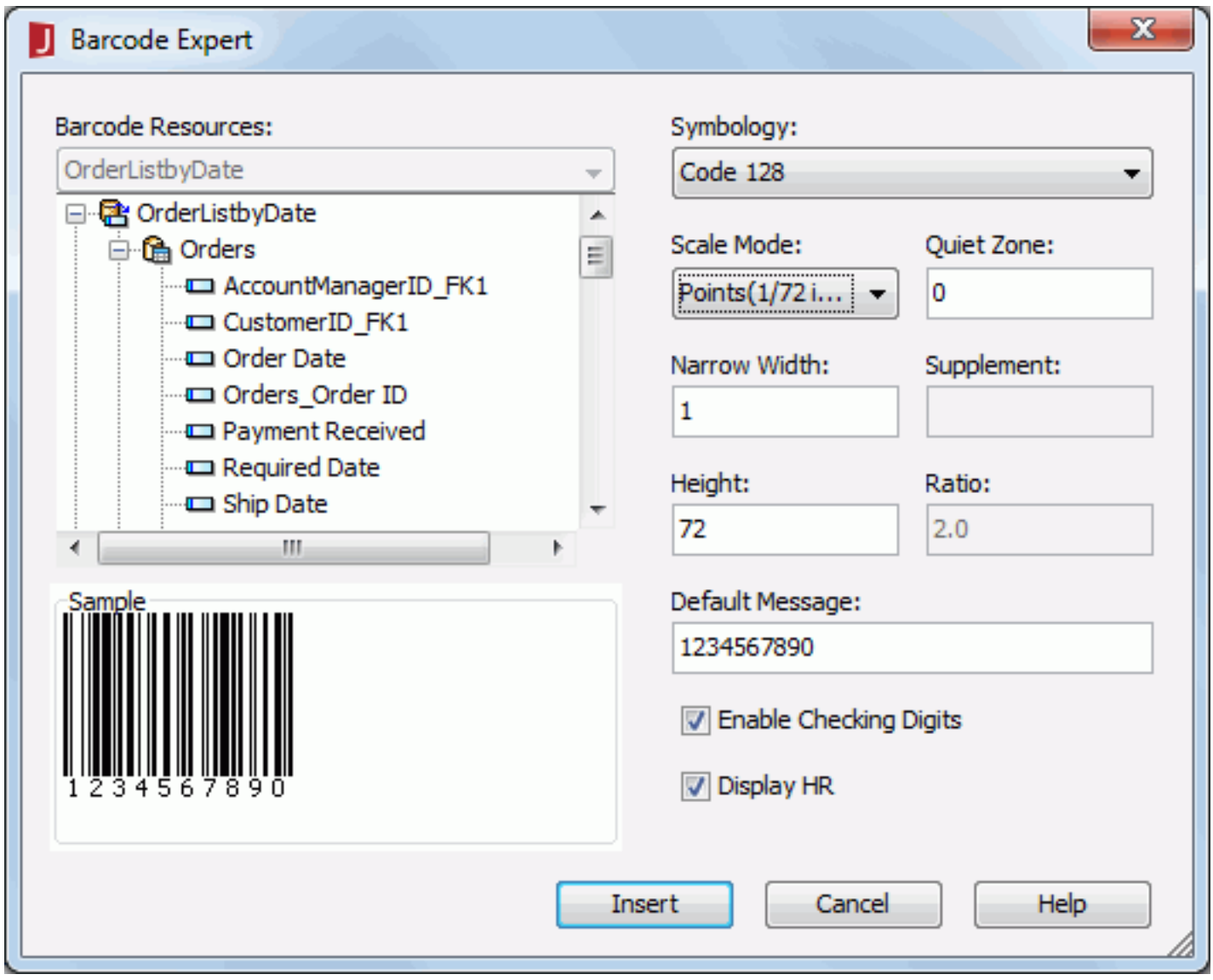
When a multimedia object has been inserted into a report, you can further edit it if required (to do this, right-click the object and select **Edit WebOLEObject** from the shortcut menu). You can also customize its appearance and behavior by setting properties in the Report Inspector, which can be saved as a CSS style for future use (for details, see Creating a CSS style).

See an example: The SampleComponents catalog, included with JReport Designer, contains reports that have examples of how each component type could be used in a report. For the label component example, open `<install_root>\Demo\Reports\SampleComponents\MultimediaObject.cls`.

Barcodes

A barcode can be used as a component and directly inserted into a report. To do this:

1. Click **Insert** > **Barcode** on the menu bar, or click the **Insert Barcode** button on the Insert toolbar. The Barcode Expert dialog appears.



2. Select a dataset from the Barcode Resources drop-down list, and then specify the source of the barcode: a DBField, formula, summary, or parameter.
3. Select the required barcode type from the Symbology drop-down list. You can preview the format of the selected symbology in the Sample box (for details about barcode symbologies, see Appendix 2: Barcode symbologies).
4. Select the unit for the values of the quiet zone, narrow width, supplement, height and ratio from the Scale Mode drop-down list.
5. Specify the space around the barcode in the Quite Zone field (most symbologies require quiet zones that precede and follow the barcode).
6. Set the barcode bar width in the Narrow Width field.
7. Specify the supplement of the barcode if necessary. If you select the barcode type as Code 39, Code 128/128A/128B/128C or Codabar, you will not need to set the supplement.
8. Set the height of the barcode bars in the Height field.
9. Specify the width ratio of the thick bar to the thin bar in the Ratio field if you selected the barcode type as Code 39 or Codabar. The ratio can only be 2.0 or 3.0. Any ratio value not equal to 2.0 or 3.0 will be regarded as 2.0.
10. Change the default value of the barcode in the Default Message field if necessary.
11. Select the **Enable Checking Digits** checkbox if you want to include check digits in the barcode.
12. If you have selected the barcode type as Code 39, Code 128/128A/128B/128C, or Codabar, then you can check **Display HR** to display the barcode numbers together with the barcode.
13. Upon finishing, click **Insert** to close the dialog, then click the mouse button in the location where you want the barcode to be displayed.

Notes:

- Where a barcode can be inserted in a report depends on what you have selected as its source.
- You can also create a barcode component by first inserting a DBField, formula, summary or parameter and then changing its display type to barcode.

See an example: The SampleComponents catalog, included with JReport Designer, contains reports that have examples of how each component type could be used in a report. For the barcode component example, open `<install_root>\Demo\Reports\SampleComponents\Barcode.cls`.

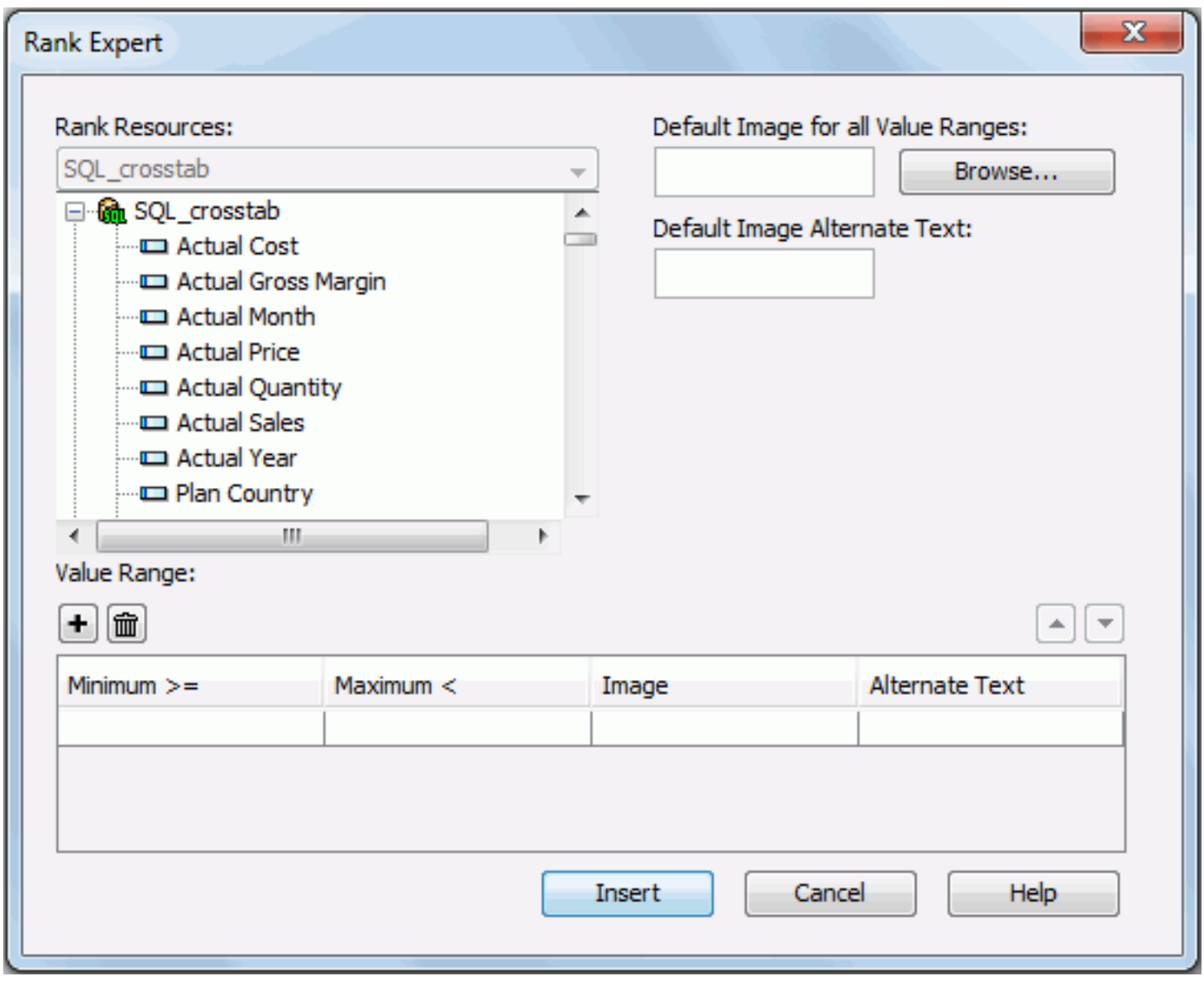
Ranks

A rank is a component that is based on the value of another component - a DBField, formula, summary, or parameter. One or more values for these fields can be mapped to an image known as the rank. The rank component allows you to convert raw data values to easy-to-understand graphical cues or to hide sensitive details but still convey the general data value to the report end users.

To insert a rank into a report:

1. Do one of the following:
   - Click **Insert** > **Rank** on the menu bar.
   - Click the **Insert Rank** button on the Insert toolbar.
   - Drag **Rank** from the Toolbox panel to the report.

   The Rank Expert dialog appears.



2. The Rank Resources drop-down list displays the dataset available for the current insertion, and the list box below shows its fields, including DBFields, formulas, summaries and parameters.

   Select the field you want to insert as rank from the list box. You can also create formulas, summaries, and parameters here.

3. Click the **Browse** button to select an image file to be the default image (used if a value does not fit any of the specified value ranges).

4. Type the alternate text of the default image in the Default Image Alternate Text box, which will be displayed when the image cannot be loaded or is unavailable.

5. In the Value Range box, specify the minimum value, maximum value, image and alternate text respectively in the Minimum, Maximum, Image, and Alternate Text columns. Click or to add or delete a range. To adjust the order of the ranges, click or . A value, which is larger than or equal to the minimum value of a range and less than the maximum value, will be displayed as the image, or the alternate text if the image cannot be loaded or is unavailable. For a value belonging to two or more overlapping ranges, the highest range in the Value Range box will take effect.

6. Click **Insert** in the dialog, then click the mouse button in the location where you want the rank to be displayed.

Notes:

- Where a rank can be inserted in a report depends on what you have selected as its source.
- You can also create a rank by first inserting a DBField, formula, summary or parameter, and then changing its display type as rank.

Example

Suppose that you want to define product quality rank based on customer grading, and you want to define three levels:

50-70, Poor

71-90, Good

91-100, Excellent

Specify the following in the Value Range box:

1. In the first range line, set the minimum and maximum values as **50** and **70** respectively in the cells of the Minimum and Maximum columns. Go to the Image cell, and then click **<Browse...>** to select an image to represent the rank. Click the **Alternate Text** cell and enter **Poor**.

2. Click to add another line. Set the minimum and maximum values as **71** and **90**, select another image to represent the rank and enter **Good** in the Alternate Text cell.

3. Add one more line, and set the values, image and name for Excellent in the same way as before.

You have successfully defined a rank for product quality.

See an example: The SampleComponents catalog, included with JReport Designer, contains reports that have examples of how each component type could be used in a report. For the rank component example, open `<install_root>\Demo\Reports\SampleComponents\Rank.cls`.

Drawing objects

You can add arcs, ovals, boxes, round boxes and lines to your report.

Drawing objects can only be inserted into banded objects. To insert a drawing object into a report, you can use either of the following two methods:

- Click **Insert** > **Drawing Objects** on the menu bar, then click the drawing object (Line, Box, Arc, Oval, or Round Box) you want to insert on the submenu.
- Select an item from the drawing object drop-down list on the Insert Toolbar.

Then, move your mouse pointer to the insertion point, drag the mouse button till you are satisfied with the drawing object's size and proportions. The selected drawing object will then be inserted into your report at the specified position. Note that only boxes, horizontal or vertical lines can be presented in Page Report Studio.

When a drawing object has been inserted into a report, you can customize its appearance and behavior by setting properties in the Report Inspector.

See an example: The SampleComponents catalog, included with JReport Designer, contains reports that have examples of how each component type could be used in a report. For the drawing object example, open `<install_root>\Demo\Reports\SampleComponents\ForDrawingObject.cls`.

UDOs

A user defined object (UDO) is an object inherited from JReport system's object. You can add and modify new properties, as well as other existing methods and specify new methods for modifying your UDO. UDOs, which have been appended to the ADDCLASSPATH variable of setenv.bat in `<install_root>\bin`, can be inserted as components into a report and consequently become a part of the report.

Go through the following topics for details about UDOs:

- Objects and interfaces for creating UDOs
- Creating a UDO manually
- Inserting a UDO into a report
- Built-in UDOs: JHyperLink and JRotator

Objects and interfaces for creating UDOs

The following objects and interfaces are necessary for creating a UDO. Refer to the corresponding class or interface in the JReport Javadoc located at `<install_root>\help\api`.

- **Jet.report.JRObjectTemplate**
  This is an object provided by the JReport system, and is used to derive a UDO. The JRObjectTemplate contains several predefined properties of the standard JReport Object used to compose a report, including: X, Y, Height and Width. You can add more properties (supported by JReport system) to your own UDO. These properties can be viewed and modified by the Report Inspector at design time.

  > **Note:** Only JReport system properties or properties inherited from them can be added, such as JetNumber, JetColor, JetString, and JetEnumeration. See the jet.controls package.

- **jet.datastream.JRVisiableResult**
  This is another object provided by the JReport system. You will need to define an object inherited from this object, so that you can specify the methods for saving and restoring the UDO object. JRVisiableResult provides methods for saving and restoring UDOs.

- **jet.datastream.JRObjectResult**
  If the UDO you define is not used for displaying, that is, it will not appear inside the report result on the screen or on paper, then it can be inherited from the JRObjectResult object.

- **jet.udo.JRObjectResultCreator**
  This is an interface. It takes JReport system's JReport Record as input parameters and produces the UDO's JRObjectResult. A definition of the JReport Record is in the jet.connect package.

- **jet.udo.JRObjectRender**
  This is another interface. It provides a method used for painting the UDO to the report. The report can then be shown on the screen or printed on a printer.

- **jet.udo.JRObjectEditor**
  This is an optional interface. If specified, the JReport system will use it to handle interactive events (key events, mouse events, paint, and so on) at design time. If this interface is not implemented, the JReport system will use the default one.

- **jet.udo.JRGroupListener**
  This interface is used when you define a UDO with a value which is calculated based on a group of data.

Creating a UDO manually

To create a UDO manually, follow the steps:

1. Inherit from the class JROblectTemplate to create a template file.
2. Inherit from the class JRVisiableResult or JRObjectResult to create a result file.
3. Implement JRObjectResultCreator to create a result creator file.
4. Implement interface JRObjectRender to create a result render file. If you want to display a UDO in the Design area, implement interface JRObjectEditor. If your UDO is a group level object, implement interface JRGroupListener.
5. Modify the file udo.ini in `<install_root>\lib` by adding the following:

```
Begin
name=MyDbField
template=myudo.MyDbFld
resultobject=myudo.MyDbFldRst
resultcreator=myudo.MyDbFldCreator
resultviewer=myudo.MyDbFldRender
End
```

6. Compile the Java files, add the classes to the class path of JReport Designer and run it.

   Note: If you want to use class files for the UDO which have been used in previous versions, you should re-compile the source files.

Following are two specific examples about creating UDOs:

- Example 1: Accessing a record for a UDO and registering the UDO with the report system
- Example 2: Making a group based on a UDO

Example 1: Accessing a record for a UDO and registering the UDO with the report system

This example describes how to access a record for your UDO and how to register your UDO with the report system.

1. A UDO needs to implement at least four classes from JReport. The four classes are for template, result, creator and render files respectively.

   - **MyDbFld.java**

```
//Template
package myudo;

import jet.report.*;
import jet.controls.*;

/**
 * This class extends form JRObjectTemplate, add two properties "ColumnName" and
 * "TextColor".
 */
public class MyDbFld extends JRObjectTemplate {
        /**
         * This property is for the column name of MyDbFld.
         */
        public JetColumnName columnName = new JetColumnName(this, "ColumnName");

        /**
         * This property is for the background color of MyDbFld..
         */
        public JetColor backColor = new JetColor(this, "TextColor", null, true);

        public MyDbFld() {
                super();
                columnName.setEditable(true);
                // set the default size.
                set("Width", 40); // before build575 should be width.set(40);
                set("Height", 20); // before build575 should be height.set(20);
                // add the property to group then they can displayed in ReportInspector.
                addPropertyToGroup("ColumnName", "Others");
                addPropertyToGroup("TextColor", "Color");
        }

        /**
         * Return the instance name prefix. >
         */
        public String getPrefix() {
                return "MyDbField";
        }
}
```

   - **MyDbFldRst.java**

```
//Result
package myudo;

import jet.connect.Record;
import jet.connect.DbValue;
import jet.util.*;
import jet.datastream.*;

public class MyDbFldRst extends JRVisiableResult {
        public MyDbFldRst() {
        }

        // retrieve value from column.
        DbValue getValue() {
                // get the column name.
                String colName = (String) getPropertyByName("ColumnName").getObject();
                // get the report record.
                Record record = getRecord();
                // return the column value.
                return record.getCell(colName);
        }

        // make the text will be displayed.
        String getText() {
                String sRet;
                DbValue value = getValue();
                if (value != null && !value.isNull()) {
                        sRet = value.toString();
                } else {
                        sRet = "NULL";
                }
                return sRet;
        }
}
```

   - **MyDbFldCreator.java**

```
//Creator
package myudo;

import java.awt.*;
import java.util.*;
import jet.report.*;
import jet.controls.*;
import jet.connect.*;
import jet.udo.*;
import jet.datastream.*;
import guitools.Painter;
import jet.util.*;

/**
 * Implements JRObjectResultCreator for creating JRObjectResult in JReport
 * Engine Bean.
 */
public class MyDbFldCreator implements JRObjectResultCreator {
        public MyDbFldCreator() {
        }

        public JRObjectResult createJRObjectResult(JRObjectTemplate rptobj,
                        Record record) {
                MyDbFld rpt = (MyDbFld) rptobj;
                MyDbFldRst dsField = new MyDbFldRst();
                dsField.setTemplate(rpt);
                return dsField;
        }
}
```

   - **MyDbFldRender.java**

```
//Render
package myudo;

import java.awt.*;
import java.awt.event.*;
import java.util.*;
import jet.datastream.*;
import jet.connect.DbValue;
import jet.connect.Record;
import jet.udo.*;

public class MyDbFldRender extends Component implements JRObjectRender {
        String text = null;

        Color color;

        Color background;

        /**
         * The default constructor.
         */
        public MyDbFldRender() {
        }

        // retrieve property and data from JRObjectResult.
        public void setProperty(jet.util.PropertySetable dsPropSet) {
                text = ((MyDbFldRst) dsPropSet).getText();
                color = (Color) dsPropSet.getPropertyByName("TextColor").getObject();
                background = (Color) dsPropSet.getPropertyByName("Background")
                                .getObject();
                setBounds(((MyDbFldRst) dsPropSet).getBounds());
        }

        /**
         * Paint the text.
         */
        public void paint(Graphics g) {
                Dimension dim = getSize();
                if (background != null) {
                        g.setColor(background);
                        g.fillRect(0, 0, dim.width, dim.height);
                }
                if (text != null) {
                        if (color == null)
                                color = Color.black;
                        g.setColor(color);
                        g.drawString(text, 10, 10);
                }
        }
}
```

   Note: Since the compiling process of JReport Designer differs with that of JReport Server, when you run UDOs in these two applications, you should pay attention to some minor differences. For example, to get width and height, in JReport Designer, you should use:

```
w = guitools.toolkit.Unit.convertUnitToPixel(((Integer)propertySetable.
getPropertyByName ("Width").getObject()).intValue());
h = guitools.toolkit.Unit.convertUnitToPixel(((Integer)propertySetable.
getPropertyByName ("Height").getObject()).intValue());
```

   While in JReport Server, you should use the following instead:

```
JRObjectResult obj = (JRObjectResult)propertySetable;
w = guitools.toolkit.Unit.convertUnitToPixel(obj.getTemplate().getWidth(obj));
h = guitools.toolkit.Unit.convertUnitToPixel(obj.getTemplate().getHeight(obj));
```

2. Compile the Java files.

   To compile these four Java files, you should add report.jar and JREngine.jar with their path into the class path. For example, use the following command:

```
Javac -classpath "C:\JReport\Designer\lib\JRengine.jar;C:\JReport\Designer\lib\report.jar;C:
\test "MyDbFld.java
```

   Here it is assumed that JReport Designer is installed to `C:\JReport\Designer`. The Java files for the example are in `C:\test\myudo`.

3. Modify the udo.ini file in the `<install_root>\lib` directory by appending the four classes as follows:

```
Begin
name=MyDbField
template=myudo.MyDbFld
resultobject=myudo.MyDbFldRst
resultcreator=myudo.MyDbFldCreator
resultviewer=myudo.MyDbFldRender
End
```

4. Edit setenv.bat in `<install_root>\bin` by appending the path of the four classes to the batch file's ADDCLASSPATH variable. Assume that the four classes are located in `D:\test\myudo`.

   ```
   set ADDCLASSPATH=%JAVAHOME%\lib\tools.jar;D:\test;
   ```

5. Start JReport Designer.

6. Click **Insert** > **UDO**. You will now find a new UDO object named MyDbFld in the drop-down list of the Insert UDO dialog.

Example 2: Making a group based on a UDO

This example describes how to make a group based on a UDO. This sample is similar to a summary field. This example includes the implementation of four classes and shares a class with Example 1.

1. Create the template, result, creator and render files as follows:
   - **MySumFld.java**

```
//Template
package myudo;

import jet.report.*;
import jet.controls.*;

/**
 * This class extends from JRObjectTemplate, add two properties "ColumnName" and
 * "TextColor".
 */
public class MySumFld extends MyDbFld {
        public MySumFld() {
                super();
        }

        /**
         * Return the instance name prefix.
         */
        public String getPrefix() {
                return "MySumField";
        }

        public boolean isGroupListener() {
                return true;
        }
}
```

   - **MySumFldRst.java**

```
//Result
package myudo;

import jet.JRStopEngineException;
import jet.connect.Record;
import jet.connect.DbValue;
import jet.util.*;
import jet.datastream.*;
import java.io.*;

public class MySumFldRst extends MyDbFldRst {
        double value;

        public MySumFldRst() {
        }

        // make the text will be displayed.
        String getText() {
                return "" + value;
        }

        /**
         * Read the data from DataInput.
         * UDO can override this method to restore its own data.
         * @throws JRStopEngineException
         * */
        protected void readProperties(DataInput in, DSDataStreamable ds)
                        throws IOException, JRStopEngineException {
                super.readProperties(in, ds);
                value = in.readDouble();
        }

        /**
         * Write the data to DataInput.
         * UDO can override this method to save its own data.
         * */
        protected void writeProperties(DataOutput out) throws IOException {
                super.writeProperties(out);
                out.writeDouble(value);
        }
}
```

   - **MySumFldCreator.java**

```
//Creator
package myudo;

import java.awt.*;
import java.util.*;
import jet.report.*;
import jet.controls.*;
import jet.connect.*;
import jet.udo.*;
import jet.datastream.*;
import guitools.Painter;
import jet.util.*;

/**
 * * Implements JRObjectResultCreator for creating JRObjectResult in JReport *
 * Engine Bean.
 */
public class MySumFldCreator implements JRObjectResultCreator, JRGroupListener {
        // This interface must implement by JRObjectCreator.
        MySumFld rpt;

        MySumFldRst dsField;

        double value;

        public MySumFldCreator() {
        }

        public void setTemplate(JRObjectTemplate rptobj) {
                rpt = (MySumFld) rptobj;
        }

        public JRObjectResult createJRObjectResult(JRObjectTemplate rptobj,
                        Record record) {
                rpt = (MySumFld) rptobj;
                MySumFldRst dsField = new MySumFldRst();
                dsField.setTemplate(rpt);
                dsField.value = value;
                return dsField;
        }

        /** * This method will be called before the first record. */
        public void prepareFetchRecords() {
                value = 0.0;
        }

        /** * This method will be called after the last record. */
        public void finishFetchRecords() {
        }

        /** * This method will be called each record. */
        public void fetchNewRecord(Record record) {
                String colName = (String) rpt.getPropertyByName("ColumnName")
                                .getObject();
                // return the column value.
                DbNumber num = (DbNumber) record.getCell(colName);
                if (!num.isNull()) {
                        value += num.intValue();
                }
        }
}
```

   - **MyDbFldRender.java** (See Example 1 for details.)

2. Compile the four Java files in the same way as Example 1.
3. Modify udo.ini in `<install_root>\lib\` by appending the four classes as follows:

```
Begin
name=MySumFld
template=myudo.MySumFld
resultobject=myudo.MySumFldRst
resultcreator=myudo.MySumFldCreator
resultviewer=myudo.MyDbFldRender
End
```

4. Edit setenv.bat in `<install_root>\bin` by appending the path of the four classes to the batch file's ADDCLASSPATH variable. Assume that the four classes are located in `D:\test\myudo`.

   `set ADDCLASSPATH=%JAVAHOME%\lib\tools.jar;`**`D:\test;`**

5. Start JReport Designer.
6. Click **Insert** > **UDO**. You will now find a new UDO object named MySumFld in the drop-down list of the Insert UDO dialog.

Inserting a UDO into a report

A UDO can be inserted in the page report areas listed in Component placement. UDOs are not supported for web reports and library components.

To insert a UDO to a report:

1. Start JReport Designer with the modified setenv batch file that includes the new classes, then create a new report, or open an existing report.
2. Do one of the following:
   - Click **Insert** > **UDO** on the menu bar.
   - Click the **Insert UDO** button on the Insert toolbar.
   - Drag **UDO** from the Toolbox panel to the destination where you want to insert the UDO.
3. In the Insert UDO dialog, select a UDO as required from the UDO drop-down. The list contains all UDOs that have been registered in the file setenv.bat in `<install_root>\bin`.
4. Click **OK** in the dialog.
   - If you use menu or toolbar command to insert the UDO, click the mouse button in the desired location to insert the UDO there.
   - If you take the dragging and dropping method to insert the UDO, the UDO will be inserted there upon clicking the OK button.

After a UDO has been inserted, you can then customize its appearance and behavior by setting properties in the Report Inspector, which can be saved as a CSS style for future use (for details, see Creating a CSS style). Once you publish your report to the server, be sure to include the classpath in the setenv batch file in JReport Server (<server_root>\bin\setenv.bat).

Built-in UDOs: JHyperLink and JRotator

JReport provides two built-in UDOs: JHyperLink and JRotator. JHyperLink is used to create hyperlinks in reports, and JRotator can be used to rotate text or images.

This section shows the usage of the two UDOs in detail:

- Creating links with JHyperLink
- Rotating text/image with JRotator

See also Properties - UDO for details about the properties of JHyperLink and JRotator in the Report Inspector.

Creating links with JHyperLink

JHyperLink can be used to create hyperlinks in page reports. This section shows you two specific examples about using JHyperLink:

- Example 1: Building hyperlinks by using JHyperLink
- Example 2: Linking reports using parameters by using JHyperLink

Example 1: Building hyperlinks by using JHyperLink

Sometimes you have a number of reports that contain different views of the same data. You may want to build a summary report to organize these reports together. For example, Report A is Summary Information for Orders, while Report B is an Invoice Report showing the detail line items of the order. When you click on the Order ID in Report A, it will go to the detail page of that order in Report B. With this hyperlink feature, you can build hyperlinks among reports. In addition, you can also insert any hyperlink to be invoked. In any case, a web browser will be displayed to view the specified pages.

1. Start JReport Designer and open the report in which you want to build a hyperlink.
2. On the menu bar, click **Insert** > **UDO**.
3. In the Insert UDO dialog, select **JHyperLink** from the drop-down list, and then click **OK**. You will then be able to insert a hyperlink into the report.
4. Keep the JHyperLink focused, and then go to the Report Inspector.
5. Input **http://localhost:8888** in the value cell of the URL property as the JReport Server address, make sure the Executer Class Name value is jet.udos.IEExecuter, and then type **JReport Server** in the Display Value property value cell as the display name of the hyperlink.
6. Insert another hyperlink using the same way and define its properties in the Report Inspector as follows:
   - **URL**: http://www.jinfonet.com
   - **Executer Class Name**: jet.udos.IEExecuter (the default value)
   - **Display Value**: Website

   Two hyperlinks have been successfully built into the report.
7. In view mode, click the **JReport Server** hyperlink that you just made. It will bring out a browser session to connect with JReport Server. Make sure that you have launched JReport Server and that the URL corresponds with your server configuration.
8. Click the **Website** hyperlink. You will then see the http://www.jinfonet.com home page in a web browser.

Notes: After you insert a JHyperLink into a report which is to be exported to PDF, Excel or HTML, you can specify whether to make the hyperlink effective by setting a property which corresponds to the format type. Respectively, they are: Enable Hyperlink in PDF for PDF, Enable Hyperlink in Excel for Excel and Enable Hyperlink in HTML for HTML.

Example 2: Linking reports using parameters by using JHyperLink

JHyperLink can be used to link reports similar to master and detail reports to show related information, such as Order detail to higher level Year to Date Sales. You can use the JHyperLink object to control the URL to show the information in the linked reports on JReport Server. It is similar to master/detail but does not have the ability to navigate through the detail and return to the master. The key to using JHyperlink is learning how to use URLs to call page reports.

For how to use JHyperLink to link reports using a URL, JReport Designer provides an example for your better understanding.

All the needed materials for this example, including the reports and catalog are in `<install_root>\help\samples\UDODemo`.

1. In JReport Designer, open the catalog for this example Demo.cat in `<install_root>\help\samples\UDODemo\reports`.
2. Select one of the existing reports and ensure the reports in this catalog can run. If there is a problem, open the Catalog Browser, highlight the connection node, and in the Properties sheet, modify the value of the URL property to connect to the demo database, which is jdbc:hsqldb:`<install_root>\help\samples\UDODemo\db\demo`.

Linking the master report to the detail report

Formulas will be used to control the property values URL and Display Value of JHyperLink to make it link the master and detail reports.

1. Open the report that contains the master report - mainreport.cls.

   A JHyperLink is inserted in the Detail panel of the report. The URL and Display Value property values of the UDO are as follows:
   - **URL**: Controlled by the formula linkToSalesYearly, which is defined as follows:

     ```
     "http://localhost:8888/jrserver?jrs.cmd=jrs.web_vw&jrs.catalog=/USERFOLDERPATH/admin/
     Demo.cat&jrs.report=/USERFOLDERPATH/admin/SalesPerformancebyYear.cls&jrs.
     result_type=8&jrs.param$pEmployeeID="+@"Employee_Employee ID"
     ```

     This URL value enables you to drill down to the detail report and also make a link for this detail report on the top of the page for you to drill up and down.
   - **Display Value**: Determined by the DBField EmployeeName to show the Employee Name in the master report.
2. Open the report that contains the detail report SalesPerformancebyYear.cls.

   In the GroupHeader panel, the value of Total is a JHyperLink, whose property values URL and Display Value are both controlled by formulas.
   - **URL**: Controlled by the formula linkToQuarter, which is defined as follows:

     ```
     "http://localhost:8888/jrserver?jrs.cmd=jrs.web_vw&jrs.catalog=/USERFOLDERPATH/admin/
     Demo.cat&jrs.report=/USERFOLDERPATH/admin/SalesYearPerformancebyQuarter.cls&jrs.
     result_type=8&jrs.param$pEmployeeID="+@"Employee_Employee ID"+"&jrs.param
     $pYear="+@getYear;
     ```

     Where, the URL is used to access the detail report SalesYearPerformancebyQuarter.cls, and the records of the detail report are filtered by the parameters pEmployeeID and pYear.
   - **Display Value**: Controlled by the formula getSalesByYear, which is defined as follows:

     ```
     "$"+ToText(@Sum_TotalSales0)
     ```

     This formula returns the total sales amount for the year.
   - **Target**: The options are self, parent, top and blank. Use self to replace the current report with the linked report. Use blank if you want to open a new browser tab so you can return to the master report window.

Running the reports on JReport Server

This example requires you to access the report by URL, you can drill down through the reports. To do this:

1. Start JReport Server.
2. Publish the reports and catalog to JReport Server as admin into the root level of My Reports.
3. Run the master report mainreport.cls in Page Report Studio.

Orders Detail Report

Order ID	Customer Name	Employee Name	Product Name
3031	American Coffee Inc.	Katherine Chen	Mocha Java
3031	American Coffee Inc.	Katherine Chen	Colombian
3031	American Coffee Inc.	Katherine Chen	Expresso Roast
3010	American Coffee Inc.	Joseph Compton	House Blend
3031	American Coffee Inc.	Katherine Chen	Java Hut Select
3010	American Coffee Inc.	Joseph Compton	Chocolate Hazelnut
3010	American Coffee Inc.	Joseph Compton	Hazelnut Crosse
3005	Bob's Coffee	Maxine Olympus	Colombian Supremo
3005	Bob's Coffee	Maxine Olympus	Mocha Java
3005	Bob's Coffee	Maxine Olympus	Guatemala
3005	Bob's Coffee	Maxine Olympus	Brazilian Santos
3009	Brazilian's Best	Maxine Olympus	Sumatra Mandheling

4. Click any name in the Employee Name column, and the year's performance of the employee will be displayed.

Sales Peformance by Year

Order Year: 1998 **Total:** $4,777.9

Order ID	Customer Name	Product Name	Product Category	Total Sales
3031	American Coffee Inc.	Mocha Java	Regular	318.00
3031	American Coffee Inc.	Colombian	Decaf	183.50
3031	American Coffee Inc.	Expresso Roast	Decaf	86.40
3031	American Coffee Inc.	Java Hut Select	Regular	170.00
3007	Café International	Vienna Roast	Regular	285.00
3007	Café International	Almond Tosted	Regular	420.00
3007	Café International	Butternut Rum	Regular	450.00
3007	Café International	Vanilla Hazelnut	Regular	900.00
3004	Café Italy	Extra Dark Italian Roast	Regular	525.00
3004	Café Italy	French Vanilla	Regular	540.00
3004	Café Italy	Irish Crosse	Regular	900.00

5. Click the **Total** number, and it will drill down to the next detail report.

Sales Peformance by Quarter

Sales Quarter: Katherine Chen
Sales Year: 1998

Order Quarter: 2 **Total:** $4,777.90

Order ID	Customer Name	Product Name	Product Category	Total Sales
3031	American Coffee Inc.	Mocha Java	Regular	318.00
3031	American Coffee Inc.	Colombian	Decaf	183.50
3031	American Coffee Inc.	Expresso Roast	Decaf	86.40
3031	American Coffee Inc.	Java Hut Select	Regular	170.00
3007	Cafè International	Vienna Roast	Regular	285.00
3007	Cafè International	Almond Tosted	Regular	420.00
3007	Cafè International	Butternut Rum	Regular	450.00
3007	Cafè International	Vanilla Hazelnut	Regular	900.00
3004	Cafè Italy	Extra Dark Italian Roast	Regular	525.00
3004	Cafè Italy	French Vanilla	Regular	540.00
3004	Cafè Italy	Irish Crosse	Regular	900.00

Note: When using JHyperLink to link reports as master and detail reports, if a parameter with its value containing special characters such as "\" and "&" is used in the URL, you need to add "\" before the special character. For example, if the parameter value is a&b, you need to enter it as a\&b.

Rotating text/image with JRotator

JRotator is used to rotate text or images. Since JReport currently supports rotating only images but not text, using JRotator to achieve text rotation becomes a feasible solution. JRotator rotates the text inside the JRotator object but not the object itself so if you rotate text or DBFields you may have to change the size of the container in order to see the full value.

The following introduces three cases of using JRotator:

Using JRotator to rotate text

1. Start JReport Designer and open the report in which you want to insert some rotated text.
2. On the menu bar, click **Insert** > **UDO**.
3. In the Insert UDO dialog, select **JRotator** from the drop-down list, and then click **OK**.
4. Keep the JRotator focused and go to the Report Inspector.
5. Locate the Display Value property and type in the required text in the value cell, then click outside of the cell. You will then see the typed text displayed in the JRotator.
6. Specify a rotation degree in the value cell of the Rotate property, for example, 180, then click outside of the cell. You will then find the text in the JRotator upside down.
7. Save the report.

Using JRotator to rotate an image

1. Start JReport Designer and open the report in which you want to insert a rotated image.
2. On the menu bar, click **Insert** > **UDO**.
3. In the Insert UDO dialog, select **JRotator** from the drop-down list, and then click **OK**.
4. Keep the JRotator focused and go to the Report Inspector.
5. Specify the local path of the required image in the value cell of the Display Image property, for example, *C:\image.jpg*, then click outside of the cell. You will then see the image displayed in the JRotator.
6. Specify a rotation degree in the value cell of the Rotate property, for example, 45, then click outside of the cell. You will then find the image in the JRotator at a 45 degree angle.
7. Resize the JRotator container as needed to view the entire image.
8. Save the report.

Using JRotator to rotate a DBField

1. Start JReport Designer and open the report in which you want to insert a rotated DBField.
2. On the menu bar, click **Insert** > **UDO**.
3. In the Insert UDO dialog, select **JRotator** from the drop-down list, and then click **OK**.
4. Keep the JRotator focused and go to the Report Inspector.
5. In the Display Value value cell, select the required DBField or create a formula for a calculated field and select the formula from the dropdown list. Then click outside of the cell and you will see the DBField displayed in the JRotator.
6. Specify a rotation degree in the Rotate value cell, for example, 90, then click outside of the value cell. You will then find the DBField in the JRotator at a 90 degree angle.
7. Change the size of the JRotator container as needed to view the entire text.
8. Save the report.

Page Reports

A page report is composed of one or more report tabs that are designed for the same purpose, or related purposes. Report developers can design, maintain, run and schedule these report tabs together or separately. Report tabs in the same page report can share the same parameters and dataset.

A page report tab is a report sheet in a page report. It is not only a passive document used only for reading or viewing but also a business document that contains the predefined data. It is a form that typically contains data from many unrelated records or transactions. JReport report tabs are usually composed of different components, with each component serving different purposes. They have the same functions as with general reports, but also include additional unique qualities.

Comparing with the single-report management system before, the page report feature is designed to improve the product performance, avoid wasting data sources by sharing them in the page report, and manage and maintain reports more easily for the report developers.

Pick a task from below:

- Creating a page report
- Opening a page report
- Creating report tabs in a page report
- Editing a page report
- Designing a report page
- Previewing a page report
- Importing report tabs into a page report
- Managing report tabs in a page report
- Making high-efficiency reports
- Saving a page report
- Configuring page reports for specific Page Report Studio features
- Page Report Manager

Creating a page report

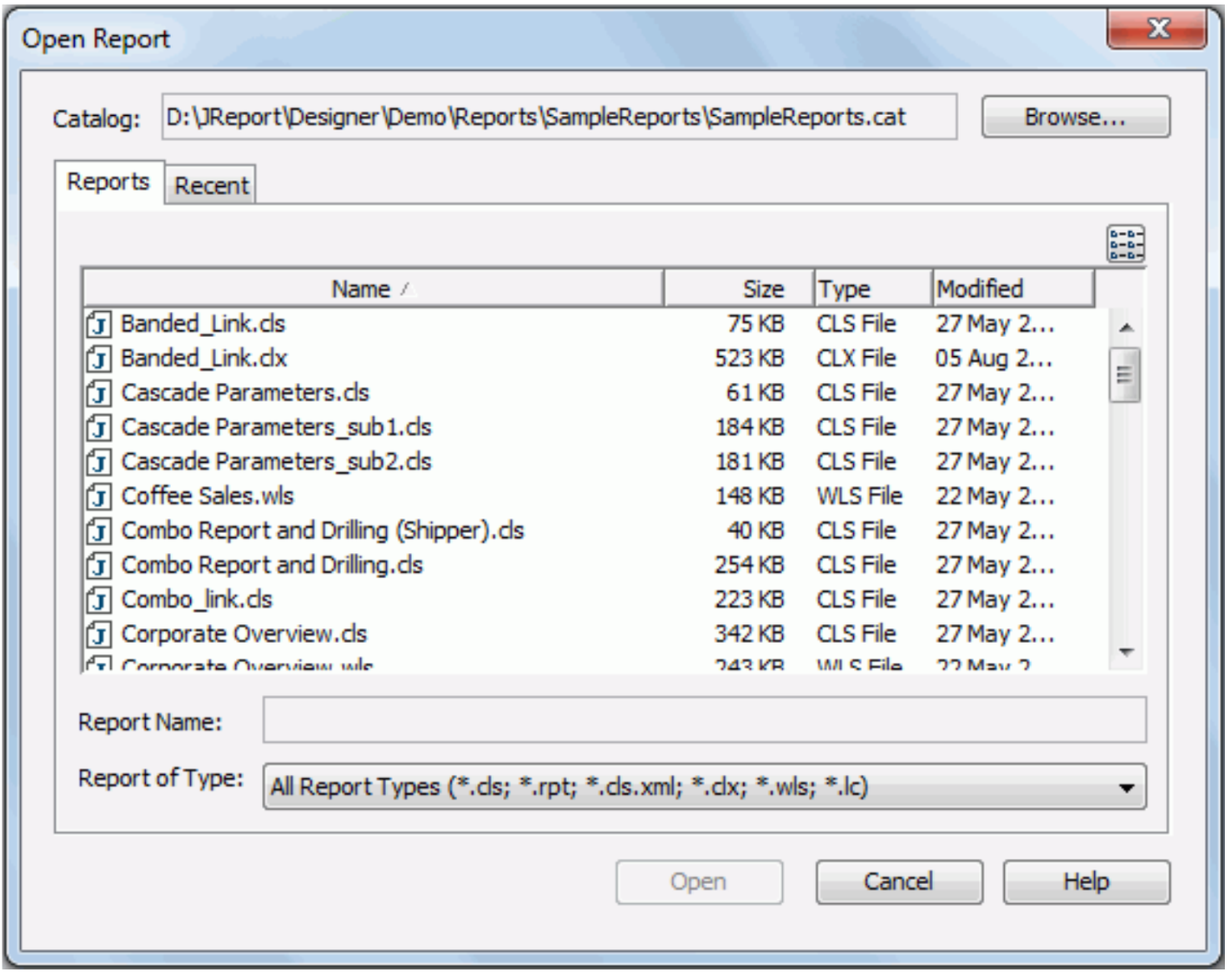
To create a page report, follow the steps below:

1. Click **File** > **New** > **Page Report** on the menu bar, or the **New Page Report** button on the Standard toolbar.
2. In the New Page Report dialog, select the layout of the first report tab in the page report.

   Since a page report cannot be empty, you will also need to create a report tab in it when creating the page report. See Creating report tabs in a page report for details about how to create the first report tab in a page report.
3. Click **OK** to create the page report.

Opening a page report

1. Click **File** > **Open** on the menu bar, or click the **Open** button on the Standard toolbar to display the Open Report dialog.



2. Click **Browse** to select a catalog of your choosing.

3. In the Reports tab, specify the corresponding type for the page report that you want to open in the Report of Type drop-down list, and then select one or more page reports in the list for opening.

   If you want to open page reports that you have opened recently, switch to the **Recent** tab, and then select the page reports you want to open.

4. Click **Open** to open the page reports that you specified.

Notes:

- If you just want to access some recently open page reports, click **File** > **Open Recent** and then select the page reports one by one from the submenu.
- When you open a self-contained page report, JReport will prompt you to specify the way in which you want the report to be opened. For details, refer to the Open As dialog.

Creating report tabs in a page report

Before you can create report tabs, you first have to create a page report or open an existing page report.

JReport provides you with eight report layouts as follows, which are designed for serving different reporting requirements:

- **Standard Banded Report**
  A report tab containing a vertical banded object.
- **Table Report**
  A report tab containing a table.
- **Chart Report**
  A report tab containing a chart.
- **Crosstab Report**
  A report tab containing a crosstab.
- **Horizontal Banded Report**
  A report tab containing a horizontal banded object.
- **Mailing Label Report**
  A report tab containing a banded object, which is in the form of a mailing label layout.
- **Tabular Report**
  A report tab with a tabular which partitions the report page into several parts.
- **Blank Report**
  A report tab with nothing in it.

The following topics explain in detail about how to create a report tab in a specific layout:

- Creating a banded report
- Creating a table report
- Creating a chart report
- Creating a crosstab report
- Creating a tabular report
- Creating a blank report

Creating a banded report

When creating a banded report, you can choose in which layout the banded object in the report will be, that is, you can choose whether to create a standard banded report, a horizontal banded report or a mailing label report, according to your requirements.

The steps for creating a standard banded report, horizontal banded report and mailing label report are mostly the same, just that the banded object is positioned differently in each report layout: one in a vertical way, one horizontally and one in the form of mailing labels.

Assume that you have already opened a page report, to create a standard banded/horizontal banded/ mailing label report within it:

1. Do either of the following:
   - Click **File** > **New** > **Page Report Tab** on the menu bar.
   - On the report tab bar, right-click an existing report tab in the report and click **Insert** on the shortcut menu.
2. In the New Page Report Tab dialog, specify the title of the report tab, select the **Banded**/ **Horizontal Banded**/**Mailing Label** layout and click **OK**. The corresponding report wizard appears.
3. In the Data screen of the wizard, specify the dataset on which the banded object will be built.
   - To build the banded object on a new dataset created from the catalog resources, check the **New** radio button, and then select the data source of the dataset. If the given data resources are not what you want, click the **<Add xxx...>** item to create one. In addition, if you choose to create the dataset from a query, you can click the **Modify** button to further modify the query if required.
   - To build the banded object on an existing dataset in the report, check the **Existing** radio button, and then select the dataset as required (click the **Modify** button to modify the selected dataset in the Dataset Editor if necessary). You can also click the **<Add Dataset...>** item to create one as required.
4. In the Display screen, add the required fields from the Resources box to be displayed in the banded object. Modify the display name of any added field if necessary by double-clicking its Display Name cell. And if required, click the **Sort Fields By** button to specify how to sort data in the banded object (for details, refer to the same topic on table).
5. In the Group screen, add the fields as the grouping criteria, and then specify the sorting direction of each group in the Sort column (for details, see the same topic on table).
6. To add summaries, go to the Summary screen. In the sum on box (the box on the right), specify the group to which the summary will be applied (if you select Banded Object, the summary will be based on the whole dataset), add a field from the Resources box as the summary field, and then select the function for the summary from the Aggregate Function column.
7. If you want to create a chart together with the banded object in the banded header panel, go to the Chart screen. Select the chart type you like. From the Category drop-down list, select the field you want to display on the category (X) axis of the chart. You can choose from the group by fields of the banded object on which some summaries are added. Choose the field to be displayed on the series (Z) axis of the chart from the Series drop-down list, where all the group by fields of the banded object are listed. From the Show Values drop-down list, select the field you want to display on the value (Y) axis of the chart. You can choose from the summary fields which are calculated based on the field you specify to display on the category axis of the chart.
8. To apply some filters to the banded object so as to reduce the data displayed in the banded object, go to the Filter screen and define the filter conditions.
9. If you are creating a mailing label report, in the Layout screen, specify the label size and the gap between the labels.
10. In the Style screen, specify the title and style of the banded object, and if required, click the **Page Setup** button to set the page settings.
    - **Create a Banded Report**
      Creates a restricted version of a flow layout report tab, one that can contain a single banded object only. This layout option is intended to provide compatibility with JReport Version 7 report templates.
    - **Create a Flow Layout Report Containing a Banded Object**
      Creates a report tab that contains a banded object mapped to the specified dataset. This layout option is specified by default.
11. When done, click **Finish** to create the report tab.

Notes:

- When you want to create a banded report on a hierarchical data source, pay attention to the unique features of HDS. For details, refer to Developing reports from HDS.
- If you have added several group by fields to the banded object in the Group screen, when the report is created, these groups will be indented according to the Customize group indent option setting in the Options dialog. You can configure this option in advance to have the groups indented according to your requirements.
- If you have added some summaries to the banded object in the Summary screen, when the report is created,
  - JReport will automatically give the summaries some proper name labels to help you clarify the meaning of the numbers. The auto generated labels are controlled by the Text property in the Report Inspector. You can also manually edit the text of the label later by setting this property.
  - By default, the summaries that have the same function will be aligned horizontally in the banded object. If you want to align summaries of the same function vertically, check the option "Align summaries vertically" in the Component category of the Options dialog in advance.

See also:

- Banded Wizard for detailed explanation about options in this wizard.
- Horizontal Banded Wizard for detailed explanation about options in this wizard.
- Mailing Label Wizard for detailed explanation about options in this wizard.
- Banded objects for details about the banded object component.
- **An example:** The SampleComponents catalog, included with JReport Designer, contains reports that have examples of report layout types. For the banded report layout example, open `<install_root>\Demo\Reports\SampleComponents\BandedObjectReport.cls`.

Creating a table report

Assume that you have already opened a page report, to create a table report within it:

1. Do either of the following:
   - Click **File** > **New** > **Page Report Tab** on the menu bar.
   - On the report tab bar, right-click an existing report tab in the report and click **Insert** on the shortcut menu.
2. In the New Page Report Tab dialog, specify the title of the report tab, select the required table type: Group Above, Group Left, Group Left Above, or Summary Table, then click **OK**. The Table Wizard appears.
3. In the Data screen of the wizard, specify the dataset on which the table will be built.
   - To build the table on a new dataset created from the catalog resources, check the **New** radio button, and then select the data source of the dataset. If the given data resources are not what you want, click the **<Add xxx...>** item to create one. In addition, if you choose to create the dataset from a query, you can click the **Modify** button to further modify the query if required.
   - To build the table on an existing dataset in the report, check the **Existing** radio button, and then select the dataset as required (click the **Modify** button to modify the selected dataset in the Dataset Editor if necessary). You can also click the **<Add Dataset...>** item to create one as required.

     Note: When you want to create a table report on a hierarchical data source, pay attention to the unique features of HDS. For details, refer to Developing reports from HDS.
4. In the Display screen, add the required fields from the Resources box to be displayed in the table. Modify the display name of any added field if necessary by double-clicking its Display Name column. If required, click the **Sort Fields By** button to specify how to sort data in the table.
5. In the Group screen, add the fields as the grouping criteria, and then specify the sorting direction of each group in the Sort column.
6. To add summaries, go to the Summary screen. In the sum on box (the box on the right), specify the group to which the summary will be applied (if you select Table, the summary will be based on the whole dataset), add a field from the Resources box as the summary field, and then select the function for the summary from the Aggregate Function column. From the Position and Column columns, specify where to place the added sum on field in the table.
7. If you want to create a chart together with the table in the report body, go to the Chart screen. Select the chart type you like. From the Category drop-down list, select the field you want to display on the category (X) axis of the chart. You can choose from the group by fields of the table on which some summaries are added. Choose the field to be displayed on the series (Z) axis of the chart from the Series drop-down list, where all the group by fields of the table are listed. From the Show Values drop-down list, select the field you want to display on the value (Y) axis of the chart. You can choose from the summary fields which are calculated based on the field you specify to display on the category axis of the chart.
8. To apply some filters to the table so as to reduce the data displayed in the table, go to the Filter screen and define the filter conditions.
9. In the Style screen, specify the style of the table, and if required, click the **Page Setup** button to set the page settings.
10. When done, click **Finish** to create the report tab.

See also:

- Tables for details about the table component.
- **An example:** The SampleComponents catalog, included with JReport Designer, contains reports that have examples of report layout types. For the table report layout example, open `<install_root>\Demo\Reports\SampleComponents\TableReport.cls`.

Creating a chart report

Assume that you have already opened a page report, to create a chart report within it:

1. Do either of the following:
   - Click **File** > **New** > **Page Report Tab** on the menu bar.
   - On the report tab bar, right-click an existing report tab in the report and click **Insert** on the shortcut menu.
2. In the New Page Report Tab dialog, specify the title of the report tab, select the **Chart Report** layout and click **OK**. The Chart Wizard will then be displayed.
3. In the Data screen of the wizard, specify the dataset on which the chart will be built.
   - To build the chart on a new dataset created from the catalog resources, check the **New** radio button, and then select the data source of the dataset. If the given data resources are not what you want, click the **<Add xxx...>** item to create one. In addition, if you choose to create the dataset from a query, you can click the **Modify** button to further modify the query if required.
   - To build the chart on an existing dataset in the report, check the **Existing** radio button, and then select the dataset as required (click the **Modify** button to modify the selected dataset in the Dataset Editor if necessary). You can also click the **<Add Dataset...>** item to create one as required.
4. In the Type screen, specify the type of the chart.
   - To create a single type chart, check the **Single Type** radio button, and select the required type from the Chart Type box and its sub type accordingly.
   - To create a combo type chart, check the **Combo Type** radio button, and the click the **<Add Combo Type>** item to add the required combo type to Primary Axis and Secondary Axis respectively.
5. The Display screen varies with chart types:
   - **For common chart types**

     Add the required fields or additional values from the Resources box to be displayed on the category (X) axis, series (Z) axis, and value (Y) axis of the chart, and define some special group, special function and Order/Select N condition on the category/series field if required.

     If you select the bubble chart type, you need to specify the fields to be shown on the bubble X axis, Y axis and the value you want to show as the bubble radius in the Show Values box. Note that when you specify a value for the bubble X axis, this value will be displayed on the category axis instead of the one specified in the Category box. However, the value defined in the Category box will also be included in data calculation.
   - **For org chart**
     a. Select a field from the Child drop-down list which defines the entities.
     b. Select a field from the Parent drop-down list which defines the ownership or reporting to relations among the entities. For example, if the child field is Employee ID, the parent field can be the one about IDs showing which employ ID reports to which employ ID. Note that the parent field should use different one from the child field.
     c. The Properties box displays a node model in the org chart. You can add objects, including data fields, labels and images, from the Resources box into the node and then adjust their positions and sizes in the node and the size of the node. Those added objects will be displayed in each node as the information about the entity members.
        - To add an object, dragging it from the Resources box and dropping to the specific position in the node area in the Properties box, or select it from the Resources box and then click .
        - To remove an object from the node, select it and then click or press Delete on the keyboard.

     For more information about this chart type, see Organization chart.
   - **For heat map**
     a. Add the fields used to group the data into the Groups box. Define some special group, special function and Order/Select N condition on the group field if required.
     b. Add proper summaries into the Summaries box.

        The summaries should match the groups. For example, if the groups level is A > B > C, the static summaries grouped by C can be inserted into the Summaries box, but the static summaries grouped by A, B or other fields cannot.

        If no group is specified in the Groups box, you can insert any static summary. And its group-by field will be inserted into the Groups box automatically.
     c. From both Groups and Summaries boxes, specify the fields to do color by and size by, and select the fields as label-by fields to display in the innermost rectangle in the heat map.

     For more information about this chart type, see Heat Map.
6. To apply some filters to the chart so as to reduce the data displayed in the chart, go to the Filter screen and define the filter conditions.
7. In the Layout screen,
   - For common chart types, with the Data category selected in the Options list box, you can specify the category/series start and end offset, whether or not to reverse the category/series field value sequence, and whether or not to swap groups. Besides, you can set whether or not to show the titles, axes, walls, gridlines, and legend. If yes, set the titles for the chart, the category axis and the value axis respectively. If you choose not to display a certain axis, the labels and tick marks along the axis will be hidden as well.
   - For organization chart, in the Type category, choose a way of expanding the org chart tree. Then configure the other settings if required.
   - For heat map, you can specify the chart title and whether to show the chart legend.
8. In the Style screen, specify the style of the chart, and if required, click the **Page Setup** button to set the page settings.
9. When done, click **Finish** to create the report tab.

Notes:

- When you want to create a chart report on a hierarchical data source, pay attention to the unique features of HDS. For details, refer to Developing reports from HDS.
- If you specify to create a pie chart, you can add several fields to be displayed on the value axis of the chart as long as the fields are of Numeric type. However, you can add formulas and DBFields to the axis at the same time, but when a summary has been added to the axis, no other DBFields or formulas can be added.
- If you have defined a subtype of the Stock type, in the Display screen, you should add fields for the subtype in the order indicated by the subtype name, for example, if you define the Open-High-Low-Close 2-D subtype, then you should add four fields to the Show Values box for this subtype, and arrange them to the order of Open-High-Low-Close by clicking the buttons and .

See also:

- Charts for details about the chart component.
- **An example:** The SampleComponents catalog, included with JReport Designer, contains reports that have examples of report layout types. For the chart report layout example, open `<install_root>\Demo\Reports\SampleComponents\Chart.cls`.

Creating a crosstab report

Assume that you have already opened a page report, to create a crosstab report within it:

1. Do either of the following:
   - Click **File** > **New** > **Page Report Tab** on the menu bar.
   - On the report tab bar, right-click an existing report tab in the report and click **Insert** on the shortcut menu.
2. In the New Page Report Tab dialog, specify the title of the report tab, select the **Crosstab** layout and click **OK**. The Crosstab Wizard appears.
3. In the Data screen of the wizard, specify the dataset on which the crosstab will be built.
   - To build the crosstab on a new dataset created from the catalog resources, check the **New** radio button, and then select the data source of the dataset. If the given data resources are not what you want, click the **<Add xxx...>** item to create one. And if you choose to create the dataset from a query, you can click the **Modify** button to further modify the query if required.
   - To build the crosstab on an existing dataset in the report, check the **Existing** radio button, and then select the dataset as required (click the **Modify** button to modify the selected dataset in the Dataset Editor if necessary). You can also click the **<Add Dataset...>** item to create one as required.

     Note: When you want to create a crosstab report on a hierarchical data source, pay attention to the unique features of HDS. For details, refer to Developing reports from HDS.
4. In the Display screen, add the required fields from the Resources box to the Columns and Rows boxes respectively to make them displayed on the columns and rows of the crosstab. For the added fields, you can specify their display names in the Label column which will label the rows and columns (by default the Label column is blank and no label will be created for the rows and columns), the background color in the Color column, set the sorting manner of the field values by clicking , and adjust their display order by clicking or . Then, add the fields as aggregate fields to the Summaries box, specify the aggregate functions for the added fields, and edit the display names of the fields in the Label column to label the summaries. If necessary, click the **Comparison Function** button to set the comparison function for the aggregate fields. You can also create some crosstab formulas and add them as aggregate fields so as to apply custom aggregate functions in the crosstab.
5. If you want to create a chart together with the crosstab in the report body, go to the Chart screen. Select the chart type you like, from the Category and Series drop-down lists, where all the fields that have been added to the columns and rows of the crosstab are listed, choose the fields to be displayed on the category (X) axis and series (Z) axis of the chart, then select the field you want to display on the value (Y) axis of the chart from the Show Values drop-down list, you can choose from all the aggregate fields in the crosstab.
6. To apply some filters to the crosstab so as to reduce data displayed in the crosstab, go to the Filter screen and define the filter conditions.
7. In the Layout screen, specify the layout of the crosstab.
8. In the Style screen, specify the style of the crosstab, and if required, click the **Page Setup** button to set the page settings.
9. When done, click **Finish** to create the report tab.

See also:

- Crosstabs for details about the crosstab component.
- **An example:** The SampleComponents catalog, included with JReport Designer, contains reports that have examples of report layout types. For the crosstab report layout example, open `<install_root>\Demo\Reports\SampleComponents\Crosstab.cls`.

Creating a tabular report

A tabular report makes you to better design the report page. It is in fact a report tab with a tabular which partitions the report page into several parts. You can insert objects and texts into the tabular cells, and their layout will be controlled by the cells.

Assume that you have already opened a page report, to create a tabular report within it:

1. Do either of the following to open the New Page Report Tab dialog.
   - Click **File** > **New** > **Page Report Tab** on the menu bar.
   - On the report tab bar, right-click an existing report tab in the report and click **Insert** on the shortcut menu.
2. Specify the title of the report tab and select the **Tabular** layout, then click **OK**.
3. In the Tabular Wizard, specify the number of rows and columns, and the border width and tabular width according to your requirements.
4. Click **Finish** to create the report tab.

When a tabular report is created, you can then insert components and texts into the tabular cells as required, and their layout will be controlled by the cells. A tabular report helps you better design the report page. For details about how to work with tabulars, refer to Tabulars.

Creating a blank report

Assume that you have already opened a page report to create a blank report tab within it:

1. Do either of the following to open the New Report Tab dialog.
   - Click **File** > **New** > **Page Report Tab** on the menu bar.
   - On the report tab bar, right-click an existing report tab in the report and click **Insert** on the shortcut menu.
2. Specify the title of the report tab, select the **Blank Report** layout and click **OK**.
3. In the Choose Data dialog, specify the dataset on which the report tab will be built.
   - To build the report tab on a new dataset created from the catalog resources, check the **New** radio button, and then select the data source of the dataset. If the given resources are not what you want, click the **<Add xxx...>** item to create one. And if you choose to create the dataset from a query, you can click the **Modify** button to further modify the query if required.
   - To build the report tab on an existing dataset in the report, check the **Existing** radio button, and then select the dataset as required (click the **Modify** button to modify the selected dataset if necessary). You can also click the **<Add Dataset...>** item to create one as required.
4. When done, click **OK** and a blank report tab will then be created. You can then insert the required objects into the report tab as described in the Components chapter.

•

Editing a page report

After a page report is built up, you can make further modifications to it, in order to make it better satisfy your requirements. In JReport, to edit a page report is actually to edit the components in its report tabs. This section only lists the basic operations used for editing a page report. For more details, you can refer to the specific section of each component in the chapter Components.

Note: The other way to edit a page report is by changing its properties in the Report Inspector. For detailed property descriptions, see Properties in the Report Inspector.

Inserting other components into a page report

After the initial design work of a page report, you are also free to insert more components into it. To insert a component, you can either use the menu command or the Toolbox panel.

Duplicating the components

You can duplicate any component in a page report without creating it the second time.

1. Select the component which you want to duplicate.
2. Right-click the component and then select **Copy** from the shortcut menu (or click the corresponding menu command or toolbar button).
3. In the required position, right-click your mouse, and then click **Paste**.

Resizing the components

To resize a component in a page report, follow the steps below:

1. Select the component with the size you want to change, and the sizing handles will then appear along the four boundaries of the component.
2. Drag a sizing handle to resize the component to the required size.
3. Repeat the steps to resize other components.

Showing components in a TOC tree

Some components in a page report can be set up to show in a table of contents. Then when previewing the report in JReport Format, viewing it in Page Report Studio, exporting it to PDF or HTML format, TOC Browser can be used to locate the corresponding component.

To show a component in a TOC tree:

1. Select the component node in the Report Inspector.
2. Set its TOC Anchor property to **true**.
3. Specify the display value for the anchor by setting the Anchor Display Value property.

JReport also allows you to specify other properties for a TOC tree, including its CSS style, colors and fonts. You can specify these properties by selecting the TOC node in the Report Inspector.

Changing the position of components

Report templates in JReport use a flow layout model. That is, paragraphs and components in the report body can flow from one page to another, maintaining their sequence, and the Position property controls whether a component is to be part of the flow, or separate from it. Besides the report body, tabular cells and text boxes themselves in JReport can also act as flow layout containers.

In a flow layout model, objects are positioned relative to one another or absolutely. The Position property controls the position of components in the flow layout container, which can be the report body, a tabular cell or a text box. Components placed in other areas, such as table cells or banded panels are not affected by the Position property.

The Position property can be one of the following values:

- **Static**
  The component will be positioned at the location in which it is inserted. The X and Y coordinate properties are disabled. The position of the component cannot be moved other than by having its insertion point moved. This can happen when the text flow preceding the insertion point expands.
- **Relative**
  The component will be positioned at an offset to the position at which it is inserted. The offset is determined by the X and Y coordinate property values. This value is not available for some kinds of component.
- **Absolute**
  The component will be located at the position specified by dragging and dropping or by setting its X and Y coordinate property values. The component insertion point does not change, for instance it is not affected by having text inserted before it.

  When components in the same flow layout container have their Position property value set to Absolute and overlap, you can set the display order of the objects by right-clicking the object and then selecting an item from the Move submenu (or clicking the corresponding move button on the Alignment toolbar).

  - **Move Backward**
    Moves the component one step towards the back.
  - **Move to Back**
    Moves the component to the back.
  - **Move Forward**
    Moves the component one step closer to the front.
  - **Move to Front**
    Moves the component to the front.

To change the position of a component:

- In the design area, right-click the component and on the shortcut menu, select the required item from the Position submenu.
- In the Report Inspector, select the node which represents the component, then set its Position property value as required.

Deleting components from a page report

If you find any components contained in a page report are not required, you can use either of the following two ways to remove it from the report tab:

- In the Report Inspector, select the node which represents the component and press the **Delete** button on your keyboard.
- In the design area, right-click the component and select **Delete** from the shortcut menu.

Designing a report page

A report page is a medium which contains the layout result for a single page of a report tab. At runtime, a page or multiple pages can be generated in a report tab according to the page panel settings. Also, each page has its own page headers and page footers, which are controlled by the page panel settings for them.

A report page has the following features.

Page panel

A page panel is a template to design a page. You can define information for it and meanwhile information for pages designed with this page panel will be defined accordingly. Here, information covers all factors of a page, including page size, margin, and orientation. Also, you can define the page headers and page footers for a page panel. By default, a page panel has only one page header and page footer, but you can also set more and remove them according to your requirements.

In addition, JReport Designer also supports multiple page panels. With multiple page panels, you have to define each page panel's page settings individually.

Inserting a page panel

When inserting a page panel into a page, you can choose whether the new page will keep the original page settings or it will apply new settings.

- **Inserting a page panel where the new page keeps the original page settings**
  1. Click **Insert** > **Page Panel**.
  2. In the Page Panel dialog, select **Keep the current page settings**.
  3. Click **OK** to insert the panel.
- **Inserting a page panel where the new page has new page settings**
  1. In the Page Panel dialog, select **Use a new page setting**.
  2. Click **OK**. The Page Setup dialog will then appear for you to adjust the settings for the new page panel.

Removing a page panel

To remove a page panel, set the mouse pointer to the beginning of the page panel and then press the **Backspace** key.

Notes:

- Page panels can be inserted into the report body level.
- Page panels can be inserted into banded reports which satisfies the following conditions:
  - The parent of the banded object is the report body.
  - The Position property value of the banded object is set to static.
  - There isn't any existing page panel in the report body.

Page headers and footers

Page headers and footers are areas in the top and the bottom page margins, where you can add any JReport components. They are added to the current page settings. Any page that uses the same settings automatically receives the header or footer that you add. You can insert fields, such as page numbers and chapter headings, in headers and footers in a text document.

A page header and page footer will be generated at the same time when a report tab is created, except for banded layout reports. For a banded layout report, a page header and page footer can only be generated after you insert a page panel into a banded panel.

You can choose to show or hide them according to your own requirements by checking or unchecking the View > Page Header or View > Page Footer command.

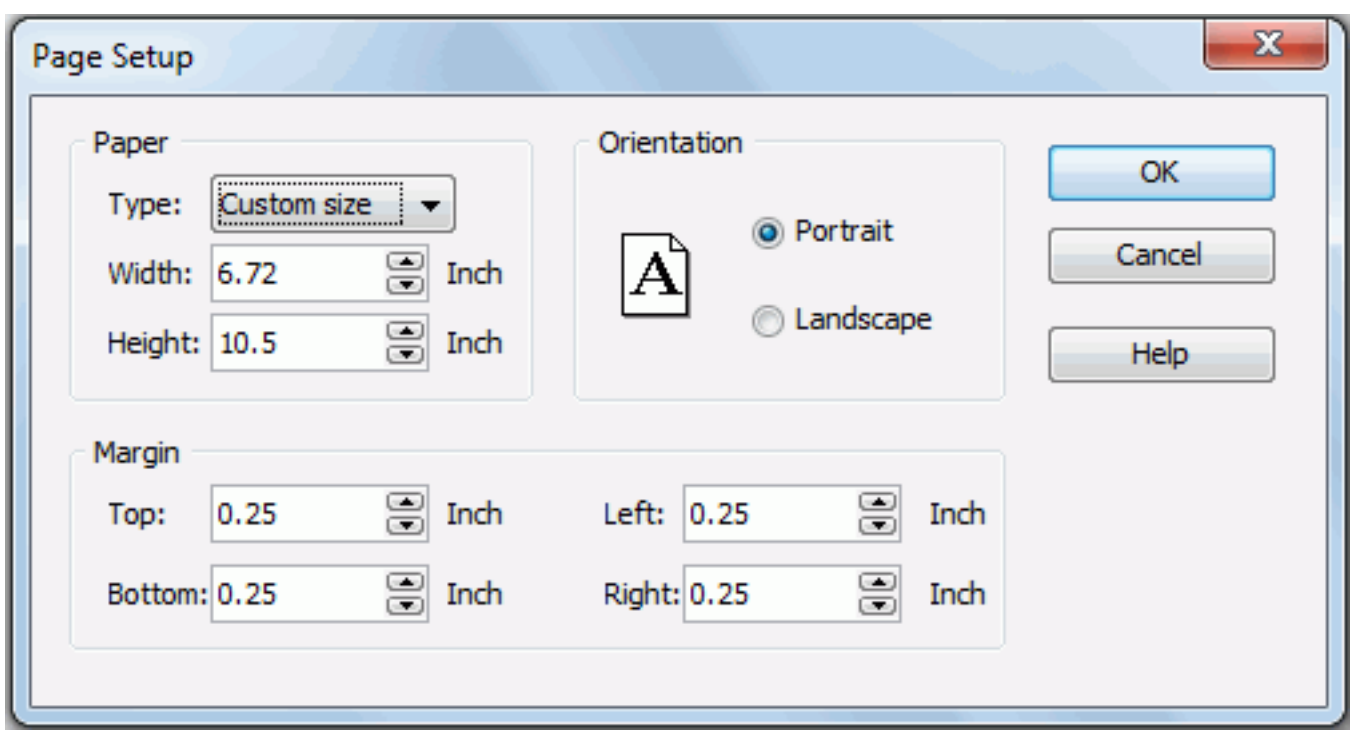
Page settings

Settings for a page are the same as with a page panel which is used as a template to design this page. All pages designed with the same page panel will apply the same page settings. So, you just need to set the settings for a page panel to apply them to these pages.

Adjusting the page settings

To adjust the page settings, the following are two ways for you to choose:

- **Adjusting through the Page Setup dialog**
  1. Click **File** > **Page Setup**. The Page Setup dialog appears.



  2. Set properties of paper, orientation and margin for the current page.
  3. Click **OK** to apply your settings.
- **Adjusting through the Report Inspector**
  1. Click **View** > **Report Inspector**.
  2. In the Report Inspector, select a PagePanel node with the properties you want to set.
  3. Set properties of Paper, Orientation and Margin for the specified page panel node according to your requirements.

Page break

A new page will be generated in a report tab only when the space in a page has been fully occupied or there exists a break control in this page, for example, a page break that is inserted in the page.

To insert a page break, click **Insert** > **Page Break**.

Notes:

- Page breaks can be inserted into the report body level.
- Page breaks can be inserted into banded objects only when the following conditions have been met:
  - The parent of the banded object is the report body.
  - There isn't any existing page panel in the report body.
  - Its Position property of the banded object is set to static.

Page mode

In JReport Designer, you can set the page mode of a report tab: pagination mode and continuous mode. This section describes these two page modes and how to switch between them.

Pagination mode

In pagination mode, the page shape and size can be specified in the page panel settings. Once set, the contents of the report tab will be laid out according to these specifications. If the contents exceed the specified size, the page break control will start a new page with the same size.

In addition, the total page size is determined both by the contents' size and by the page breaking control, such as Fill Whole Page, On New Page, Page Breaker.

Continuous mode

In continuous page mode, only the component breaking control will work. The whole report tab is laid out in a single page, and has no specific shape and size. If no component breaking control has been set, all of the components will be laid out in the same page, and the size of the page will depend on the size of the contents.

In addition, all of the properties set in pagination mode will be simply skipped, including the page breakers in page panel. In this way, if there is more than one page panel in a report tab, all contents contained in different page panels will be laid out in the same page, and only the first page panel's page header and footer will be laid out.

Component breaking rule

In continuous page mode, JReport Designer provides you with an option for breaking a component in continuous page mode. This kind of breaking is not based on space, but on logic. For example, you can set a group break to occur every 20 detail sections, no matter how big these 20 detail sections are.

The following three components have logic break functionality: banded object, table and crosstab:

- For banded objects, the logic break acts only on the sub-components - detail panel and group panel.
- For tables, the logic break acts only on the sub-components - table detail and table group.
- For crosstabs, the logic break acts on the whole crosstab.

And, several properties are related to the component logic break:

- For banded objects and tables, there are two relative properties: Current Block Index and Items per Block.
- For crosstabs, besides the two properties mentioned above, there are also four extra properties: Current Row Block Index, Current Column Block Index, Items per Row Block and Items per Column Block.

Switching between the two page modes

To switch between the two page modes, you can choose either of the following ways:

- Select or unselect **Page Layout** on the View menu.
- Click the **Page Layout** button on the Report toolbar.

Previewing a page report

In the process of designing a page report, you can preview the report tabs in it at any time in the following two ways:

- Previewing a report tab in JReport format
- Previewing a report tab in other formats

Moreover, when previewing a report tab, if it is developed with National Language Support, you can specify the language in which you want the report tab to display; if it contains parameters, you will be prompted to specify values for the parameters in the Enter Parameter Values dialog.

Previewing a report tab in JReport format

To preview a report tab in JReport format when designing it at any time, just click the **View** tab, which then causes JReport Engine to run the report tab. The result is then shown in the viewer. To optimize performance, JReport Designer displays report pages before they have been completely generated. For details about the options in the View tab, see Design/View area.

In addition, when previewing a banded object, you are enabled to dynamically sort the records and show details of aggregate level fields.

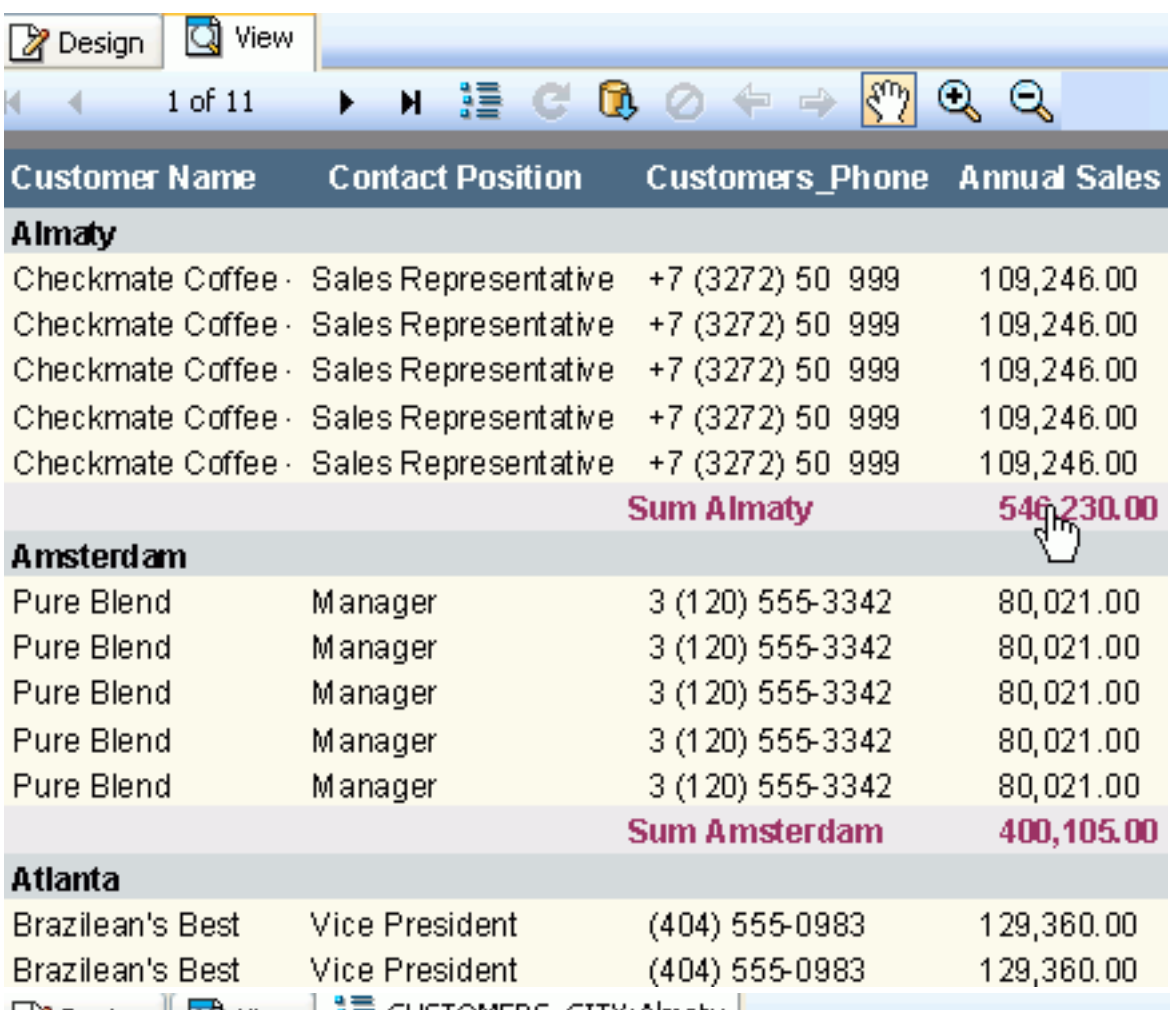
Dynamic sort

You can sort the records in a banded object dynamically in the View area. To do this:

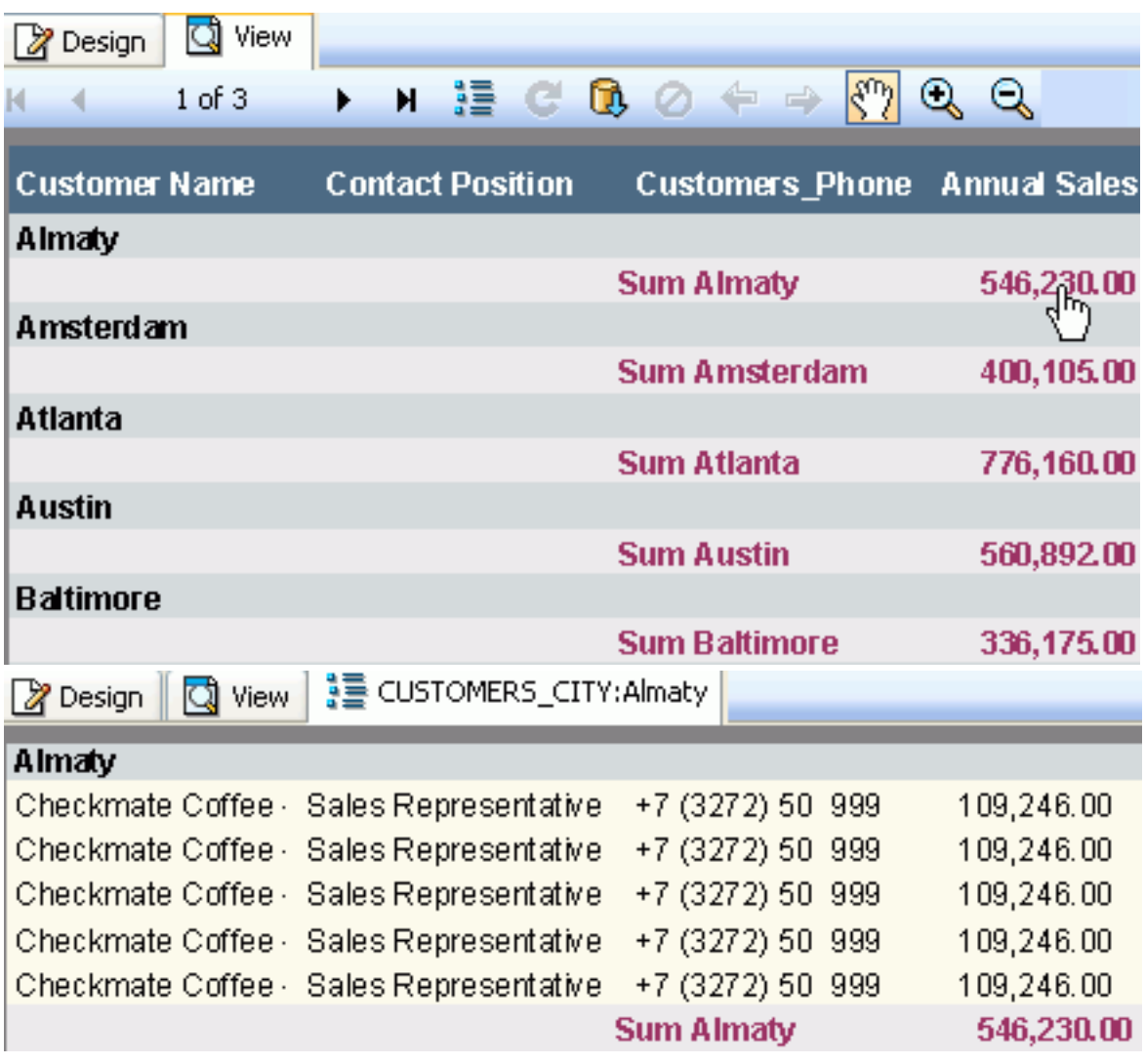
1. Point to the field you want to sort.
2. Right-click the field and select the option **Sort Ascending** or **Sort Descending** from the shortcut menu.

Showing aggregation detail

You can show details of aggregate level fields in a banded object. Showing detail is made possible where there is a summary for a group. For example, if a banded object has been grouped by Customer City, and the group summary is on Annual Sales. When previewing the banded object, if you point the mouse at the total annual sales for a city, it will turn into a small hand. Click on the total, and the detail report for the selected city will be displayed.



By default, in cases like the above, JReport's convention is to hide the detail panel of each group and only show it in a detail report. To do this, select the detail panel, go to the Report Inspector and set its Invisible property to **true** (make sure the Suppress property is false). Then view the banded object again, and it will be shown without details. If you want to see the details of a certain city, point the mouse at the city's total annual sales, when the mouse turns into a small hand, click on the total, and the detail report for the city will be displayed.



Note: After some modifications in the design area, if you find the report tab is shown very quickly and the changes are not applied in view mode, this means you need to click the Refresh Data button on the toolbar to re-run your report tab.

Previewing a report tab in other formats

JReport also allows you to preview your report tab in the following formats: Page Report Result, HTML, PDF, Excel, Text, RTF, XML and PostScript.

To preview a report tab in other formats, take one of the following:

- On the View menu, select the specific format from the Preview As submenu.
- Click the **Preview As** button on the Standard toolbar and select the required item from the drop-down list.
- On the File menu, select **Export To** and select the format you want to use.

By selecting one of the formats in the View > Preview As menu, the report tab will be exported to a temporary file which will automatically be opened in the format with the appropriate application. The temporary file will be removed when you exit JReport Designer.

By selecting one of the formats in the File > Export To menu, the report tab will be exported to a file in the same directory as the catalog but will not be opened automatically. The exported file can be used to e-mail or share with other members of your team for review purposes and will not be removed automatically.

Notes:

- The function of previewing a report tab in Page Report Result format is enabled only if you have checked the option Server for Previewing Reports when installing JReport Designer.
- The preview server is a web ad hoc disabled server, and it supports only one user and one report concurrently. Therefore, when you close the report you are previewing, you should use the exit button of Page Report Studio instead of the browser's. In this way, you can release the user session you occupy at the same time, so that you are able to previewing another report later.
- JReport Designer and the preview server are running in two separated JVMs. The preview server applies `-Djreport.url.encoding=UTF-8` in the startup batch file and this setting covers URL encoding/decoding. Therefore, when you preview a report tab in Page Report Result format, if the name of the report tab or its report or catalog file contains non-Latin characters, you need add `-Djreport.url.encoding=UTF-8` into the startup batch file of JReport Designer as well; otherwise the preview will fail.
- When previewing a report tab in Page Report Result format,
  - Make sure the JDBC driver's class path has been appended to setenv.bat in `<install_root>\server\bin`.
  - If the report tab contains some user defined objects (UDO), to have them work correctly in the preview server, you need to compile your Java files again using report.jar and JREngine.jar in the `<install_root>\server\lib` directory, modify the udo.ini file in `<install_root>\server\lib` and append the classes to the class path of JRServer.bat in `<install_root>\server\bin`. For details, see Creating a UDO manually.
  - If the report tab contains some user defined data sources (UDS) or user defined formulas (UDF), you can use the classes you have compiled for using in Designer, but remember to append the classes to the class path of JRServer.bat in `<install_root>\server\bin`.

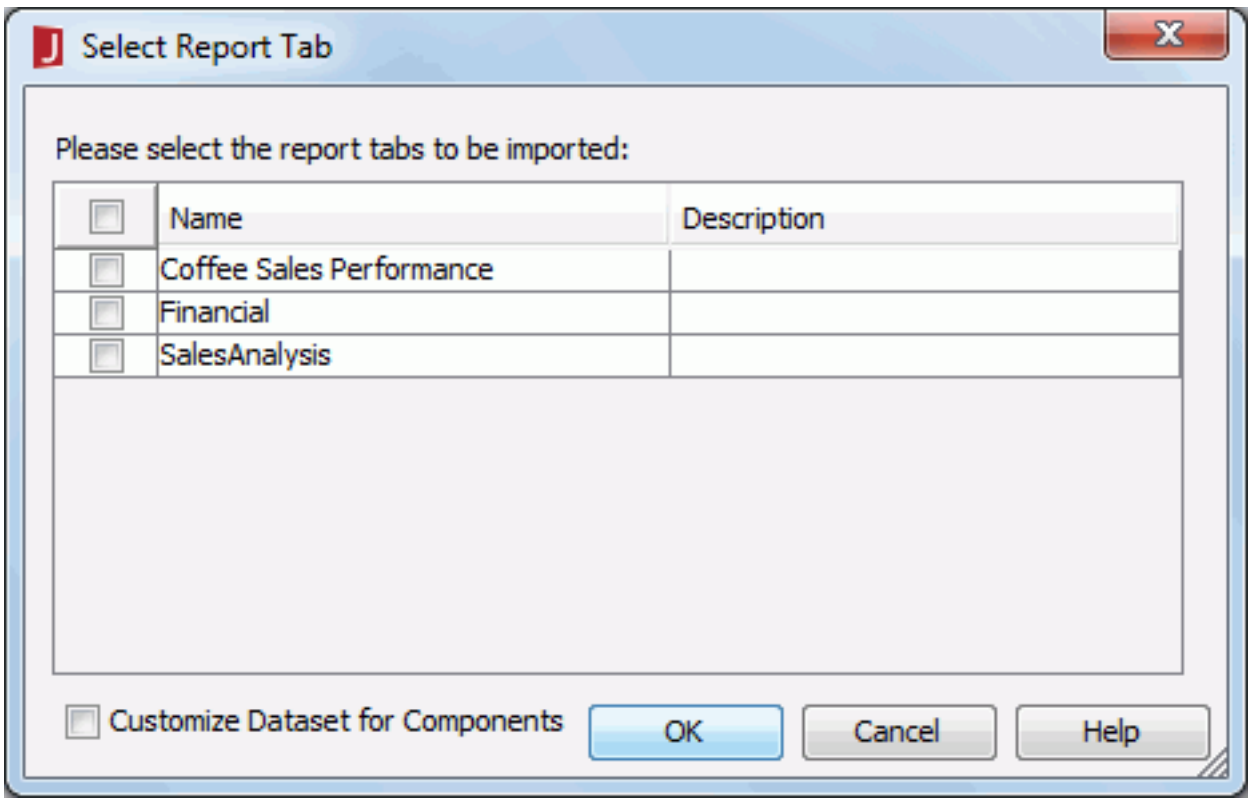
Importing report tabs into a page report

JReport enables you to import report tabs to a page report from another page report in the same catalog, and specify dataset for the imported components according to your requirements.

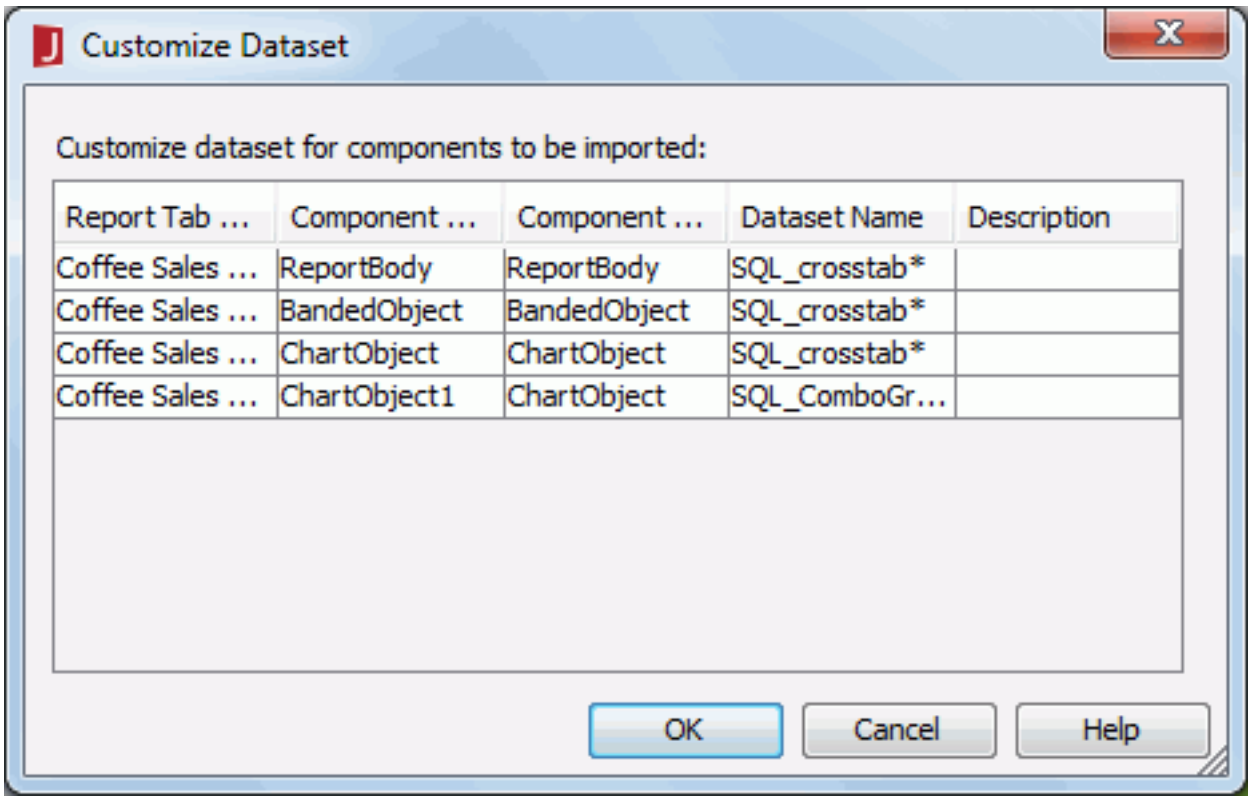
To import report tabs from another page report:

1. Click **File** > **Import From** > **JReport Page Report**.
2. In the Open Report dialog, specify the page report that contains the report tabs you want to import, then click **Open**.

   The Select Report Tab dialog appears.



3. Select the report tabs that you want to import.
4. If you want to customize datasets used by components within the to be imported report tabs, check the **Customize Dataset for Components** option. Then click **OK** and the Customize Dataset dialog appears.



5. All components in the selected report tabs, which have self-bound datasets, will be listed in the dialog. Specify which dataset you want the component to use by clicking its Dataset Name cell and choosing from the drop-down list.
6. Click **OK** to import the report tabs with specified datasets to their components. The imported report tabs will be listed after the existing report tabs in the page report.

Managing report tabs in a page report

This section presents to you how to manage report tabs in a page report with the report tab bar as follows:

- **Inserting a new report tab into a page report**
  1. On the report tab bar, right-click any report tab and select **Insert** from the shortcut menu.
  2. In the New Page Report Tab dialog, specify the layout of the new report tab, then click **OK**.
  3. Create the report tab according to your requirements.
- **Copying a report tab**

  To copy a report tab, on the report tab bar, right-click the report tab you want to copy and click **Copy** on the shortcut menu. Then,
  - If you want to make a copy of the report tab within the current report, right-click any report tab and select **Paste** from the shortcut menu. You will then be prompt to give the report tab a new name. Specify the name as required.
  - If you want to copy the report tab to another page report of the same catalog, open the report to which the report tab will be copied, then right-click any report tab and select **Paste** from the shortcut menu.
- **Moving a report tab to another page report**
  1. On the report tab bar, right-click the report tab you want to move and click **Cut** on the shortcut menu.
  2. Open the page report where the report tab will be moved.
  3. Right-click any report tab and click **Paste** on the shortcut menu.
- **Renaming a report tab**
  1. On the report tab bar, right-click the report tab with the name you want to change and click **Rename** on the shortcut menu.
  2. In the Input Report Name dialog, enter a new name for the report tab.
  3. Click **OK** to confirm the name.
- **Duplicating a report tab**

  To make a duplicate report tab in the current page report, right-click the report tab on the report tab bar, and then click **Duplicate** on the shortcut menu.
- **Removing a report tab**

  To remove a report tab from a page report, right-click the report tab on the report tab bar, and then click **Remove** on the shortcut menu.

  > **Note:** Since a page report cannot be empty, you will not be able to remove all of the report tabs in the page report.

Making high-efficiency reports

The following ways are provided for you to make reports efficient, reliable, and fast at runtime.

- Suppress all useless panels in a banded object, such as Group Header and Footer panels which are not adding values to the report tab.

  To do this, select the panel, go to the Report Inspector, and then change the property Suppress to true.

  This is more efficient than hiding the panel.

- Set the precision (length) of DBFields or record level formulas to the size of the maximum value you need to display in the report tab. Make them larger than necessary wastes resources.

  To do this:

  1. Make sure the option "Forbid editing data object properties" is unchecked in the Catalog category of the Options dialog.
  2. Go to the Catalog Browser, expand the Properties sheet.
  3. Select the DBField or record level formula, then change the Precision property to an appropriate value.

- You can improve the report performance by setting the data buffer size to a larger value.

  By default, the size of data buffer is 2MB. When data exceeds the default size, data will be written to disk, which will slow down the report processing. JReport Designer enables you to set data buffer size by specifying two properties of a dataset or subreport in the Report Inspector: Records per Page and Max Page Number. For details, see Data Buffer.

- Adjust Result Buffer.

  The result buffer is used to store the report result in pages. Its default size is 4 which indicates 4 pages of the report result are allocated to the result buffer, other pages will be stored on disk. If you have enough memory, to get better performance, you can increase the result buffer size to store more pages of report result.

  To adjust the result buffer size, in the Report Inspector, highlight the node which represents the report tab, and then change the value of Result Buffer Size to the number of pages you want to hold in memory.

- Adjust Excel Buffer for exporting to Excel.

  The Excel buffer is used to store the report result in excel buffer sheets when exporting to Excel format. Its default size is 1, which indicates 1 sheet of the report result are allocated to the result buffer; other sheets will be stored on disk. If you have enough memory, you can increase the Excel buffer size to store more sheets of excel format report result in memory. For details, see Setting the Excel buffer size.

- When running a report to the Delimited Format (such as the CSV format) on JReport Server, the property Fast Pass can help you speed up the generation process. Setting Fast Pass to true will enable you to specify the position of the object you want to display in the result file, thus JReport will not calculate the position for every object during the generation process.

  To use this method:

  1. In the Report Inspector, select the node that represents the report tab and set the properties Fast Pass and Columned to **true**.
  2. For each DBField or label, set the exact coordinate using the properties Column Index and Row Index.

     For example, set the properties Column Index=0 and Row Index=2 for a DBField, JReport Designer will put this DBField into the first column and the third row in the generated Text file.
  3. Set the property **engine.single_thread=false** in the file server.propertise in `<server_install_root>\bin`. This enables the property Fast Pass to take effect when running the report to the Delimited Format.

- Cache Section.

  The Cache Section property is used to cache the objects in the detail panel of a banded object instead of creating them repeatedly. You can set Cache Section in the Report Inspector to true to improve the performance.

  > **Note:** Cache Section doesn't work if there is crosstab, text object, chart, and so on in the detail panel of the banded object, because these objects contain child-DBFields.

- Cache Value (only for DBFields).

  The Cache Value property is used to cache the layout attributes (font size, font face, width and so on) of DBFields when exporting the report to any format. To get better performance, set it to true in the Report Inspector.

- Push Down Group Query.

  The Push Down Group Query property controls whether to push down group level summary computations in reports to the DBMS at runtime. By pushing down the computations, the DBMS' computation capability can be made use of, and thus the reports' running efficiency will be improved.

  The property can be specified at three levels for page report: JDBC connection, query within the connection, and page report. The setting on page report has the highest priority. While, for web report and library component, the property can be specified only at the JDBC connection or the query within the connection. The setting on the query has higher priority.

Saving a page report

After you have created or modified a page report, you should then save it in order to be able to load it the next time.

By default, page report files contain a .cls suffix. This is the format that provides optimal performance in the JReport toolset. Page report files can also be stored in the following formats:

- **Report Template (.rpt)**
  It is a file saved in text format, which can be edited in any text editor.
- **Self Contained Page Report (.clx)**
  It is a binary report file, which contains not only the report's layout but also the catalog with its own resources.
- **Page Report XML Format (.cls.xml)**
  It follows the XML standard, which can be edited with both JReport Designer and external XML editors. Page reports that contain customized user defined objects (UDO) cannot be saved to XML format.

  > **Note**: The XML feature for page reports is under license control. To obtain the special key, contact Jinfonet Support (support@jinfonet.com).

Saving a page report in its current format

To save a page report in its current format, click **File** > **Save** on the menu bar, or the **Save** button on the Standard toolbar. If the page report to be saved is new, the Save Report As dialog will appear for you to input a name.

This operation can also be accomplished by pressing the Ctrl + S keys.

Saving a page report in a different name in the current directory

1. Click **File** > **Save As** on the menu bar.
2. In the Save Report As dialog, type a new file name in the Report Name field.
3. Select a report type from the Report of Type drop-down list.
4. Click **Save** to save the page report in the new name.

Saving a page report to a different catalog

Since the resources for a catalog file are shared by all reports based on it, if you have different developers develop reports at the same time, and you want to use the same catalog in order for all the resources to be reused, you can have them save their reports with their related resources to a "universal" catalog file. In this way, the to-be-saved report's resources such as queries, formulas, and parameters will be merged into the target "universal" catalog file.

To save a page report to a different catalog:

1. Click **File > Save To** on the menu bar.
2. Browse to the location of the target catalog, then click **Save**.
3. In the Warning message box, click **Yes**.

   This saves your report into a target catalog file, which means:

   - The report and the relevant files alike will be copied to the directory where the target catalog file is located.
   - The resources (query, formulas, parameters, and so on) that are referenced by this report in the current catalog will be merged to the target catalog.
   - If the connection does not exist yet in the target catalog, it will be added using the same connection information as in the source catalog.

In this way, even while users work with their own individual catalog files, they can still publish their reports and associated resources to the target "universal" catalog file. Note that JReport does not handle the concurrent issue. Two people cannot update the "universal" catalog at the same time.

See also Merging catalogs for more information about the Save To command.

•

Configuring page reports for specific Page Report Studio features

Page reports developed by JReport Designer can be published to a standalone JReport Server or to web applications integrated with JReport Server. After that they can be viewed and analyzed in Page Report Studio. Certain Page Report Studio features require a page report to be pre-configured in JReport Designer before being published to JReport Server.

Pick a task from the following:

- Obtaining detailed information from a banded object
- Customizing the field display names
- Predefining sort and filter conditions
- Setting filter options for a field

Note: In addition to the above features, JReport allows you to create a table of contents for Page Report Studio use by setting the TOC Anchor and Anchor Display Value properties for each component in a report, then in Page Report Studio, you can use the TOC Browser to locate the component. For details about how to show components in a TOC tree, see Editing a page report.

Obtaining detailed information from a banded object

In a banded object which contains group information, you may only care about the details in a certain group. To meet your expectation, JReport provides a function: go-to-detail.

Going to detail from a field/label/image

If a banded object contains group information, then a field (DBField, parameter, formula, summary, special field), label, or image in a group header/footer panel of the banded object, including the case that the field, label or image is in a tabular cell or text box which is in the banded object's group header/ panel panel, can be used to obtain information of that group. To do this:

1. In JReport Designer, open a page report which contains a grouped banded object.
2. Select the object which will be used for going to detail, set its property Go to Detail to **true** in the Report Inspector.
3. Find the property Detail Target Frame, then from its value drop-down list, select the target window or frame in which to display the detailed information.
4. Publish the page report to JReport Server and run it in Page Report Studio.
5. Right-click the object and select **Go to Detail**, then the related group information will be displayed.

Going to detail from a map

A map in a grouped banded object can also be used for going to detail with the following preconditions:

- The map inherits dataset from the banded object.
- The banded object has at least two groups.
- The map is inserted in the group header/footer panel that is not the lowest level group of the banded object.

To go to detail from a map:

1. In JReport Designer, open a page report which contains a grouped banded object with a map.
2. Select the map, then in the Report Inspector set its property Go to Detail to **true**, from the Detail Target Frame value drop-down list, select the target window or frame in which to display the detailed information.
3. Publish the page report to JReport Server and run it in Page Report Studio.
4. Right-click the map and select **Go to Detail**, then the related group information will be displayed.

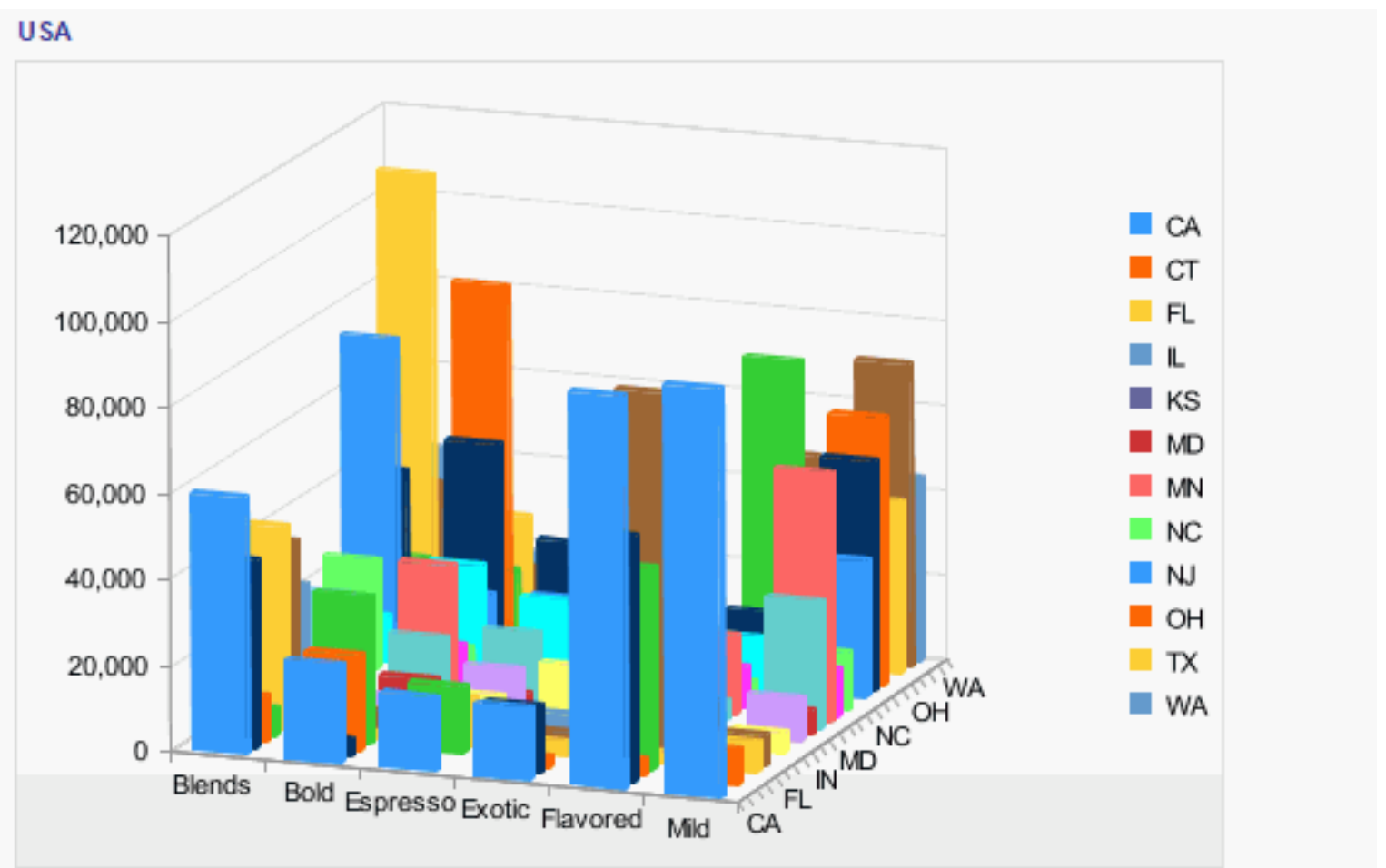
Going to detail from a chart

A chart in a banded object can also be used for going to detail, however, some preconditions must be met:

- The value axis of the chart shows summary information.
- The chart inherits dataset from the banded object, and has no data container link with the banded object.
- In the case that FieldA and FieldB are respectively displayed on the series and category axes of the chart, then the two fields must be group fields of the banded objects, and FieldA is one-level higher than FieldB, and the chart must be inserted into the header/footer panel of the group which is one-level higher than FieldA (if FieldA is the highest group level in the banded object, then the chart should be inserted into the banded header/footer panel). In the case that the chart only shows FieldB as category information (no series axis), then FieldB must be a group field of the banded object, and the chart must be inserted into the header/footer panel of the group which is one-level higher than FieldB, or the banded header/footer panel if FieldB is the highest level.

The following example shows how to obtain detailed information from a chart:

1. Create a standard banded report based on the query WorldWideSales in Data Source 1 of the catalog file SampleReports.cat. The standard banded report displays the following fields: the DBFields Product Name, Unit Price, Discount and Quantity and the formula Total, is grouped by Country, State and Category (Country is the highest level and Category the lowest), and applies the Classic style.
2. In the Resource View panel, click **<Add Summary...>** from the Summaries node and create a summary named Sum_Total, which uses the function Sum, sums on the formula Total, and applies to the Category group level, then drag the summary to the footer panel of the Category group level in the banded object.
3. Insert a Bar 3-D chart into the header panel of the Country group level. The chart inherits the dataset from the banded object (make sure to check the **Current** radio button in the Data screen of the Create Chart wizard), and displays Category on the category axis, State on the series axis and Sum_Total on the value axis.
4. In the Report Inspector, select the node that represents the chart from the resource tree and set its Go to Detail property to **true**.
5. Save the report and click **View** > **Preview As** > **Page Report Result** to preview it in Page Report Studio.
6. Turn to page around 25 to view records in the USA country.



CA				
Blends				
Gold Coast Blend	11.45	.00	105	1,202.25
Gold Coast Blend	11.45	.00	105	1,202.25
Gold Coast Blend	11.45	.00	105	1,202.25
Gold Coast Blend	11.45	.00	105	1,202.25
Gold Coast Blend	11.45	.00	105	1,202.25
Colombia El Tambo	12.00	.00	80	960.00
Colombia El Tambo	12.00	.00	80	960.00
Colombia El Tambo	12.00	.00	80	960.00
Colombia El Tambo	12.00	.00	80	960.00
Colombia El Tambo	12.00	.00	80	960.00
Blue Mountain	9.95	.00	104	1,034.80
Blue Mountain	9.95	.00	104	1,034.80
Blue Mountain	9.95	.00	104	1,034.80

7. Now point to the bar of State=CA and Category=Bold, the mouse pointer will become a hand. Click the bar and the result will be regenerated.

Bold				
Brazil Ipanema Bour	12.95	10.00	33	384.61
Brazil Ipanema Bour	12.95	10.00	33	384.61
Brazil Ipanema Bour	12.95	10.00	33	384.61
House Blend	9.95	20.00	529	4,210.84
House Blend	9.95	20.00	529	4,210.84
Breakfast Blend	11.15	20.00	282	2,515.44
Breakfast Blend	11.15	20.00	233	2,078.36
Breakfast Blend	11.15	20.00	233	2,078.36
Breakfast Blend	11.15	20.00	233	2,078.36
Breakfast Blend	11.15	20.00	233	2,078.36
Mexico Organic	13.45	20.00	144	1,549.44
Mexico Organic	13.45	20.00	144	1,549.44
			Sum_Total	**23,503.28**

8. If you click a label on the category axis of the chart, you will go to detailed information concerned with the product category represented by that label; if you click a series axis label, detailed information about the state indicated by the label will be displayed.

Notes:

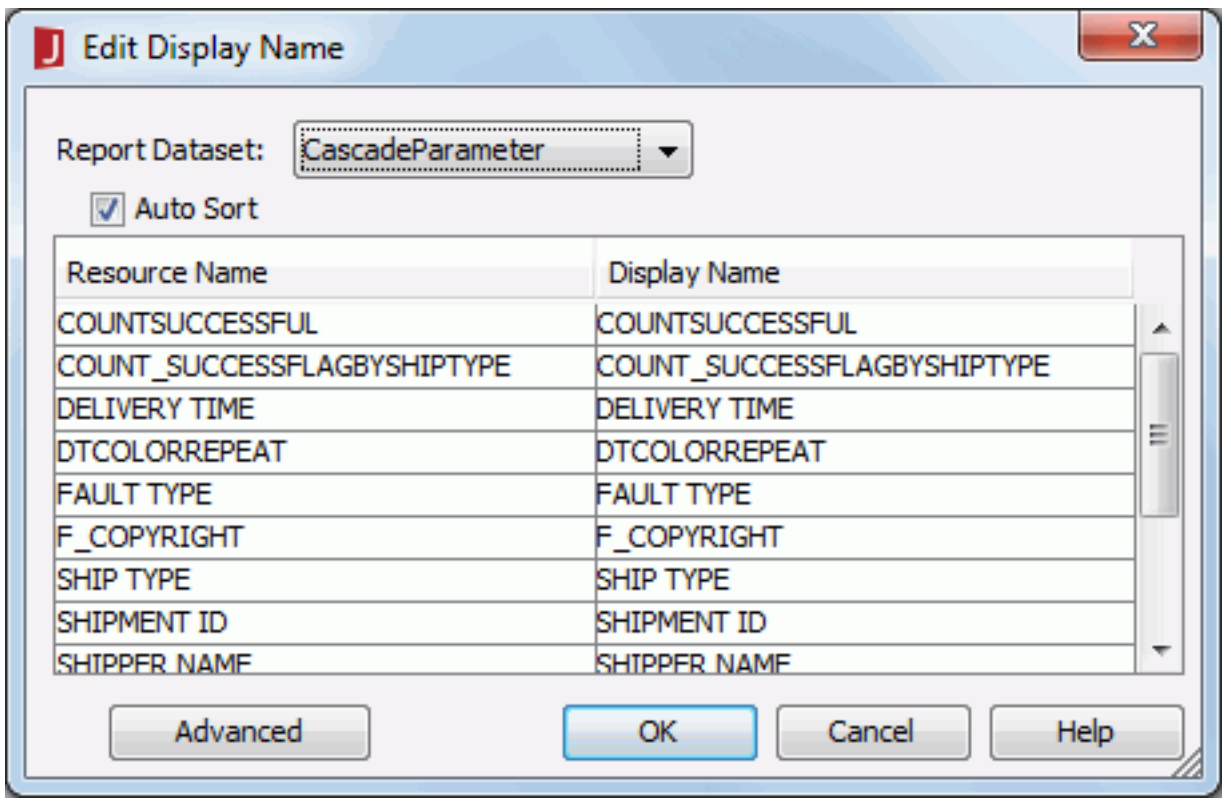
- The Go to Detail property is enabled by default for summaries on the group header/footer panel of a banded object. This is the same case when the formulas on the group header/footer panel of a banded object are group level formulas.
- Go-to-detail from a summary or group level formula in a banded object can also be performed in view mode in JReport Designer, and in HTML/PDF report outputs.
- For a chart, its Hyperlink, X Hyperlink, and Z Hyperlink properties allow you to set a URL for the graphic objects (such as bars, benches, and so on), category axis labels, and series axis labels respectively. If you have set any of the three properties, the corresponding object will no longer support the go-to-detail function in Page Report Studio, and clicking it will go to the URL. If Interactive is set to true and you have set Hyperlink, X Hyperlink, and/or Z Hyperlink, double-clicking the corresponding part of the chart will go to the URL you set.
- If there is no other actions bound on the object which is enabled with the go-to-detail action, you can also directly click the object to get the detail information at runtime. If there is other actions bound on the object, you can click the object to obtain the detail information only when the go-to-detail action is specified with the highest click priority.

Customizing the field display names

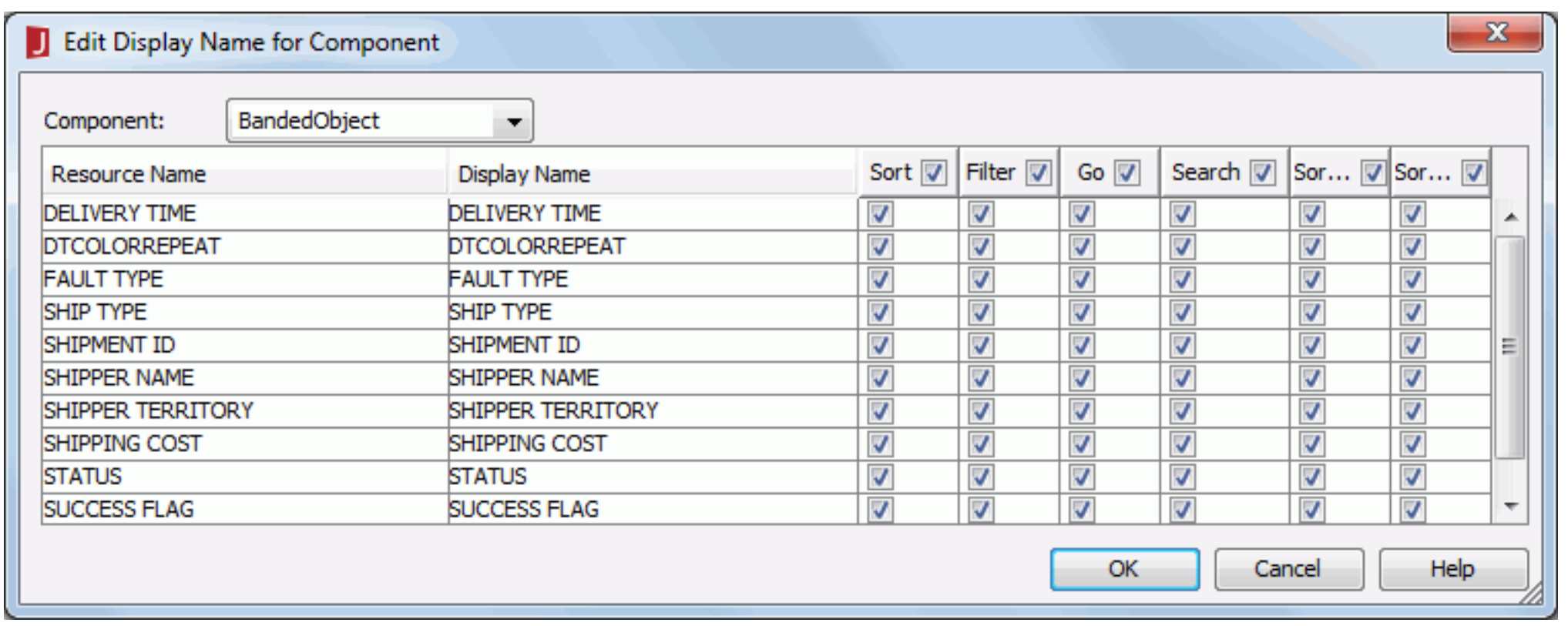
In Page Report Studio, you can drill around report data, sort report data on certain fields, search a report for some text, and filter the report data using filter conditions. When you perform these operations, you will be working with field mapping names. However, with just the field mapping names, you may find it inconvenient for end users, especially when the field mapping name is obscure. To help you, JReport provides a display name customizing function for you to define the column names as required, and also to specify the actions which the display names will take part in.

To customize the field display names, follow the steps below:

1. Open a page report in JReport Designer.
2. Click **Report** > **Edit Display Name** on the menu bar to display the Edit Display Name dialog.



3. Select a dataset from the Report Dataset drop-down list, and all the fields in the dataset will be shown in the mapping name box.
4. To make the resource names sort automatically, check the **Auto Sort** checkbox.
5. Specify the display names for the fields in the Display Name column. A formula can be selected as the display name of the field.
6. Select another dataset and repeat the above steps to edit the display names of fields in it.
7. If required, you can click the **Advanced** button to further customize the display names in specific component in the Edit Display Name for Component dialog.



8. From the Component drop-down list, select the component you want to customize.

   Tip: You can also right-click a component and then select Edit Display Name from the shortcut menu to display the Edit Display Name for Component dialog. However, if the dialog is opened in this way, only the component you right-click on will be listed in the Component drop-down list.

9. In the action columns, check the corresponding checkboxes to indicate whether the actions are enabled for the fields. Check the checkbox on the action column header if you want the corresponding action to be enabled for all fields. If any action is not supported on the selected component type, the corresponding column will be disabled.

   When an action is checked for a field, the field's display name instead of mapping name will be shown in the corresponding dialog or submenu in Page Report Studio. If you uncheck the box for any field in any action column, the field will not be available for the action. Moreover, if you set the display name of any field to be blank in the Edit Display Name dialog, all actions will be disabled for the field, which means end users will not be able to perform all these actions on the field in Page Report Studio.

10. Upon finishing, click **OK** to accept the changes.

Note: If you set the display name of any field to be blank, the field will not be shown in the lists where display names are displayed in Page Report Studio.

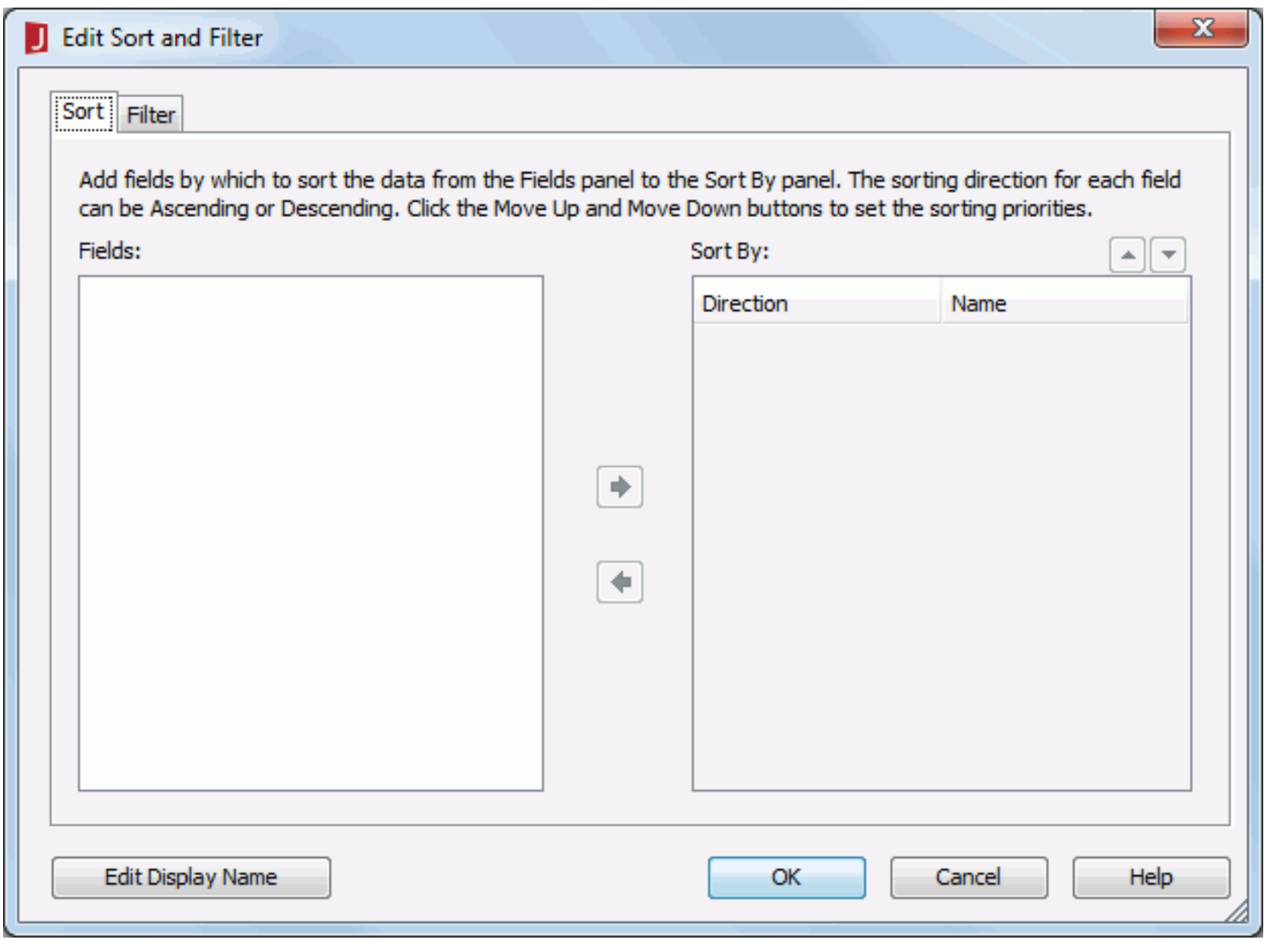
Predefining sort and filter conditions

You can predefine some sort and filter conditions for data components, which refer to banded objects, tables, charts and crosstabs, in page reports, so that when the page reports are published to JReport Server and are loaded in Page Report Studio, data in the components will be displayed in the order you prefer, or be narrowed down as you desire.

Predefining sort conditions

To predefine sort conditions for a data component, follow the steps below:

1. Select the component to which the sort conditions will apply.
2. Click **Report** > **Edit Sort and Filter** to show the Edit Sort and Filter dialog. The Sort tab is active by default.

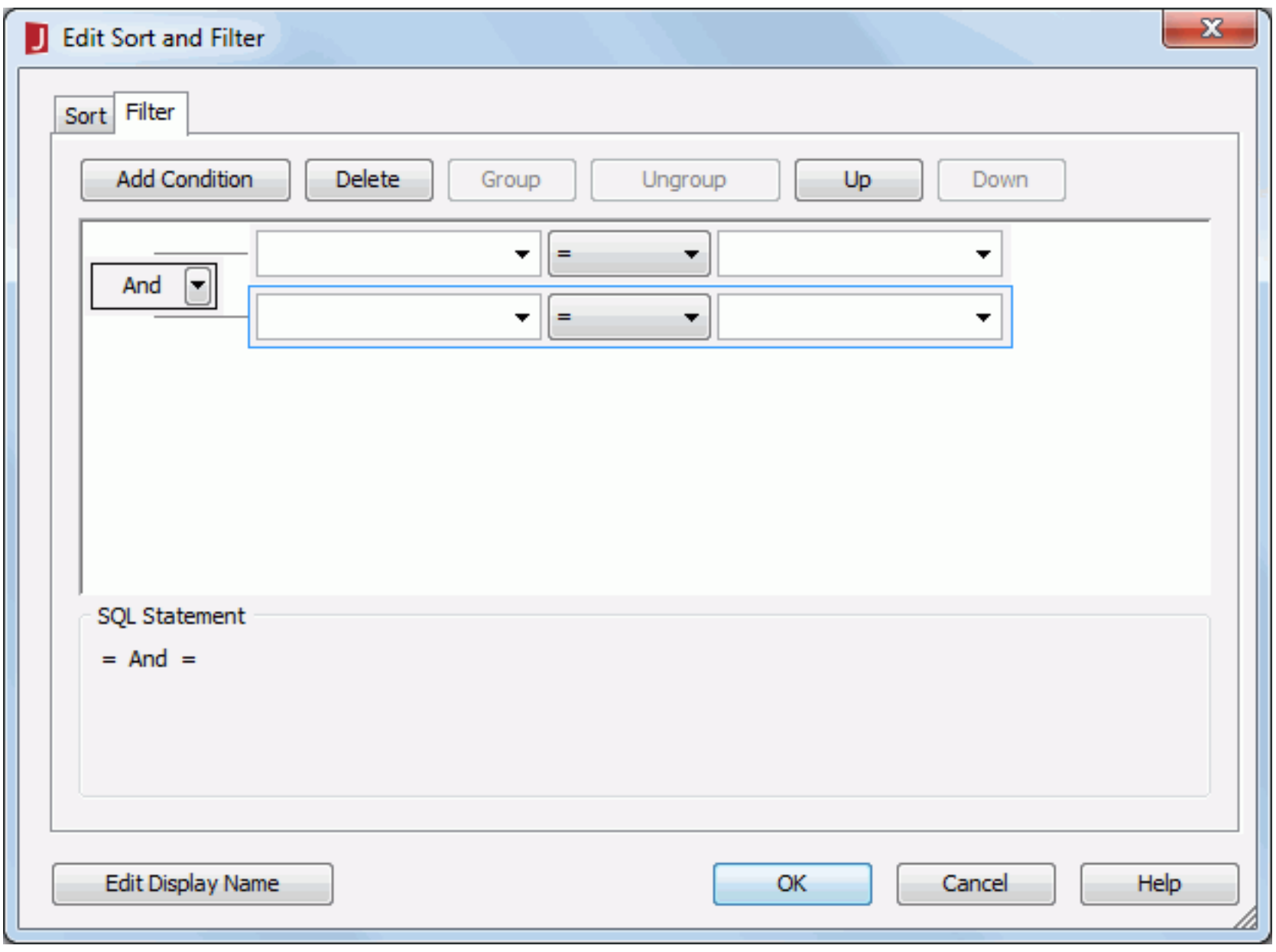


3. In the Fields box, all the fields in the dataset the component uses are displayed. Select the field by which to sort the data from the box and click to add it to the Sort By box. To sort by two or more fields, select all of them and then click .
4. To modify the sorting direction for a field, select (ascending) or (descending) in the Direction column.
5. To change the sorting priority of the fields, select the field in the Sort By box, and then click or to move it up or down.
6. Click **OK** to accept the settings.

Predefining filter conditions

To predefine filter conditions for a data component, follow the steps below:

1. Select the component to which the filter conditions will apply.
2. Click **Report** > **Edit Sort and Filter** to show the Edit Sort and Filter dialog, then switch to the **Filter** tab.



3. Click the **Add Condition** button to add a condition line.
4. From the field drop-down list, select the field on which the filter will be based.
5. From the operator drop-down list, set the operator with which to compose the filter expression.
6. In the value combo box, type the value of how to filter the field or select the value from the drop-down list.
7. Click **Add Condition** to define another condition line and set the relationship between the two lines if required.
8. When done, click **OK** to apply the settings.

Notes:

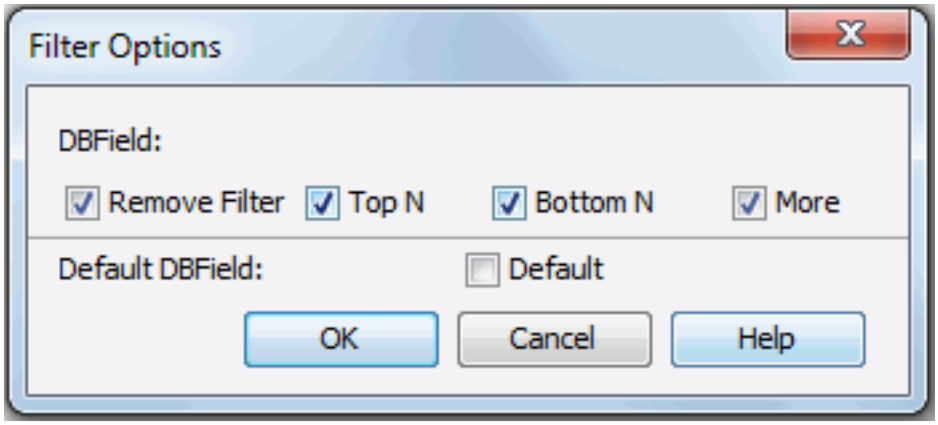
- If the Customize Display Name feature hasn't been applied to the dataset on which the component is created, when you open the Edit Sort and Filter dialog, there will be no fields displayed either in the Fields box of the Sort tab or the field drop-down list in the Filter tab, in which case, you need to click the Edit Display Name button in the dialog first to enable the feature on the dataset. You can also modify the display names in the Edit Display Name dialog if desired.
- The predefined sort and filter conditions will also affect the preview result in JReport Designer, and the result produced from being run in other formats besides Page Report Studio.

•html>

Setting filter options for a field

In Page Report Studio, when you right-click a field (including DBField, formula, summary and label) you will find the Filter submenu. This submenu can list items such as Remove Filter, Top N, Bottom N, and More. With these menu items you can easily filter the records. If you do not want to show some of the four items for a field, you can set the filter options in JReport Designer as follows.

1. Select the field and locate its Filter Options property in the Report Inspector.
2. Click ... to bring out the Filter Options dialog.



3. Check the items you want to show on the Filter submenu of the field in Page Report Studio, and then click **OK**.
4. The properties value will be generated, equal to sum of values of those selected items. The items and corresponding values when checked are as follows:

   Remove Filter: 1
   Top N: 2
   Bottom N: 4
   More: 8
   Default: 16

5. After saving the page report and publishing it to JReport Server, you will see the effect.

Note: In the Filter Options dialog, only when Default is deselected will the remaining four checkboxes be enabled. If you check Default, they will be disabled, but their values will still affect the Filter Options property value. If Default is checked (Filter Options >= 16), which items will be displayed on the Filter submenu in Page Report Studio will be determined by settings in the Profile dialog in JReport Server (Profile > Customize Profile > Properties > Default Properties > Filter Menu).

Page Report Manager

All created page report files (.cls) are physically placed in the same directory where the catalog file resides. Due to this, you will not be able to get detailed information about a page report until it is opened in JReport Designer. For example, if you want to find a page report with a special condition, you will have to load each page report. For just a few page reports, this is not a problem. However, the case is different when there are many reports.

JReport provides you with a tool for managing page reports by means of adding them into folders with the Page Report Manager (the Reports tab of the Catalog Browser). You can build folder items in the manager, helping you to divide the page reports into different types at your disposal. Then, the reports can be added to the folder, and worked on just like processing files in a file system--add, open, delete, move, rename and search items. In all, the feature of report folder implemented in catalogs enables you to manage your page reports in a way that can better satisfy your requirements.

The following are some concepts used in the Page Report Manager:

- **Folder Item**: A container in which you can operate page report items or sub folder items.
- **Report Item**: This represents a page report. You can edit name, keywords and description properties of this item.

In the Page Report Manager, you can manage your page reports as follows:

Adding an item

To add a folder item to the Page Report Manager:

1. In the Reports tab of the Catalog Browser, select a folder and right-click it.
2. Select **Add Folder** from the shortcut menu.

Now, a new folder will have been added as a sub node, and the default name for the folder is Folder1, Folder2.... However, you can rename the folder as required. To do this, right-click the folder item and select **Rename** from the shortcut menu, then type a new name in the name box, and then click outside the box (or just press **Enter** on the keyboard).

To add a report item to the Page Report Manager:

1. In the Reports tab of the Catalog Browser, right-click the folder to which you want to add the page report.
2. On the shortcut menu, click **Add Page Report**.
3. In the Choose Page Report dialog, select a page report. Here you can select multiple reports at a time. Then, click **Open**.

Now, the selected reports have been added to the specified folder. You can double-click any report item in the Page Report Manager to open the report. The default name of the report item is the file name of the report. However, you can give the item another name as you would with a folder item.

A report can be added to multiple folders in case you want to include it in multiple categories.

Editing the properties of an item

Before you can edit the properties of any item, you should first make sure the "Forbid editing data object properties" option in the Catalog category of the Options dialog is unchecked (to access the Options dialog, click **File** > **Options** on the Designer menu bar, or the **Options** button on the Catalog Browser toolbar).

To edit the properties of an item:

1. Click the **Expand** button on the Catalog Browser toolbar, or right-click any item in the Reports tab and click **Properties** on the shortcut menu to display the Properties sheet.
2. Select the item the properties of which you want to edit, then edit its property values as required. After editing a value, press **Enter** on the keyboard to confirm the change.

The properties of a folder item are:

Property Name	Description
Description	Specifies the description of the folder. Data type: String
Name	Specifies the name of the folder. Data type: String

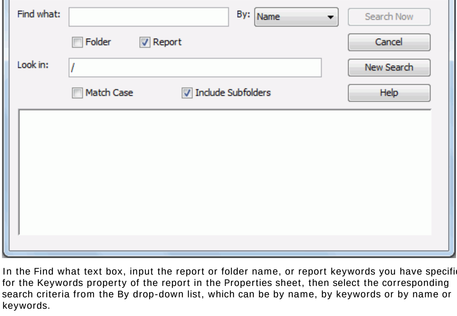
The properties of a report item are:

Property Name	Description
Author	Specifies the author of the report. Data type: String
Description	Specifies the description of the report. Data type: String
File Name	Specifies the name of the report. Data type: String
Keywords	Specifies the keywords of the report. Data type: String
Last Modified Time	Specifies the last time at which the report was modified. Data type: String
Last Print Time	Specifies the last time at which the report was printed. Data type: String
Last Run Time	Specifies the last time at which the report ran. Data type: String
Name	Specifies the display name of the report. Data type: String
Query Name	Specifies the query used by the report. Data type: String
Value Type	Specifies the type of the report. Data type: String

Searching for a report or folder item

Wildcard searches for report name or folder name in the Page Report Manager are supported. To search a report or folder item, follow the steps below:

1. In the Reports tab of the Catalog Browser, select a folder item and right-click it.
2. Click **Search** on the shortcut menu to display the Search Reports dialog.



3. In the Find what text box, input the report or folder name, or report keywords you have specified for the Keywords property of the report in the Properties sheet, then select the corresponding search criteria from the By drop-down list, which can be by name, by keywords or by name or keywords.

   If you just input part of the report or folder name, or report keywords, use * to replace the other part of the name or keywords. For example, if you want to search for the report EmployeeInformation, you should input *Info* instead of *info* in the text box.
4. Specify what you want to find: a folder, a report or both.
5. Enter the search path in the Look in text box.
6. Specify whether to limit the search to text that exactly matches the case of the search text.
7. Specify whether to find the folder or report also in the subfolders of the search path.
8. Click the **Search Now** button to start searching. Matched results will be listed in the search result box.
9. If you want to start another search, click the **New Search** button and set the condition using the method described above.

Exporting the report information to a table

You can export report information in the Page Report Manager to a table in a DBMS. The default name of the table is ReportsInfo. You can change the name as required. Each row of the table is a report description. To do this:

1. In the Reports tab of the Catalog Browser, select the root folder Reports and right-click it.
2. On the shortcut menu, click **Export**. The Get JDBC Connection Information dialog appears.
3. Fill in the required fields for connecting to the DBMS, and then click the **OK** button.
4. Give a name for the table, and then click **OK** in the Input dialog. The report information is added to the table in the DBMS.

You can then set up a JDBC connection in a catalog to connect with the DBMS and add the table to the catalog via the JDBC connection for use.

Note: If a table already exists, it will be replaced. So you must be sure that there are no duplicate names in the DBMS.

Moving an item to another folder

Any items added to the Page Report Manager can be moved to another folder if required. To do this:

1. In the Reports tab of the Catalog Browser, select the item that you want to move and right-click it.
2. Click **Move to** on the shortcut menu.
3. The Move Reports dialog will then appear. All the available folders will be listed in the dialog.
4. Select the folder you want the item to be moved to, and click **OK**. The item will then be moved to that folder.

Deleting items

By using the Catalog Browser, you can delete a report from the file system.

To delete a report or a folder or multiple items:

1. In the Reports tab of the Catalog Browser, select the report or the folder or the multiple items on the Reports tree, and then take either of the following ways:
   - Press the **Delete** key on the keyboard.
   - Click the **Delete** button on the toolbar of the Catalog Browser.
   - Right-click and select **Delete** from the shortcut menu.
2. A Warning message box is displayed. Check the option **Also delete report files from file system** if you want to remove them from your local directory permanently. Otherwise leave the option unchecked and only remove the items from the virtual tree.
3. Click **Yes** in the box. If a report is selected, the report will be deleted. If a folder is selected, the folder and all its contents will be removed.

If a report is deleted manually from the local file system, later when you open the Catalog Browser, you will see the node icon that represents the report is marked with a red cross. You can use the above method to remove the unused node.

Web Reports

JReport provides the web reporting solution which is aimed at easier and faster report creation and design, faster report execution, easier customization, and better presentation style using a newer Rich Internet Application (RIA) Web 2.0 interface. A web report contains only one report and uses .wis as the file suffix.

The data sources that can be used to create web reports are business views that are resources built on top of queries. In JReport Designer, you can create web reports based on business views, and publish them to JReport Server. JReport Designer supports opening, creating, and editing of web reports.

In JReport Server, end users can create web reports via the Web Report Wizard using a browser and open and edit web reports via Web Report Studio. Web reports created in JReport Server can also be downloaded to JReport Designer for editing.

Pick a task from the following:

- Components supported in web reports
- Creating a web report
- Editing a web report
- Using dynamic resources in a web report
- Using local parameters in a web report
- Using formulas to control showing or hiding components
- Previewing a web report
- Saving a web report
- Opening a web report

Notes:

- A JReport Live license for JReport Designer is required in order to create web reports and use all the related features. If you do not have the license, contact your Jinfonet Software account manager to obtain it.
- The resources, components, properties and options that are not supported by web reports are not enabled on the user interface.

Components supported in web reports

Components are the objects that you can place in a report. JReport provides a full set of components that allow you to present and control the report data and presentation in a wide variety of ways.

Web reports support the following report components:

- Tables
- Crosstabs
- 2-D charts
- Tabular
- Labels
- Images
- DBFields
- Formulas
- Summaries
- Parameters
- Special fields

  You can insert all special fields into web reports in JReport Designer. However, due to the characteristics of Web Report Studio, only these can be rendered and edited in Web Report Studio: User Name, Modified Date, Modified Time, Fetch Date, Fetch Time, Print Date, and Print Time.
- Multimedia objects
- Four web controls: parameter control, parameter form control, filter control, and navigation control.

Creating a web report

To create a web report:

1. Click **File** > **New** > **Web Report** on the menu bar.
2. A blank web report with a tabular of one cell is displayed. You can then split the tabular, insert components to the tabular cells, and modify the component properties if required. For details, see the corresponding topics about the specific component in the Components chapter.

Editing a web report

After a web report is built up, you can further modify it the same as you do with page report. For details, see Editing a page report.

Using dynamic resources in a web report

When working with a web report, sometimes you may find that the view elements that have been predefined in the business view the web report uses cannot meet your requirements, in which case, you can create some dynamic resources and use them in the report to get the desired data. Then when you save the web report, the dynamic resources will be saved along with it as its resources.

Dynamic resources are report level resources, which means they are only available to the report for which they are created.

Dynamic resources that can be used in web reports include formulas and aggregations, and their usage in web reports is the same as that in library components. For details about how to use them, refer to Using dynamic resources in a library component.

Using local parameters in a web report

When working with a web report, sometimes you may find that the predefined parameters in the catalog cannot meet your requirements, in which case, you can create some local parameters and use them in the report to get the desired data. Then when you save the web report, the local parameters will be saved along with it as its resources.

Local parameters are report level resources, which means they are only available to the report for which they are created. The usage in web reports is the same as that in library components. For details about how to use them, refer to Using local parameters in a library component.

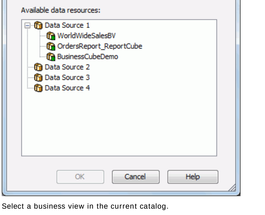
Using formulas to control showing or hiding components

The Invisible property of the objects in a web report can be controlled by a formula. For details about how to use formulas to control object properties, click here.

However, for any object whose parent doesn't have a data source, for example, a label in the tabular cell of a web report, if you want to use a formula to control its Invisible property, you need to first bind a business view to the web report, then you can create formulas of Boolean type based on this business view to control the property.

To bind a data source to a web report:

1. Click **Edit** > **Bind Data** on the menu bar. The Bind Data dialog appears.



2. Select a business view in the current catalog.
3. Click **OK** to bind the business view to the web report.

After binding the data source, you can now create formulas in this business view to control the Invisible property of objects in the web report, whose parents don't have data source. To do this:

1. Select the object and find its Invisible property in the Report Inspector.
2. Click in the value cell of the Invisible property and select the **<Create Formula...>** item from the drop-down list.
3. In the Input Formula Name dialog, specify the name of the formula and click **OK**. The Formula Editor appears.
4. Create the formula according to your requirement. You need to specify whether the formula is used as Group, Detail, or Aggregation by setting the Use As option on the toolbar.
5. Save the formula and exit the Formula Editor.
6. The formula name is now displayed in the value cell of the Invisible property.

   Then at runtime, whether the object is shown or hidden will be determined by the return value of the formula. A return value of true will hide the component.

The formula you created on the bound data source will be regarded as a dynamic formula in the selected business view, and you can find it in the Dynamic Resources > Formulas node of the Resource View panel. So there is another way to create formulas in the business view and use it directly when editing the Invisible property. To do this:

1. Click any blank area in the report body. The business view bound to the web report will be displayed in the Resource View panel.
2. Expand the **Dynamic Resources** > **Formulas** node, click **<Add Formula•?gt;**, then create the formula as required.
3. Edit the Invisible property of the desired object, click in the value cell and you will find the formula is listed in the drop-down list.

Previewing a web report

To preview a web report, do either of the following:

- Click **View** > **Preview As** > **Web Report Result** on the menu bar.
- Click the **Preview As** button on the Standard toolbar and then select **Web Report Result**.

The web report will then be opened in Web Report Studio. You can also click the View tab directly to preview the web report in JReport format, or select other format from the Preview As submenu to display it in that format.

Moreover, when previewing a web report, if it is developed with National Language Support, you can specify the language in which you want the report to display; if it contains parameters, you will be prompted to specify values for the parameters in the Enter Parameter Values dialog.

Notes:

- The function of previewing a web report in Web Report Studio also requires that:
  - You have checked the option Server for Previewing Reports when installing JReport Designer.
  - You have a JReport Live license for your JReport Server.
- The preview server is a web ad hoc disabled server, and it supports only one user and one report concurrently. Therefore, when you close the report you are previewing, you should use the exit button of Web Report Studio instead of the browser's. In this way, you can release the user session you occupy at the same time, so that you are able to previewing another report later.
- JReport Designer and the preview server are running in two separate JVMs. The preview server applies `-Djreport.url.encoding=UTF-8` in the startup batch file and this setting covers URL encoding/decoding. Therefore, when you preview a web report in Web Report Studio, if the name of the report or catalog file contains non-Latin characters, you need to add `-Djreport.url.encoding=UTF-8` into the startup batch file of JReport Designer as well; otherwise the preview will fail.
- When previewing a web report in Web Report Studio,
  - Make sure the JDBC driver's class path has been appended to setenv.bat in `<install_root>\server\bin`.
  - If the report contains some user defined data sources (UDS), you can use the classes you have compiled for using in Designer, but remember to append the classes to the class path of JRServer. bat in `<install_root>\server\bin`.

Saving a web report

To save a web report, click **File** > **Save** on the menu bar, or the **Save** button on the Standard toolbar. This operation can also be accomplished by pressing the Ctrl + S keys. If the web report to be saved is new, the Save Report As dialog will appear for you to input a name.

To save a web report with a different name in the current directory:

1. Click **File** > **Save As** on the menu bar.
2. In the Save Report As dialog, type a new file name in the Report Name field.
3. The web report file type Web Report (*.wis) is the only default value in the Report of Type drop-down list.
4. Click **Save** to save the web report with the new name.

To save a web report to a different catalog, refer to Saving a page report to a different catalog.

Opening a web report

To open a web report:

1. Click **File > Open** on the menu bar, or the **Open** button on the Standard toolbar.
2. In the Open Report dialog, browse to find the catalog that contains the web report you want to open.
3. In the Reports tab, select **Web Report (*.wis)** from the Report of Type drop-down list.
4. All the web reports in the current catalog will be displayed. Select the desired web reports.

   If you want to open web reports that you have opened recently, switch to the **Recent** tab, and then select the web reports you want to open.
5. Click **Open** to open the web reports that you specified.

Tip: If you just want to access some recently open web reports, click **File** > **Open Recent** and then select the web reports one by one from the submenu.

Library Components

Library components are used by end users to build dashboards using JDashboard. JDashboard is a separately licensed product in JReport Server. Library components are created and edited in JReport Designer, and then are published along with their catalog to the component library in JReport Server for use in dashboards.

A library component can contain multiple data components and uses .lc as the file suffix.

The data sources that can be used to create library components are business views that are resources built on top of queries, imported SQLs, stored procedures and all other data sources.

Each library component can be equipped with a configuration panel which is used to specify parameter values for the library component, filter or sort on the library component, or change properties of the library component.

Messages can be delivered between library components, which makes the communication between the library components more flexible.

This chapter introduces what you can do with library components in JReport Designer:

- Library component view
- Components supported in library components
- Creating a library component
- Using dynamic resources in a library component
- Using local parameters in a library component
- Using the configuration panel
- Delivering messages between library components
- Previewing a library component
- Opening a library component
- Saving a library component

Note: A JDashboard license for JReport Designer is required in order to create library components and use all the related features. If you do not have the license, contact your Jinfonet Software account manager to obtain it.

Library component view

A library component contains three parts: wrapper, configuration and content.

The wrapper defines some basic properties of the library component, such as the properties of the title and the CSS class that will be applied to the title bar. In addition, it also defines the auto-refresh rules, such as whether to enable auto refresh and the type of auto refresh. You can select the wrapper by clicking the Wrapper node in the Report Inspector.

The configuration panel shows the title bar and the panel is used to set parameter values for the library component, filter or sort on the library component, or change properties of the library component in the content. You can display the configuration panel by checking the Display Configuration Panel checkbox on the top-right corner.

The content contains a body and a page panel which has no page header or page footer. The body is a container containing a number of components such as charts, crosstabs, tables, geographic maps, labels, web controls and so on. The page panel is used to print or export the library component. You can bind data source to the content and create formulas based on this data souce to control the Invisible property of objects in the library component, whose parent does not have data source. For details, see Using formulas to control showing or hiding components.

The content supports auto fit only when the library component is published into the component library in JReport Server and is used in JDashboard. The style of the title bar, configuration panel and content borders are controlled by the JDashboard theme and the library component view in JDashboard is of the same style, which can only be edited in the CSS file instead of being edited individually. While, the body in the content and the components in the body, such as tables, charts, labels and so on can have their own formats and their properties that can be edited individually.

Components supported in library components

Components are the objects that you can place in a library component. JReport provides a full set of components that allow you to present and control the component data and presentation in a wide variety of ways.

Library components support the following components:

- Tables
- Crosstabs
- 2-D charts
- Labels
- Images
- DBFields
- Summaries
- Formulas
- Geographic maps
- Web controls: parameter control, parameter form control, filter control, navigation control, slider, text field, checkbox, drop-down list, and list.

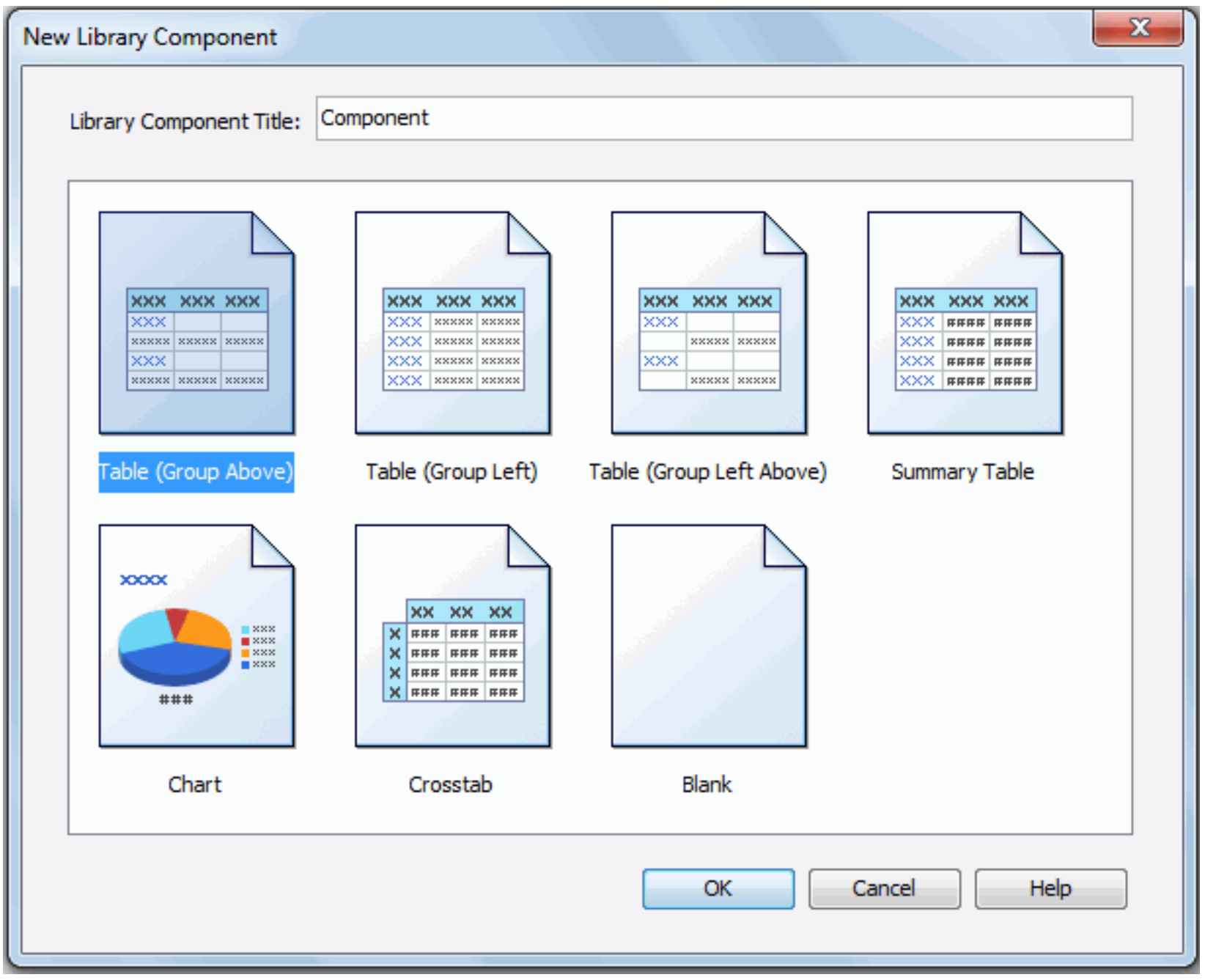
Creating a library component

There are two ways to create a library component: as a new library component or save a data component in a web report as a library component.

Creating as a new component

To create a new library component:

1. Click **File** > **New** > **Library Component** on the menu bar to display the New Library Component dialog.



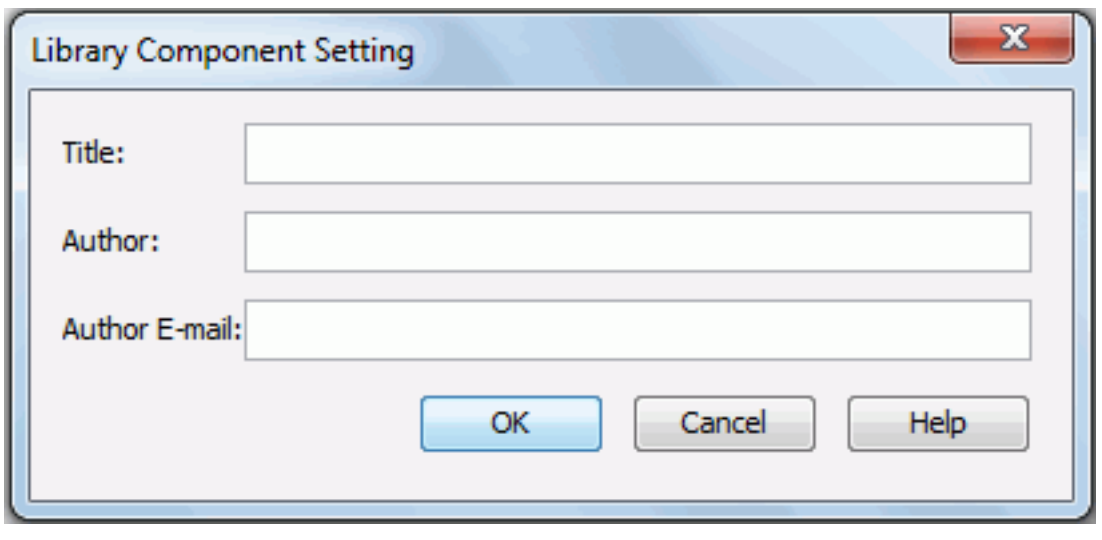
2. Input a title for the component in the Library Component Title text box and select a type for the component in the component type box.
3. Follow the wizard to create the library component with the first data component in it. For the usage of the wizard, refer to Creating report tabs in a page report.
4. Insert other components, such as charts, crosstabs, tables, geographic maps, labels, web controls and so on into the library component if needed. For details, see the corresponding topics about the specific component in the Components chapter.
5. If required, equip the library component with a configuration panel, which can be used to specify parameter values when there are parameters used in the library component, to filter or sort on the library component, or to change properties of the library component.

Creating by saving a data component in a web report

The data components, including tables, charts, crosstabs, and geographic maps in a web report can be saved as a library component, provided that all the objects in the data components can be supported by library components.

To save a data component in a web report as a library component:

1. Open the web report which contains the data component you want to save as a library component.
2. Right-click the data component and select **Save as Library Component** from the shortcut menu.
3. In the Save As dialog, specify the name for the library component in the File Name text box.
4. The library component file type (*.lc) is the only default value in the Files of Type drop-down list.
5. Click **Save**. The Library Component Setting dialog appears.



6. Specify the title, author and the author's e-mail address in the Title text box, Author text box and Author Email text box respectively.
7. Click **OK** to save the selected data component as a library component with the specified name and title.

Note: The resources, components, properties and options that are not supported by library components are not enabled on the user interface.

Using dynamic resources in a library component

When working with a library component, sometimes you may find the predefined view elements in the business view on which the library component is based cannot meet your requirements. In this case, you can create some dynamic resources and use them in the library component to get the desired data. Then when you save the library component, the dynamic resources will be saved along with it as its resources.

Dynamic resources are library component level resources, which means they are only available to the library component for which they are created.

Dynamic resources that can be used in library components include formulas and aggregations.

Creating and using dynamic formulas

To create a dynamic formula:

1. In the Resource View panel of the main window, or in the Resources box of the report wizard, expand the **Dynamic Resources** > **Formulas** node, click **<Add Formula•gt;**.
2. In the Input Formula Name dialog, specify the name of the formula and click **OK**. The Formula Editor appears.
3. Create the formula according to your requirement.
4. From the Use As drop-down list on the toolbar, specify whether the formula is used as Group, Detail or Aggregation.

   Whether a dynamic formula can be used as a certain type depends on the following rule:

   - Any formula can be used as Detail.
   - Record level pass one formula can be used as Group.
   - Comparable and record level pass one formula can be used as Group.
   - Group level formula can be used as Aggregation.
5. Save the formula and exit the Formula Editor.

Once a dynamic formula has been created, you can then drag it from the Resource View panel to the desired position in the library component as a detail/group or an aggregation object as defined, or use it when working with the report wizard.

In the Resource View panel, you can make use of the shortcut menu of an existing dynamic formula to edit or delete the formula, or change the formula type as Group, Detail, or Aggregation if needed.

Notes:

- You can only use JDK (not JRE) to compile formulas created in JReport and save a dynamic formula with no errors into a library component.
- Currently, global variables are not supported in dynamic formulas.
- When dynamic formulas reference display names or mapping names, the names should not contain any of the following characters if the names are not quoted by double-quotation marks "":

  "~", "`", "!", "@", "#", "$", "%", "^", "&", "*", "(", ")", "-", "+", "=", "{", "}", "[", "]", "|", "\\", ":", ";", "\", " ' ", "<", ",", ">", ".", "?", "/"

  Examples:

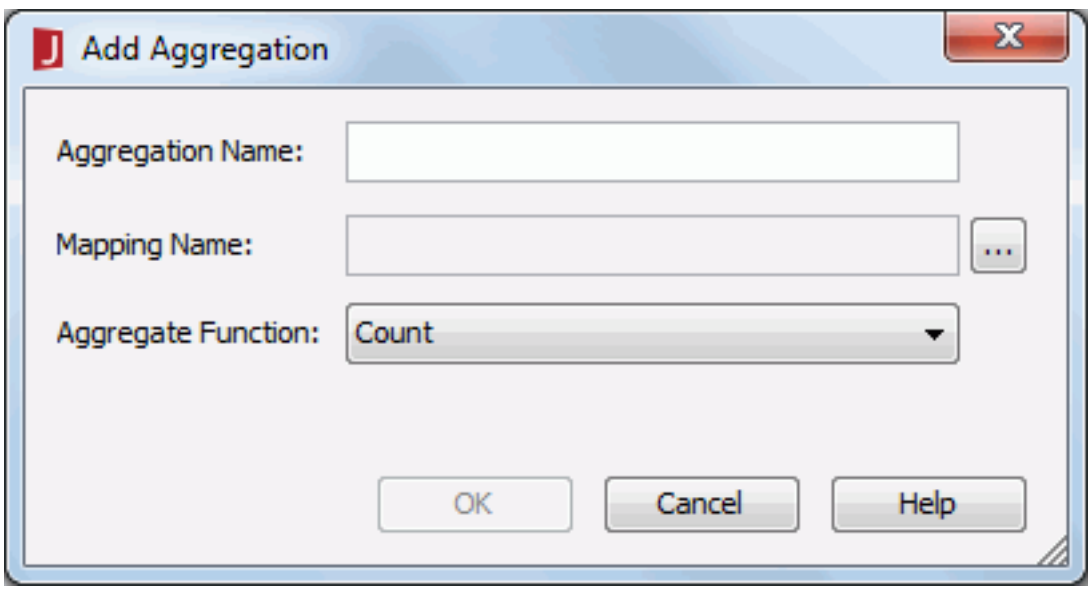
  - Expression @Customer#; will cause a syntax error. But @"Customer#" is ok.
  - If a field has the display name Category.Aggregation, when adding it to a formula, quote it as "Category.Aggregation" or "Category"."Aggregation".

Creating and using dynamic aggregations

You can also create dynamic aggregations by mapping them to the available resources such as group objects, detail objects in the current business view, and the dynamic formulas that have been created in a library component.

To create a dynamic aggregation:

1. In the Resource View panel of the main window, or in the Resources box of the report wizard, expand the **Dynamic Resources** > **Aggregations** node, then click **<Add Aggregation•gt;**. The Add Aggregation dialog appears.



2. In the Aggregation Name text field, specify the name of the dynamic aggregation.
3. Click ... next to the Mapping Name text field to specify the field on which the dynamic aggregation is based.
4. From the Aggregate Function drop-down list, specify the aggregate function.
5. When done, click **OK** to create the dynamic aggregation.

Once a dynamic aggregation is created, you can then drag it from the Resource View panel to the desired position in the library component to get the desired data, or use it when working with the report wizard.

For any dynamic aggregation, you can further modify if needed. To do this, right-click the aggregation and select **Edit** from the shortcut menu. Then in the Edit Aggregation dialog, edit the aggregation as required.

For dynamic aggregations that are no longer required, you can remove them. To delete an aggregation, right-click it and click **Delete** on the shortcut menu. Aggregations that have been used in the library component cannot be deleted.

Using local parameters in a library component

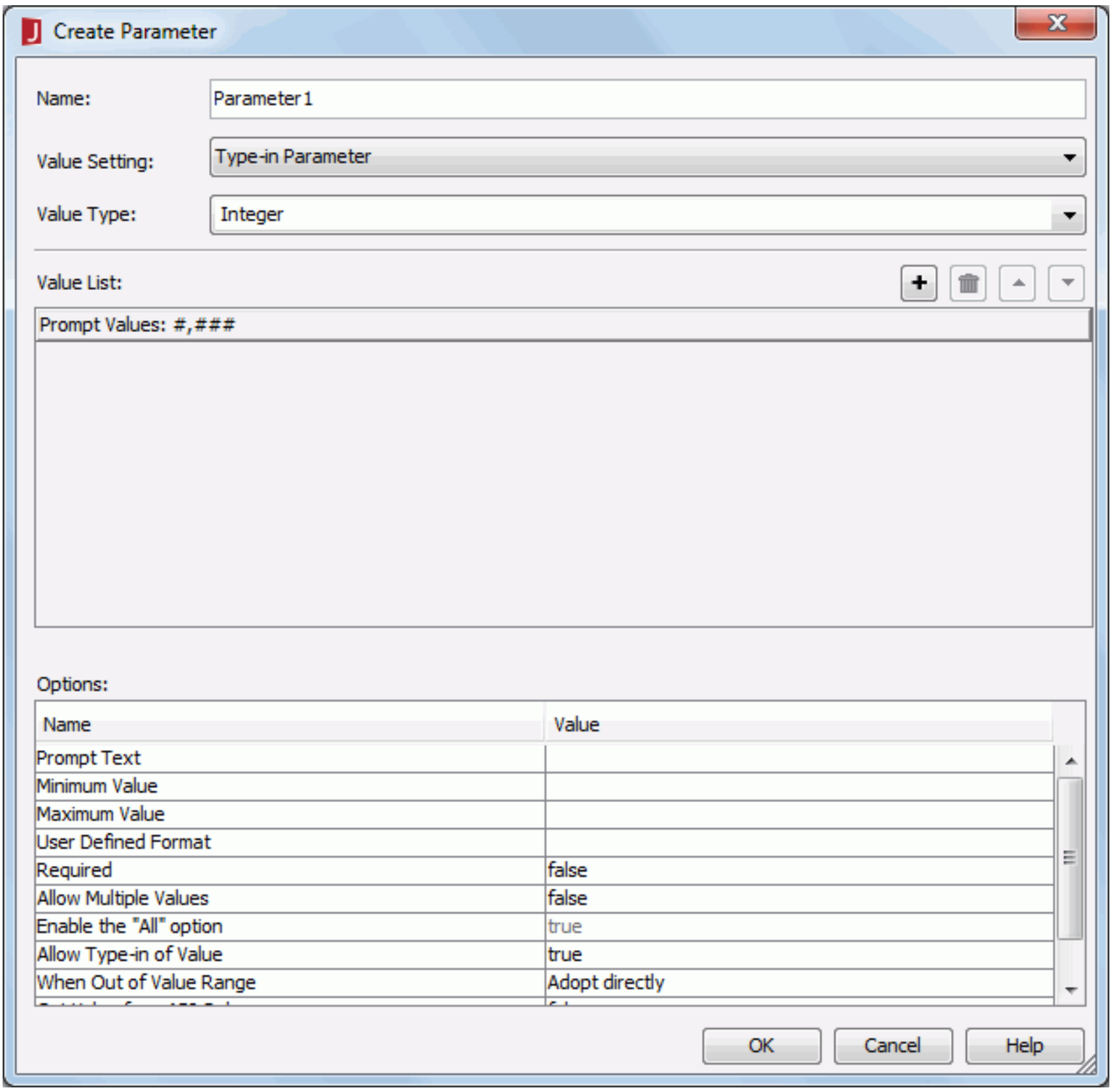
When working with a library component, sometimes you may find the predefined parameters in the catalog cannot meet your requirements. In this case, you can create some local parameters and use them in the library component to get the desired data. Then when you save the library component, the local parameters will be saved along with it as its resources.

Local parameters are library component level resources, which means they are only available to the library component for which they are created.

To create a local parameter:

1. Do one of the following:
   - In the Resource View panel of the main window, click **<Add Parameter•gt;** in the **Local Parameters** node.
   - Right-click a data component in the library component and select **Format Filter** from the shortcut menu. In the Edit Filter dialog, click ... next to the value text field, then in the Values dialog, click **<Add Parameter•gt;** in the Local Parameters node.

   The Create Parameter dialog appears.



2. Create the parameter according to your requirement.
3. Click **OK** to create the parameter and exist the dialog.

Once a local parameter is created, you can then drag it from the Resource View panel to the desired position in the library component, use it to dynamically filter data of a data component in the library component, add it to a parameter control or parameter form control in the library component, or reference it in a dynamic record level pass one formula used in the library component.

For any local parameter, you can further modify it if needed. To do this, right-click the parameter and select **Edit** from the shortcut menu. Then in the Edit Parameter dialog, edit the parameter as required. You can also change its name by right-clicking it and selecting **Rename** from the shortcut menu.

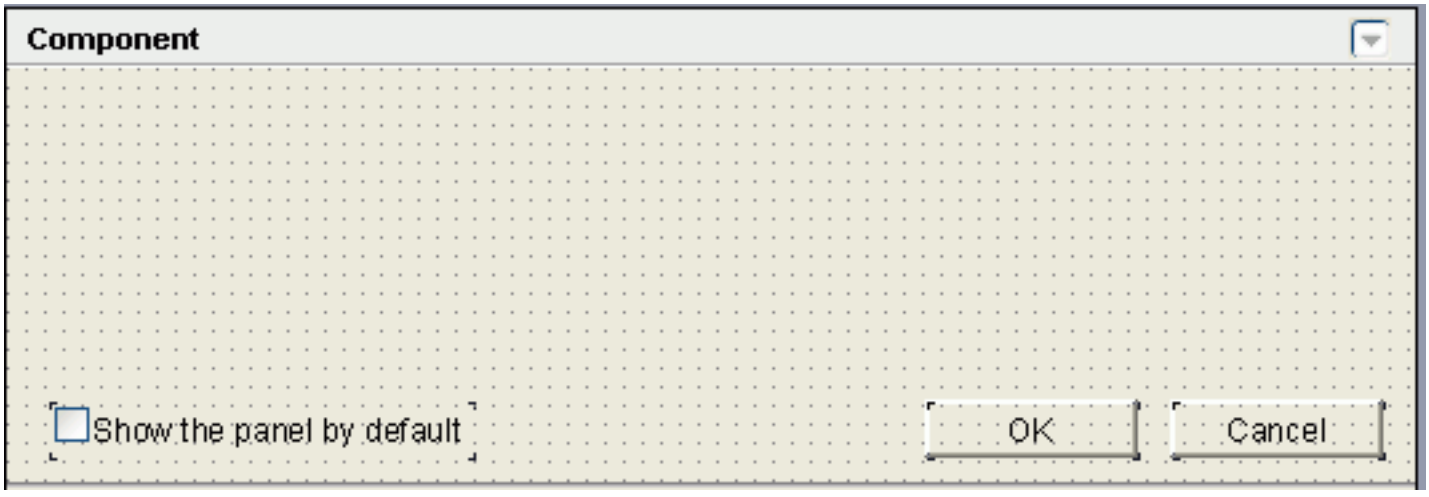
For local parameters that are no longer required, you can remove them. To delete a parameter, right-click it and click **Delete** on the shortcut menu. Local parameters that have been used in the library component cannot be deleted.

Using the configuration panel

Each library component can be equipped with a configuration panel. You can insert labels and web controls such as text fields, checkboxes, drop-down lists, and lists into the panel to filter or sort on the library component, or change properties of objects in the library component. The configuration panel is also used to specify parameter values.

You can display the configuration panel by checking the Display Configuration Panel checkbox on the top-right corner. The panel needs to be configured in JReport Designer and then used in a dashboard after the library component is published to JReport Server and inserted into a dashboard.

The initial configuration panel contains a title bar, a checkbox, two buttons as well as a label and text field for each parameter used by the data component.



The title bar contains the title of the library component and a button . You can use the button to show or hide the panel when the library component is in view mode as a dashboard component or inserted into a dashboard. The checkbox is used to specify whether to show the panel by default when the library component is inserted into a dashboard in JReport Server. You can check the checkbox by setting a property of the configuration panel Default Show to true in the Report Inspector.

All actions in this panel will be bound with the OK and Cancel buttons in it. You can click OK to apply the actions to the library component. The button Cancel is used to make this panel invisible and initialize the value of the web controls to the last values.

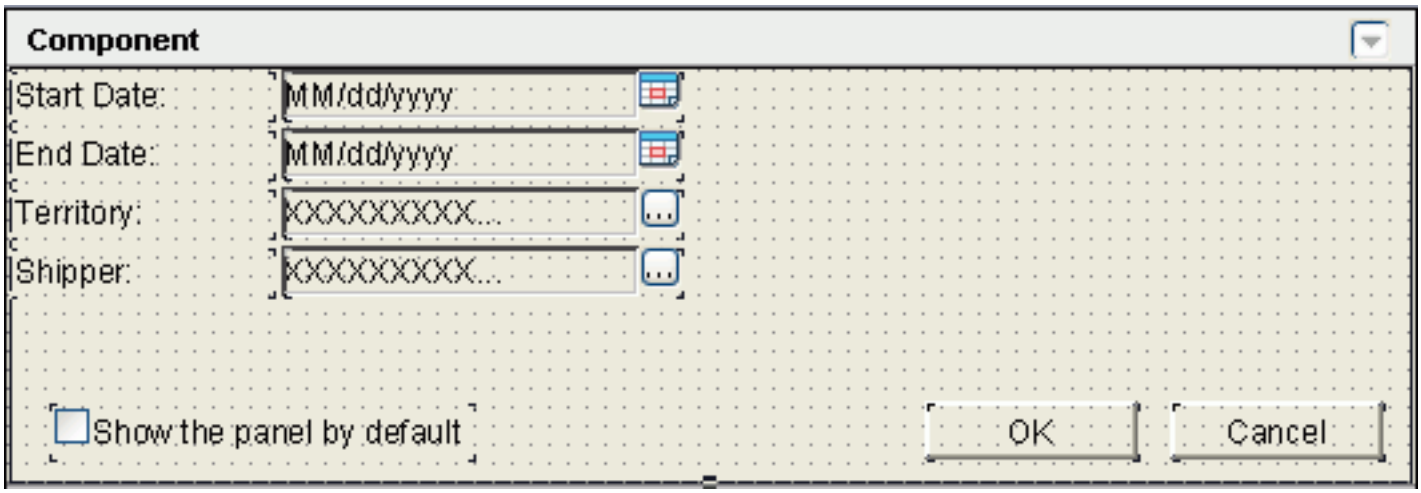
The height of the panel can be adjusted to the size of the content, but its width is equal to the width of the library component and cannot be adjusted. You can resize its width only by resizing the width of the library component. The two buttons are built-in buttons that cannot be deleted, cut, copied, or pasted, but their properties can be edited in the Report Inspector and they can be moved by dragging and dropping. Other components in the panel can be deleted.

The following topics shows what you can do with the configuration panel:

- Specifying parameter values using the panel
- Inserting objects into the panel
- Applying web behaviors to web controls in the panel

Specifying parameter values using the panel

When a library component contains parameters, the configuration panel will be automatically populated with all the parameters needed to run the library component, and web controls for the parameters will be created automatically in this panel. Therefore, this panel can be used to specify parameter values for the library component at runtime.



When the library component is published to JReport Server and inserted into a dashboard using JDashboard, the values of the parameters saved in the library component will be used as the default values. If JDashboard users want to change the parameter values, they can make use of the configuration panel published with the library component. After specifying new values in the configuration panel and clicking OK, the specified parameter values will be used to populate the data. The changed values will be saved in the dashboard and take effect when the dashboard is opened next time.

Inserting objects into the panel

You can insert labels and web controls, which are used as variables, into the configuration panel. The web controls that can be inserted into the configuration panel of a library component are: text field, checkbox, drop-down list, and list.

Inserting a label

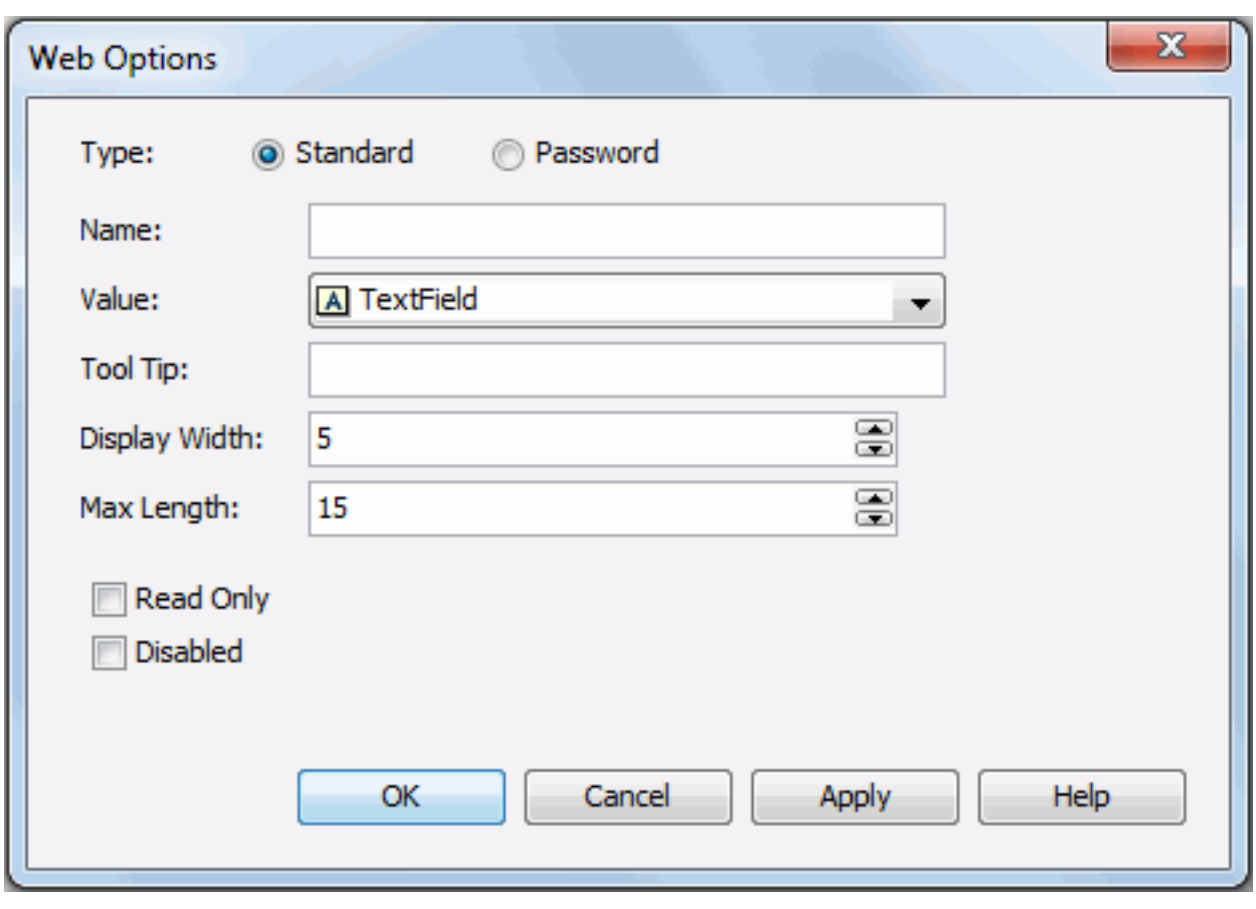
To insert a label into the configuration panel:

1. Do one of the following to insert the label:
   - Drag **Label** from the Toolbox panel to the destination.
   - On the menu bar, click **Insert** > **Label**, then drag the label to the required destination.
   - On the Insert toolbar, click the **Insert Label** button , then drag the label to the required destination.
2. Double-click the label to edit the text.

Inserting a text field

To insert a text field into the configuration panel and define its web options:

1. Do one of the following to insert the text field:
   - Drag **Text Field** from the Toolbox panel to the destination.
   - On the menu bar, click **Insert** > **Web Controls** > **Text Field**, then drag the text field to the required destination.
   - On the Insert toolbar, click the down arrow on the **Insert Control Objects** button and select **Text Field** from the drop-down list, then drag the text field to the required destination.
2. Right-click the text field and select **Web Options** from the shortcut menu. The Web Options dialog displays.



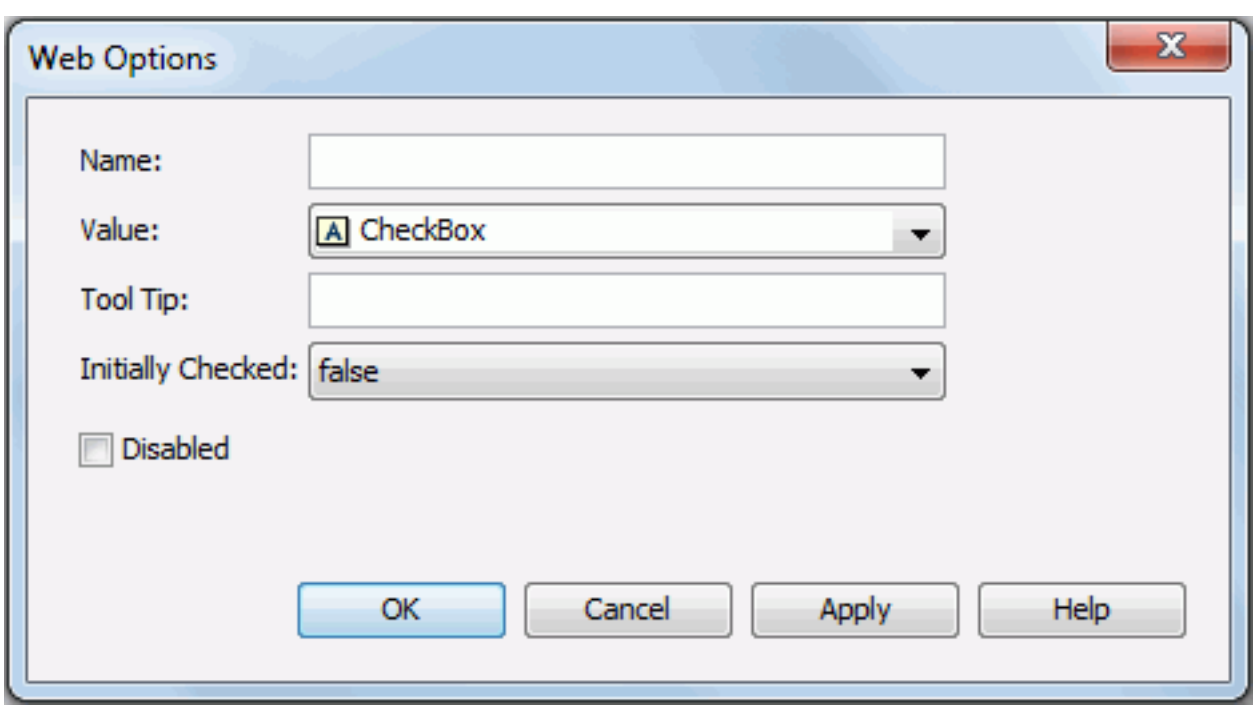
3. Select the type of the text field: Standard or Password.

   Standard means that the text field will be a normal text field. Password means that the text field will be a password box in which the typed characters will be displayed as asterisks.
4. Enter the name, value for the text field in the Name and Value fields.
5. In the Tool Tip field, enter the tip you want to show for the text field. When the component is displayed in a dashboard, when you hover the mouse over the text field, the tooltip will be displayed.
6. Set the character width in the Display Width field, and the maximum number of characters the user can enter in the Max Length field.
7. Check the **Read Only** option if you would like to set this text field to be read-only.
8. Check the **Disabled** checkbox if you want to make the text field disabled.
9. Upon finishing, click the **OK** button to close this dialog.

Inserting a checkbox

To insert a checkbox into the configuration panel and define its web options:

1. Do one of the following to insert the checkbox:
   - Drag **Checkbox** from the Toolbox panel to the destination.
   - On the menu bar, click **Insert** > **Web Controls** > **Checkbox**, then drag the checkbox to the required destination.
   - On the Insert toolbar, click the down arrow on the **Insert Control Objects** button and select **Checkbox** from the drop-down list, then drag the checkbox to the required destination.
2. Right-click the checkbox and select **Web Options** from the shortcut menu. The Web Options dialog displays.



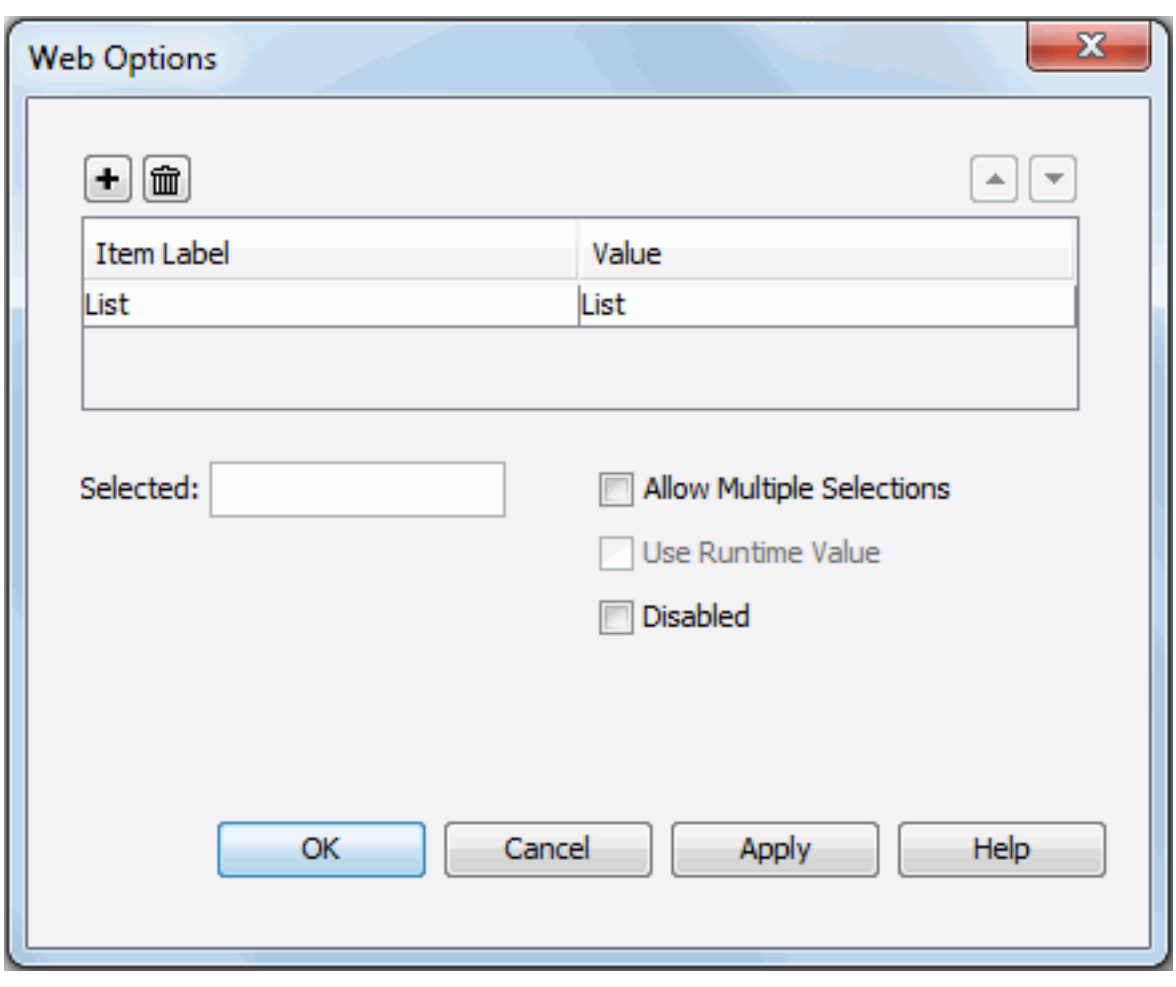
3. Specify the name, value of the checkbox in the Name and Value fields.
4. In the Tool Tip field, enter the tip you want to show for the checkbox button. When the component is displayed in a dashboard, when you hover the mouse over the text field, the tooltip will be displayed.
5. Set Initially Checked to **true** if you want the checkbox to be selected by default.
6. Check the **Disabled** checkbox if you want to make the checkbox disabled.
7. Upon finishing, click **OK** to close this dialog.

Inserting a list/drop-down list

List and drop-down list are generally called multivalue containers. A list or drop-down list provides a list of values for users to select from. A drop-down list occupies a smaller space than a list box so that the configuration panel can be laid out more compactly.

To insert a list/drop-down list into the configuration panel and define its web options:

1. Do one of the following to insert the list/drop-down list:
   - Drag **List/Drop-down List** from the Toolbox panel to the destination.
   - On the menu bar, click **Insert** > **Web Controls** > **List/Drop-down List** , then drag the list/drop-down list to the required destination.
   - On the Insert toolbar, click the down arrow on the **Insert Control Objects** button and select **List/Drop-down List** from the drop-down list, then drag the list/drop-down list to the required destination.
2. Right-click the list/drop-down list and select **Web Options** from the shortcut menu. The Web Options dialog displays.



3. Specify the item labels and the value of each item in the items box by inputting some strings as the labels and values respectively.

   If you want to use a DBField/formula field/parameter field to control the value, click in the Value column to insert one with the Insert Fields dialog, and then select the format for the inserted field from the drop-down list in the Item Label column. Check the **All** radio button in the Insert Field dialog if you want to add an All value to the list/drop-down list. Then when All is selected as value of the list/drop-down list at runtime, all filter actions defined on the list/drop-down list will not take effect, and if you applied some other web action that needs value from the list/drop-down list, a Null value will be returned.
4. For a list web control, you can check the **Allow Multiple Selections** checkbox if you want to allow multiple items to be selected.
5. In the Selected field, specify the selected value for the list/drop-down list.
6. To use the runtime value as the selected value, check **Use Runtime Value**.
7. If you want to disable the list/drop-down list, check the **Disabled** option.
8. When done, click **OK** to apply the settings.

Note: If you have specified a tooltip for any web control, when the library component is opened in JDashboard, the tooltip of the web control will be displayed differently on different browsers:

- On Internet Explorer, the tip text will be automatically wrapped if its length exceeds the maximal tip width the browser allows. In addition, for Internet Explorer, you can manually add a new line by adding &#10 or &#13 ahead of the text when editing it in the Tool Tip field.
- On Firefox, the tip text will be displayed in one line with the maximal tip width the browser allows, and the text that cannot be displayed within the width will be cut off and replaced by ellipsis.

Applying web behaviors to web controls in the panel

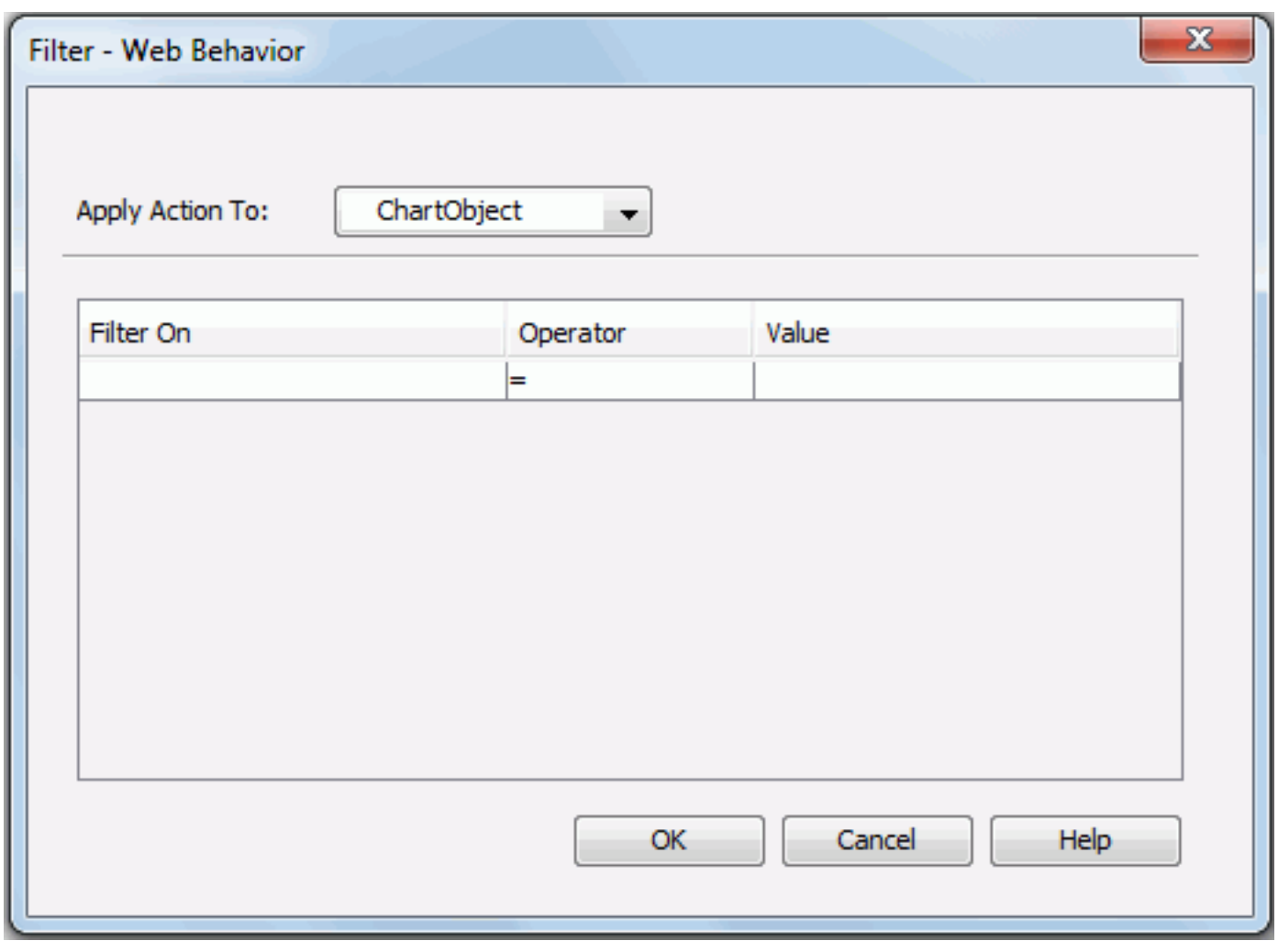
You can apply some web behaviors, including Filter, Sort and Change Property, to the web controls you insert in the configuration panel of a library component so as to filter or sort the records in the table, chart or crosstab of the library component or change the properties of an object in the library component at runtime.

Filter

The Filter web behavior enables you to filter the records in a table, chart or crosstab.

To bind the Filter web behavior to a web control:

1. Right-click the web control, and then select **Web Behaviors** > **Filter** from the shortcut menu. The Filter - Web Behavior dialog appears.



2. From the Apply Action To drop-down list, select the data component the records of which you want to filter.
3. In the Filter On column, specify the field on which to filter the records. The field may be a column in the component, or be specified by the value of a web control.
4. Specify the operator and value. The value may be input by yourself, or be the value of a web control.
5. Click **OK** to accept the settings.

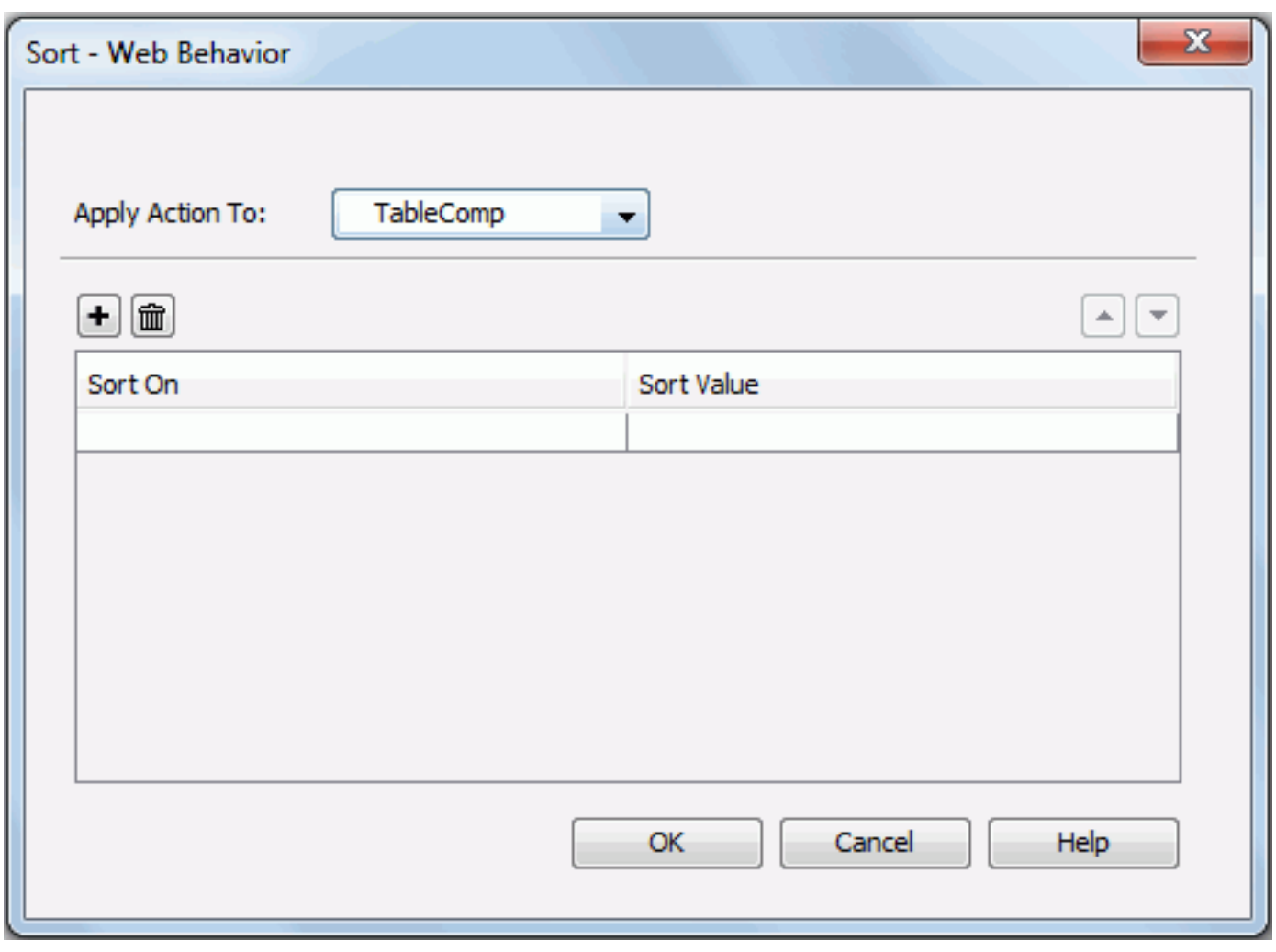
   Then at runtime, when end users specify a value for the web control and click OK in the configuration panel, the records in the data component will be filtered.

Sort

The Sort web behavior enables you to sort the records in a table, chart or crosstab.

To bind the Sort web behavior to a web control:

1. Right-click the web control, and then select **Web Behaviors** > **Sort** from the shortcut menu. The Sort - Web Behavior dialog appears.



2. From the Apply Action To drop-down list, select the data component the records of which you want to sort.
3. In the Sort On column, specify the field on which to sort the records. The field may be a column in the component, or be specified by the value of a web control.
4. Specify the sort value. The value may be Ascending or Descending, or be the value of a web control.
5. If necessary, click to add a new sort condition.

   To delete a sort condition, click . To adjust the order of the sort conditions, click or .
6. Click **OK** to accept the settings.

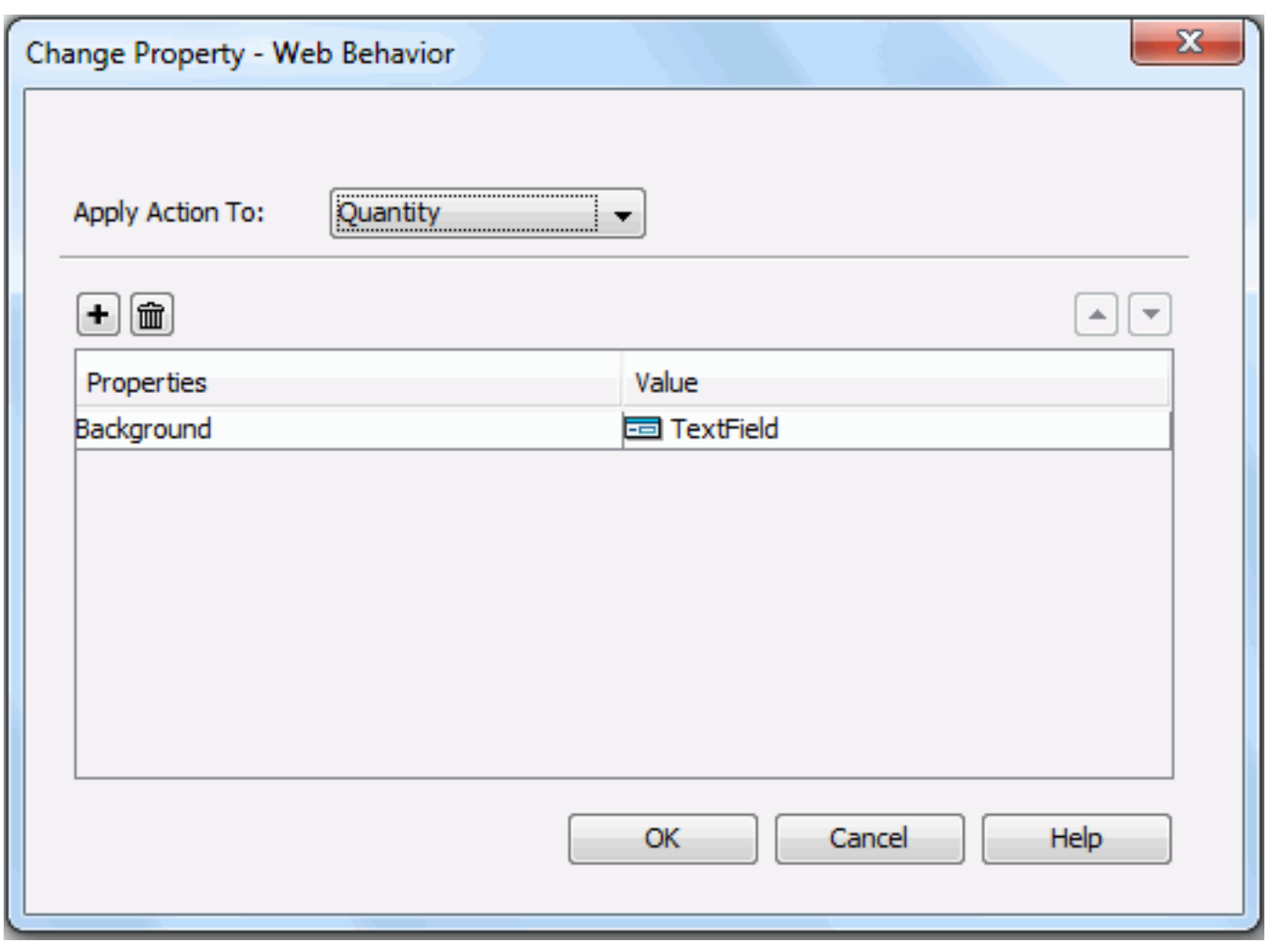
   Then at runtime, when end users specify a value for the web control and click OK in the configuration panel, the records in the data component will be sorted.

Change Property

The Change Property web behavior enables you to change properties of an object in the library component.

To bind the Change Property web behavior to a web control:

1. Right-click the web control, and then select **Web Behaviors** > **Change Property** from the shortcut menu. The Change Property - Web Behavior dialog appears.



2. From the Apply Action To drop-down list, select the object the properties of which you want to change.
3. In the Properties column, specify the property you want to change. All the properties of the object you select from the Apply Action To drop-down list are listed here.
4. In the Value column, specify the value of the property. The value may be input by yourself, or be specified by the value of a web control.
5. If necessary, click to add a new line to change a property.

   To delete a property line, click . To adjust the order of the properties, click or .
6. Click **OK** to accept the settings.

   Then at runtime, when end users specify a value for the web control and click OK in the configuration panel, the properties of the specified object in the library component will be changed.

Delivering messages between library components

A message is a carrier of information that can be sent or received between library components. It makes the communication between the library components more flexible. A library component can send out a built-in message such as Filter, Sort, Parameter, Parameters or a user defined message when a specific event occurs on it. The data component in a library component can do actions such as filter, sort, re-running using predefined parameter values or changing properties to respond a message when it receives the message. The way to send or receive a message between library components is defined in JReport Designer, then in JReport Server, when the library components are inserted into a dashboard using JDashboard, a library component will receive the message sent out by another library component or itself when a specific event occurs and do corresponding action to respond to the message.

A message consists of message ID, message name and message value set. A message ID is a natural number. JReport reserves the IDs from 1 to 1000, which are used in built-in messages, so you can only use IDs beyond 1000 to identify user defined messages. A message name is the name of a message. A message value set is a set of key-value pairs. Each key-value pair has one key and one value. A key or a value can be static or dynamic. A key is of String type, and each value has its own data type, which can be Number, Integer, String, Date, Time, Date Time, Boolean or Currency.

The following topics detail how to send or receive a message:

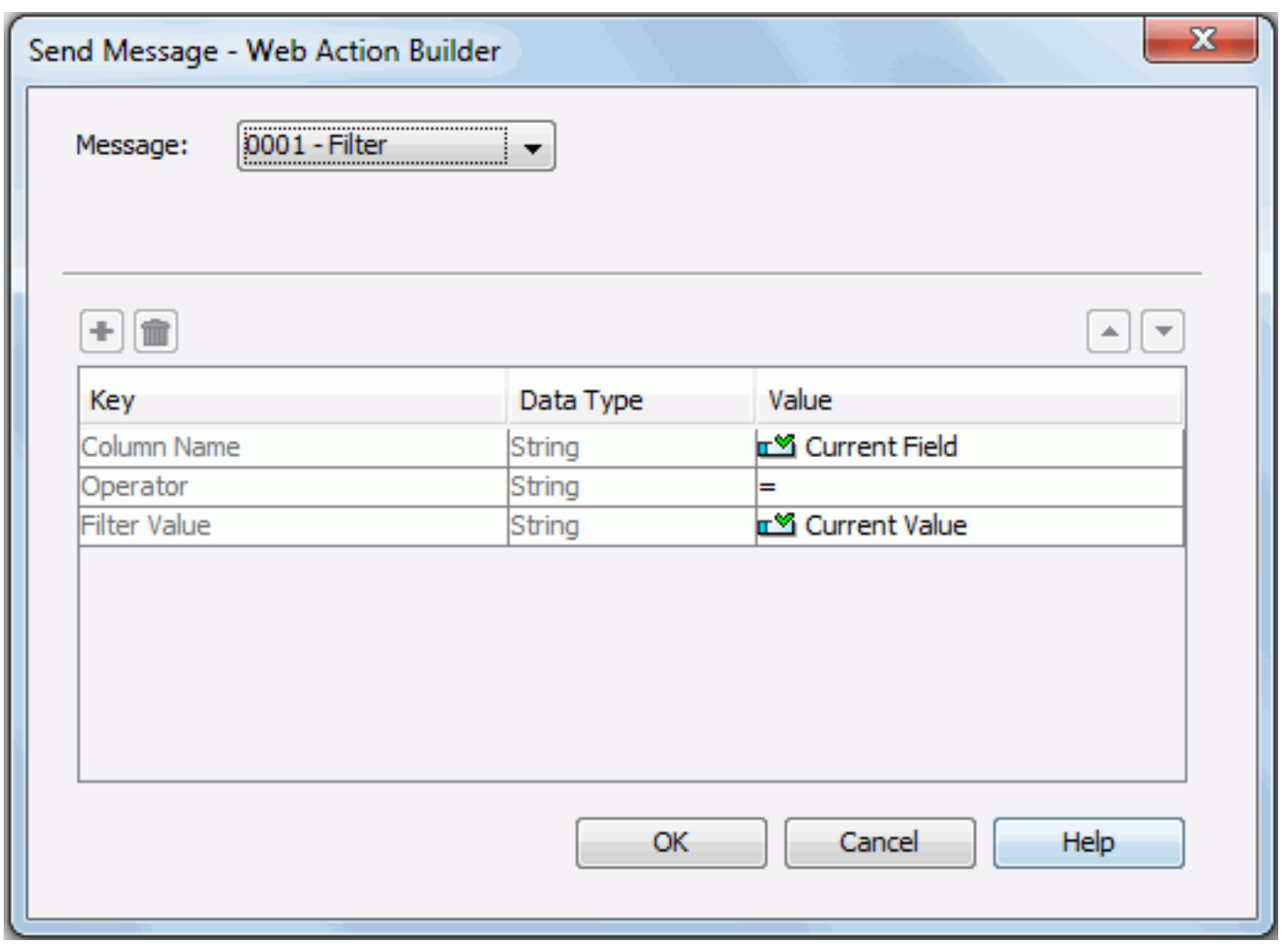
- Sending out messages
- Receiving messages
- Example of delivering a message

Sending out messages

You can specify whether to send out a message, including the built-in message 0001 - Filter, 0002 - Sort, or 0003 - Parameter, or a user defined message when a specific event occurs on a certain object in a library component. The objects from which messages can be sent out include fields in table/crosstab, labels in the content, geographic maps, parameter controls, the Submit button of parameter form controls, and some chart elements (data markers, legend and category axis labels).

To send out a message:

1. Do one of the following to open the Send Message - Web Action Builder dialog.



   - Right-click a field or label in a table/crosstab, or a label, a parameter control or the Submit button of a parameter form control which is not in the library component's configuration panel, select **Display Type** from the shortcut menu to display the Display Type dialog, select a trigger event from the drop-down list in the Events column and click ... in the Actions column of the Web Behaviors box to display the Web Action List dialog, then select ***SendMessage** and click **OK**.
   - In the Behaviors tab of some chart formatting dialogs, select a trigger event from the drop-down list in the Events column and click ... in the Actions column, then select ***SendMessage** and click **OK** in the displayed Web Action List dialog.
   - Right-click a geographic map and select **Geographic Map Wizard** from the shortcut menu, or drag **Geographic Map** from the Toolbox to a library component to display the Create Geographic Map wizard, select a dataset in the Data screen and add a field to the Drill Path box in the Display screen of the wizard, click **Web Behaviors** button in the Display screen to display the Web Behaviors dialog, select a trigger event from the drop-down list in the Events column, click in the Actions column and click ... in the Actions column. Then select ***SendMessage** in the Web Action List dialog and click **OK**.
2. Select the message you want to send out from the Message drop-down list.
   - **0001 - Filter**
     Specifies to send out the built-in Filter message, which will ask the data component that receives the message to filter itself based on the filter condition defined in the message.
   - **0002 - Sort**
     Specifies to send out the built-in Sort message, which will ask the data component that receives the message to sort itself based on the sort condition defined in the message.
   - **0003 - Parameter**
     There are two types:
     - **Automatic**
       Specifies to send out the built-in Parameter message with all the parameter values used in the current library component, which will ask the library component in which the data component that receives the message is to rerun itself using the same parameter values.
     - **Customized**
       Specifies to send out the built-in Parameter message, which will ask the library component in which the data component that receives the message is to rerun itself using the parameter value defined in the message.
   - **User Defined**
     Specifies to send out a user defined message.
3. Define the message as you want.

   For the built in message, you can change the values for the keys in the Value column. To make the condition dynamic, select the values under the Dynamic Key node.

   For a user defined message,

   a. Specify the ID and Name of the message in the ID and Name text boxes. The ID should be a natural number beyond 1000.

   b. Click + to add a message key-value line for the user defined message.

   c. In the Key column, specify a key from the drop-down list or select the **<Input...>** item from the list and then input a key into the text box.

   d. In the Value column, specify the value of the key from the drop-down list or select **<Input...>** from the list and then input a value into the text box.

   e. If you choose to input the value by yourself in the above step, you need to specify the data type of the value. If the value is selected from the Value drop-down list, the data type of the value is automatically displayed in the Data Type column and cannot be changed.

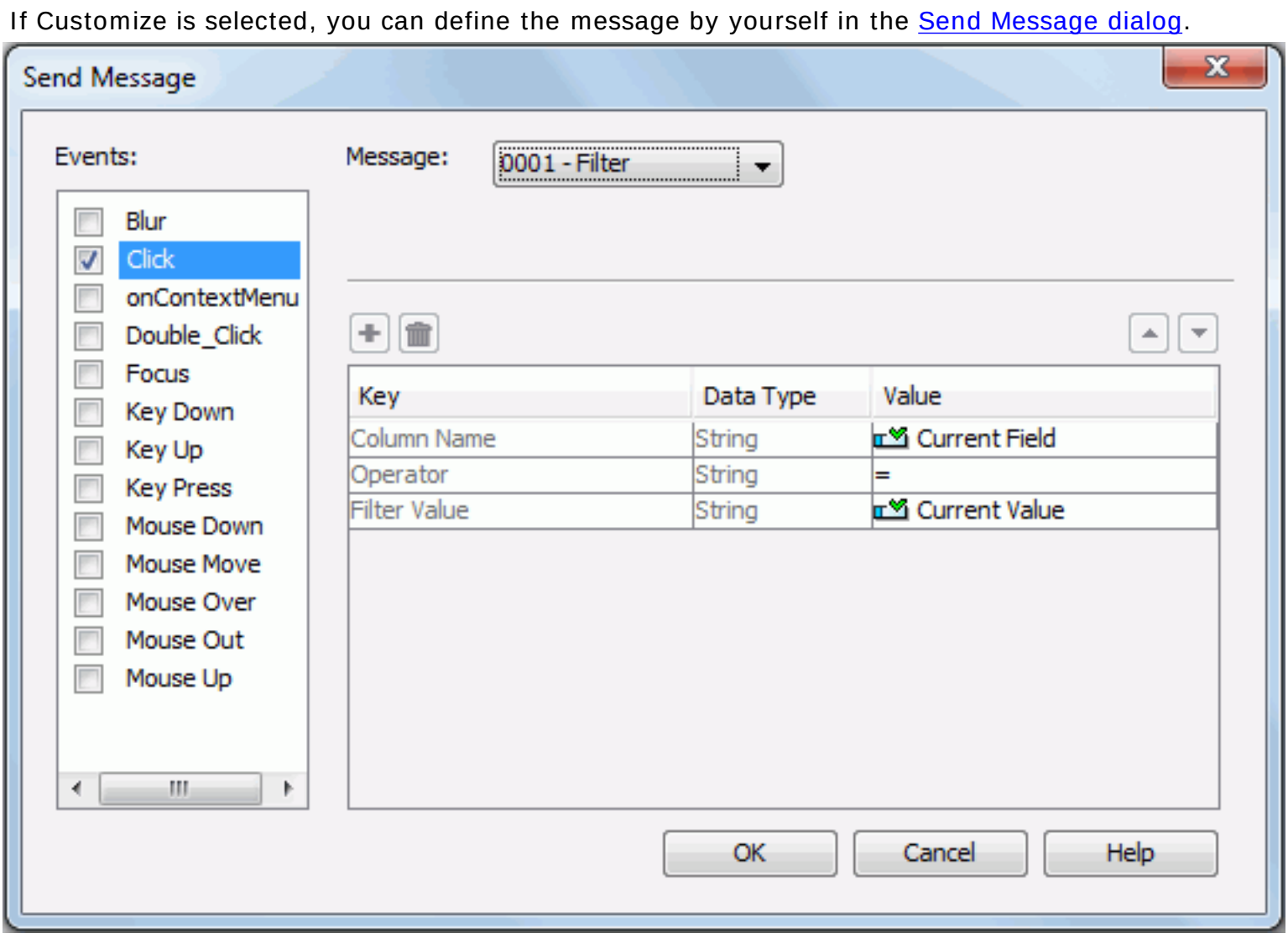
   f. If necessary, click + to add a new message key-value line and specify the key, data type and value respectively.

   To delete a message key-value line, select it and click 🗑. To adjust the order of the lines, click ▲ or ▼.

   Tip: You can also create a blank user defined message by only specifying the ID and name of the message.
4. Click the **OK** button to confirm the settings.
5. In the dialog you are returned to, add more event lines and repeat the above steps to specify to send out more messages as you want.
6. When done, click **OK** to finish defining the messages to send out.

In addition to the above method, another more convenient way to send a message is by using the Send Message command. To do this, right-click a field in a table/crosstab, a label in the content, a parameter control, the Submit button of a parameter form control, a data marker or an category axis label or the legend in a chart in a library component, from which you want to send the message, and select **Send Message** > **Filter**/**Sort**/**Parameter**/**Customize** from the shortcut menu (for the Submit button of a parameter form control, only Parameter and Customize are available).

- If Filter is selected, a built-in Filter message, whose trigger event is Click by default, will be created.
- If Sort is selected, a built-in Sort message, whose trigger event is Click by default, will be created.
- If Parameter is selected, a built-in Parameter message with the Automatic type, whose trigger event is Click by default, will be created.
- If Customize is selected, you can define the message by yourself in the Send Message dialog.



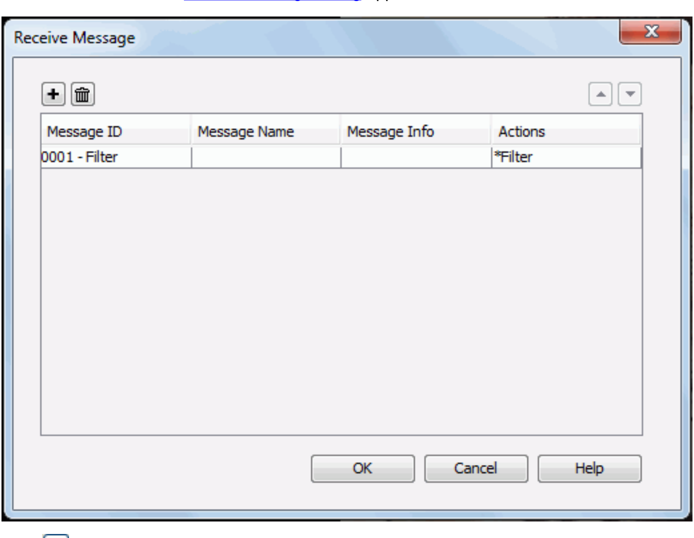
1. Check a trigger event in the Events box, then click on the name of the event to select it.
2. Define the message that will be triggered on the event.
3. Repeat the above two steps to define other messages that will be sent out from the object on different events.
4. Click **OK** to accept the settings.

Receiving messages

The data components, which refers to tables, charts, crosstabs and geographic maps, in a library component can receive messages, including the built-in messages 0001 - Filter, 0002 - Sort, 0003 - Parameter, and user defined messages that are sent out by itself or another library component. You can then specify actions such as filtering, sorting, running library component with predefined parameter values and changing object properties on the data component to respond the received messages.

To make a data component receive messages and respond corresponding web actions, take the following steps:

1. Select the data component in a library component, right-click and select **Receive Message** from the shortcut menu. The Receive Message dialog appears.



2. Click to add a message line.
3. Select the message you want the data component to receive from the Message ID column.
   - 0001 - Filter
     When the data component receives this message, it will filter itself based on the filter condition defined in the message.
   - 0002 - Sort
     When the data component receives this message, it will sort itself based on the sort condition defined in the message.
   - 0003 - Parameter
     When the data component receives this message with the type set to Automatic, the library component which contains the data component will rerun with all the parameter values defined in the message.

     When the data component receives this message with the type set to Customized, the library component which contains the data component will rerun with the parameter value defined in the message.
   - **User Defined**
     A data component in a library component can also receive user defined messages and then respond web actions, such as Filter, Sort, Property and Parameter.

     The Filter web action enables you to filter the records of the data component when it receives the message.

     The Sort web action enables you to sort the records of the data component when it receives the message.

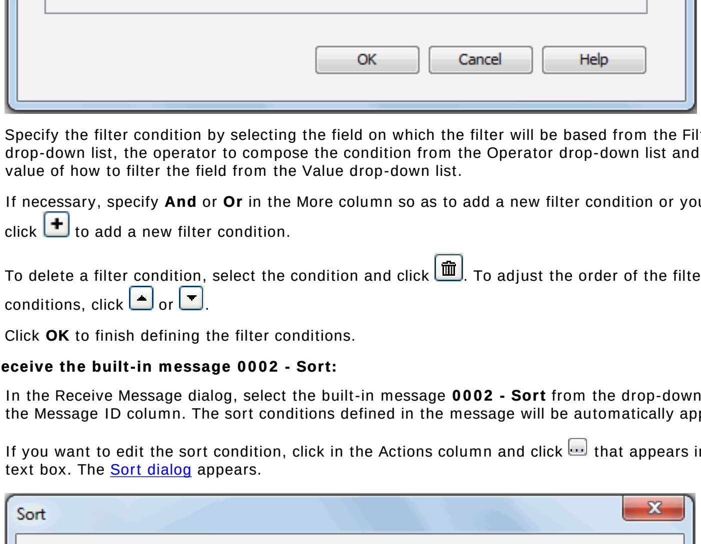
     The Property web action enables you to change properties of the data component itself when it receives the message.

     The Parameter web action enables you to run a library component, especially a library component with parameters using predefined parameter values, or re-run the library component with predefined parameter values when the message is received.
4. If a built-in message is selected, the action the message will respond cannot be changed, but you can edit how the action will be performed as you want.

   If a user define message is selected, you can specify the action the data component will respond when it receives the message and define how the action will be performed.
5. Repeat the above steps to add more messages to receive and define the actions to respond the messages. To delete a message, select it and click .
6. For message of the same ID and name, you can make the data component receive it more than once, and in which order the actions defined in the message will be executed is determined by the order of the messages in the list. To adjust the order, select the message and click or , then for message with the same ID and name, the higher a message is placed in the list, the earlier the action defined in it will be executed.
7. Upon finish, click **OK** to close the dialog.

To receive the built-in message 0001 - Filter:

1. In the Receive Message dialog, select the built-in message **0001 - Filter** from the drop-down list of the Message ID column. The filter condition defined in the message will be automatically applied.
2. If you want to edit the filter condition, click in the Actions column and click that appears in the text box. The Filter dialog appears.

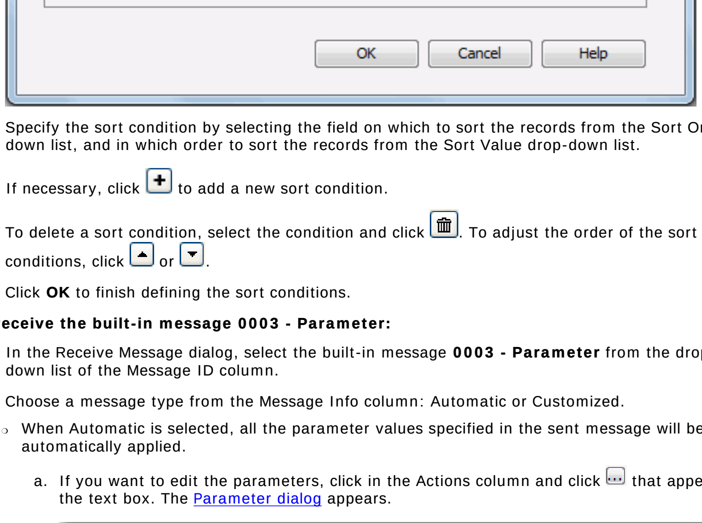


3. Specify the filter condition by selecting the field on which the filter will be based from the Filter On drop-down list, the operator to compose the condition from the Operator drop-down list and the value of how to filter the field from the Value drop-down list.
4. If necessary, specify **And** or **Or** in the More column so as to add a new filter condition or you can click to add a new filter condition.

   To delete a filter condition, select the condition and click . To adjust the order of the filter conditions, click or .
5. Click **OK** to finish defining the filter conditions.

To receive the built-in message 0002 - Sort:

1. In the Receive Message dialog, select the built-in message **0002 - Sort** from the drop-down list of the Message ID column. The sort conditions defined in the message will be automatically applied.
2. If you want to edit the sort condition, click in the Actions column and click that appears in the text box. The Sort dialog appears.

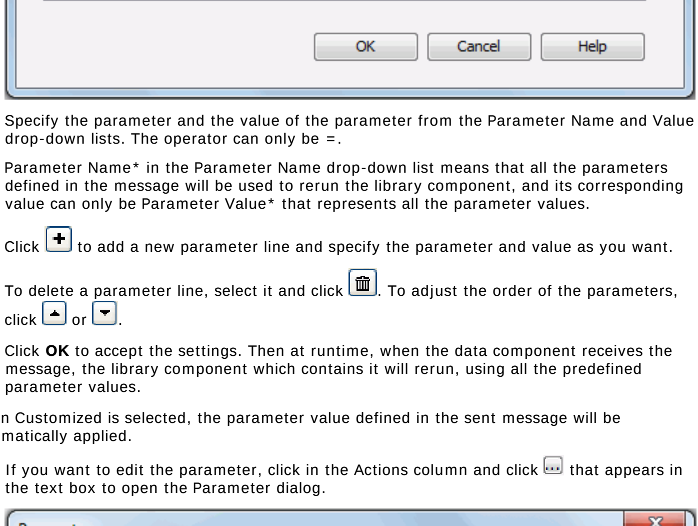


3. Specify the sort condition by selecting the field on which to sort the records from the Sort On drop-down list, and in which order to sort the records from the Sort Value drop-down list.
4. If necessary, click to add a new sort condition.

   To delete a sort condition, select the condition and click . To adjust the order of the sort conditions, click or .
5. Click **OK** to finish defining the sort conditions.

To receive the built-in message 0003 - Parameter:

1. In the Receive Message dialog, select the built-in message **0003 - Parameter** from the drop-down list of the Message ID column.
2. Choose a message type from the Message Info column: Automatic or Customized.
   - When Automatic is selected, all the parameter values specified in the sent message will be automatically applied.
     a. If you want to edit the parameters, click in the Actions column and click that appears in the text box. The Parameter dialog appears.



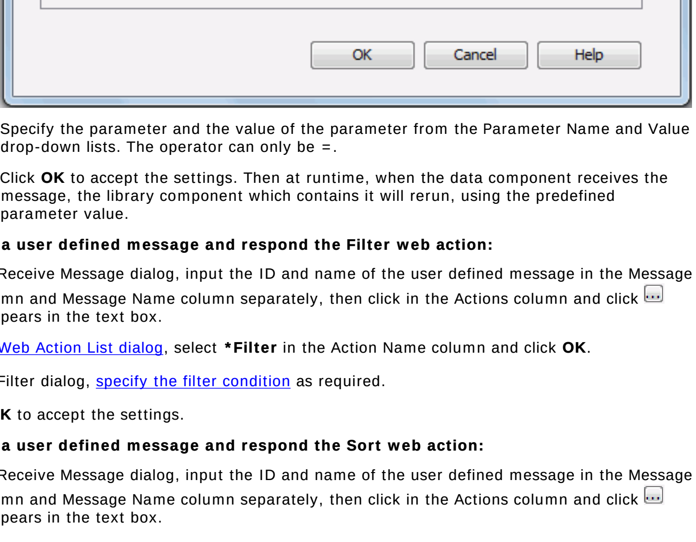
     b. Specify the parameter and the value of the parameter from the Parameter Name and Value drop-down lists. The operator can only be =.

     Parameter Name* in the Parameter Name drop-down list means that all the parameters defined in the message will be used to rerun the library component, and its corresponding value can only be Parameter Value* that represents all the parameter values.

     c. Click to add a new parameter line and specify the parameter and value as you want.

     To delete a parameter line, select it and click . To adjust the order of the parameters, click or .

     d. Click **OK** to accept the settings. Then at runtime, when the data component receives the message, the library component which contains it will rerun, using all the predefined parameter values.
   - When Customized is selected, the parameter value defined in the sent message will be automatically applied.
     a. If you want to edit the parameter, click in the Actions column and click that appears in the text box to open the Parameter dialog.



     b. Specify the parameter and the value of the parameter from the Parameter Name and Value drop-down lists. The operator can only be =.

     c. Click **OK** to accept the settings. Then at runtime, when the data component receives the message, the library component which contains it will rerun, using the predefined parameter value.

To receive a user defined message and respond the Filter web action:

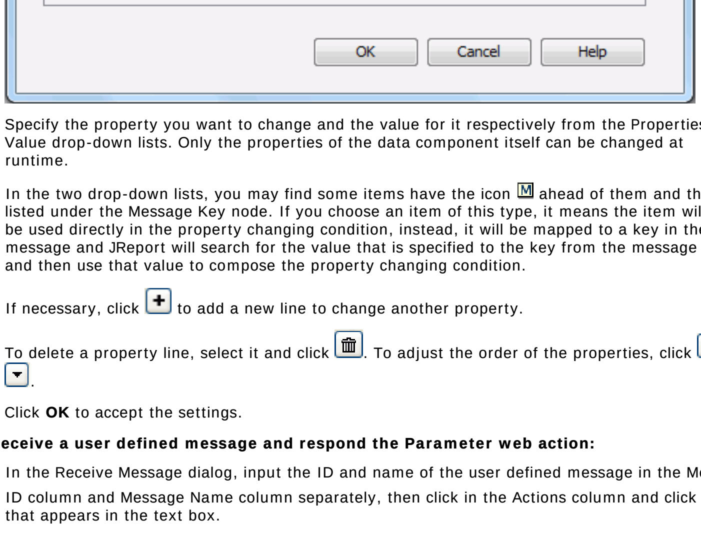
1. In the Receive Message dialog, input the ID and name of the user defined message in the Message ID column and Message Name column separately, then click in the Actions column and click that appears in the text box.
2. In the Web Action List dialog, select ***Filter** in the Action Name column and click **OK**.
3. In the Filter dialog, specify the filter condition as required.
4. Click **OK** to accept the settings.

To receive a user defined message and respond the Sort web action:

1. In the Receive Message dialog, input the ID and name of the user defined message in the Message ID column and Message Name column separately, then click in the Actions column and click that appears in the text box.
2. In the Web Action List dialog, select ***Sort** in the Action Name column and click **OK**.
3. In the Sort dialog, specify the sort condition as required.
4. Click **OK** to accept the settings.

To receive a user defined message and respond the Property web action:

1. In the Receive Message dialog, input the ID and name of the user defined message in the Message ID column and Message Name column separately, then click in the Actions column and click that appears in the text box.
2. In the Web Action List dialog, select ***Property** in the Action Name column and click **OK**. The Change Property dialog appears.



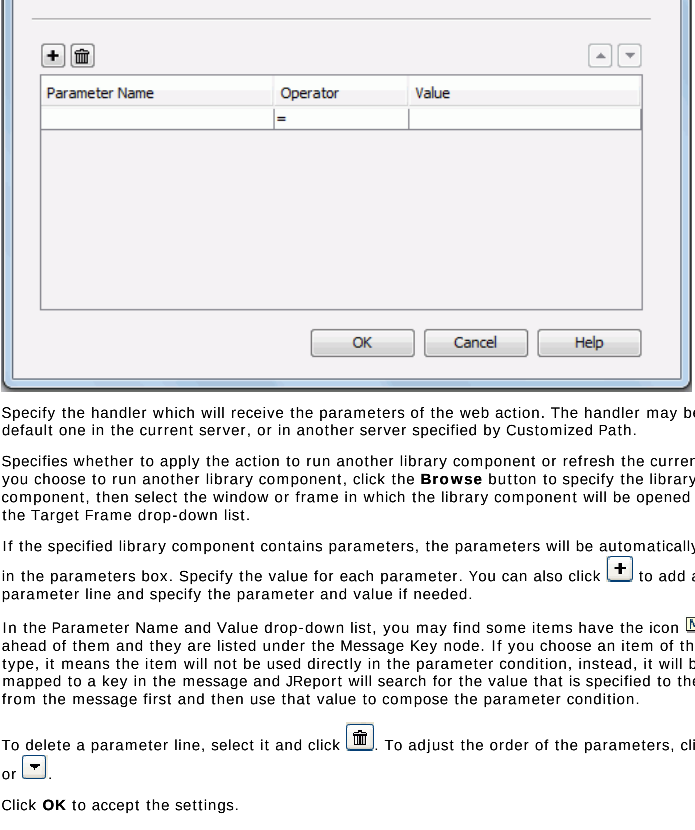
3. Specify the property you want to change and the value for it respectively from the Properties and Value drop-down lists. Only the properties of the data component itself can be changed at runtime.

   In the two drop-down lists, you may find some items have the icon ahead of them and they are listed under the Message Key node. If you choose an item of this type, it means the item will not be used directly in the property changing condition, instead, it will be mapped to a key in the message and JReport will search for the value that is specified to the key from the message first and then use that value to compose the property changing condition.
4. If necessary, click to add a new line to change another property.

   To delete a property line, select it and click . To adjust the order of the properties, click or .
5. Click **OK** to accept the settings.

To receive a user defined message and respond the Parameter web action:

1. In the Receive Message dialog, input the ID and name of the user defined message in the Message ID column and Message Name column separately, then click in the Actions column and click that appears in the text box.
2. In the Web Action List dialog, select ***Parameter** in the Action Name column and click **OK**. The Parameter dialog appears.



3. Specify the handler which will receive the parameters of the web action. The handler may be the default one in the current server, or in another server specified by Customized Path.
4. Specifies whether to apply the action to run another library component or refresh the current. If you choose to run another library component, click the **Browse** button to specify the library component, then select the window or frame in which the library component will be opened from the Target Frame drop-down list.
5. If the specified library component contains parameters, the parameters will be automatically listed in the parameters box. Specify the value for each parameter. You can also click to add a parameter line and specify the parameter and value if needed.

   In the Parameter Name and Value drop-down list, you may find some items have the icon ahead of them and they are listed under the Message Key node. If you choose an item of this type, it means the item will not be used directly in the parameter condition, instead, it will be mapped to a key in the message and JReport will search for the value that is specified to the key from the message first and then use that value to compose the parameter condition.
6. To delete a parameter line, select it and click . To adjust the order of the parameters, click or .
7. Click **OK** to accept the settings.

Note: In the drop-down lists for editing the conditions to respond a message, you may find that some items have the icon ahead of them and they are listed under the Message Key node. If you choose an item of this type, it means the item will not be used directly in a condition, instead, it will be mapped to a key in the message and JReport will find the value that is specified to the key from the message first and then use that value to compose the condition.

Example of delivering a message

This example demonstrates how to deliver a filter user defined message between two library components. It is divided into the following steps:

1. Create two library components.
2. Define to deliver a filter user defined message between the library components.
3. Publish the library components to JReport Server.
4. Deliver the message in a dashboard.

Creating the library components

1. Open the catalog SampleReports.cat.
2. Use the business view SaleStat in Data Source 1 to create two library components and save them as Component 1.lc and Component 2.lc.

   For Component 1, create a table which displays the Order ID, Order Date, Quantity and Unit Price DBFields, is grouped by Assigned Region, and uses Total Actual as the summary field. For Component 2, create a crosstab using Order ID and Assigned Region respectively as the row and column fields, and Total Actual as the summary field. Apply the style **Commercial** to them.

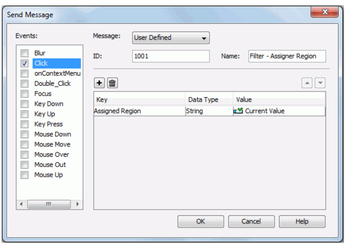
Defining to deliver a filter user defined message

In this step, we will define to send a filter user defined message from Component 1 when the Click event occurs on the Assigned Region group by field in the table, and make Component 2 receive the message and filter on the Assigned Region field once it receives the message.

To define to send a filter user defined message on Component 1:

1. Right-click the group by field **Assigned Region** and click **Send Message** > **Customize** on the shortcut menu to bring out the Send Message dialog.
2. In the Events box, check **Click** as the trigger event, then click on the event to activate the message options on the right.
3. From the Message drop-down list, select **User Defined**, then input **1001** and **Filter - Assigned Region** as the message ID and name.
4. Click ⊞ to add a message line.
5. Select the **<Input>** item from the drop-down list in the Key column and input **Assigned Region** in the text box.
6. Select **Current Value** from the drop-down list in the Value column. String is displayed automatically in the Data Type column.

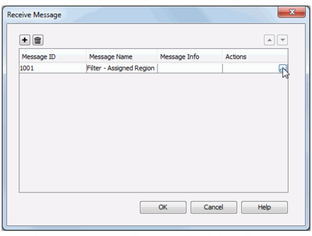
   The message is defined as follows:



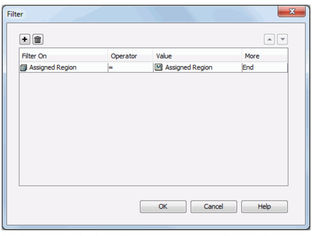
7. Click **OK** in the dialog to finish defining the message.

To make Component 2 receive the message and filter the Assigned Region field:

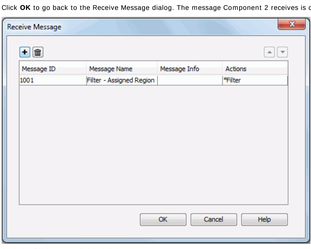
1. Right-click the crosstab and select **Receive Message** from the drop-down list to display the Receive Message dialog.
2. Click ⊞ to add a message line.
3. Select the **<Input>** item from the drop-down list in the Message ID column, then input **1001** in the text box.
4. Input **Filter - Assigned Region** in the text box of the Message Name column.
5. Click the blank text box in the Actions column, then click ⊞ that appears.



6. In the Web Action List dialog, select ***Filter** and click **OK** to display the Filter dialog.
7. Select **Assigned Region** from the Filter On drop-down list, keep the default operator, then select **<Input>** under the Message Key node in the Value drop-down list and input **Assigned Region** in the text box.



8. Click **OK** to go back to the Receive Message dialog. The message Component 2 receives is defined as follows:



9. Click the **OK** button to finish defining the message.

Publishing the library components

After defining how to deliver the message, save the two library components. Then, start JReport Server and publish the library components along with the catalog SampleReports.cat to the component library in JReport Server. For detailed steps about publishing resources from JReport Designer to JReport Server, see Publishing resources remotely.

Delivering the filter user defined message in a dashboard

When the two library components are inserted into a dashboard, when the Click event occurs on the Assigned Region group by field in the table, the message will be delivered between them, and the records in the crosstab will then be filtered according to the specified condition.

To deliver the message in a dashboard:

1. Open the JReport Console page, select **New Report** > **Dashboard** in the Reports tab to create a dashboard.
2. Open the Components panel and browse to the folder where the two library components are published.
3. Drag and drop the two library components to the dashboard.

Component

Order ID	Order Date	Quantity	Unit Price
Asia-Pacific			
3130	7/21/2014	196	9.90
3344	7/27/2015	1638	9.90
3130	7/21/2014	656	11.15
3113	6/13/2014	1227	8.00
3148	8/13/2014	1227	8.00
3183	10/13/2014	1227	8.00
3218	12/13/2014	1227	8.00
3253	2/13/2015	1227	8.00
3288	4/13/2015	1227	8.00
3358	8/13/2015	1227	8.00
3323	6/6/2015	1227	8.00
3393	10/13/2015	1227	8.00
3428	12/13/2015	1227	8.00
3083	4/23/2014	1753	10.45
3118	6/23/2014	1753	10.45
3153	8/23/2014	1753	10.45
3188	10/23/2014	1753	10.45

Component

	Asia-Pacific	Europe, Middle	Latin America	North America	Total
3003				31,053.98	31,053.98
3004				20,971.97	20,971.97
3005				10,679.96	10,679.96
3006		41,797.91			41,797.91
3007		82,821.05			82,821.05
3008		207,433.29			207,433.29
3009				15,935.59	15,935.59
3010				29,372.80	29,372.80
3011				1,533.28	1,533.28
3012				11,769.18	11,769.18
3013				16,354.70	16,354.70
3015				27,639.08	27,639.08
3016				14,490.54	14,490.54
3017				6,767.05	6,767.05
3018	25,727.90				25,727.90
3020				6,669.75	6,669.75
3021				1,728.00	1,728.00
3022				14,209.02	14,209.02
3023				9,696.50	9,696.50
3024				13,760.02	13,760.02

4. Click the assigned region **Asia-Pacific** in the table in Component 1 to send out the message, then you can see how the crosstab in Component 2 is filtered after it receives the message.

Component

	Asia-Pacific	Total
3018	25,727.90	25,727.90
3033	36,718.90	36,718.90
3049	55,465.92	55,465.92
3064	25,727.90	25,727.90
3078	7,956.14	7,956.14
3083	16,486.96	16,486.96
3099	26,302.65	26,302.65
3113	55,769.58	55,769.58
3117	39,287.45	39,287.45
3118	16,486.96	16,486.96
3130	15,252.47	15,252.47
3134	26,302.65	26,302.65
3148	55,769.58	55,769.58
3152	39,287.45	39,287.45
3153	16,486.96	16,486.96
3169	26,302.65	26,302.65
3183	55,769.58	55,769.58
3187	39,287.45	39,287.45
3188	16,486.96	16,486.96
3204	26,302.65	26,302.65

5. Turn to Page 7 and click **Europe, Middle East, Africa** in the table, then you can see the records about this assigned region is displayed in the crosstab.

Component

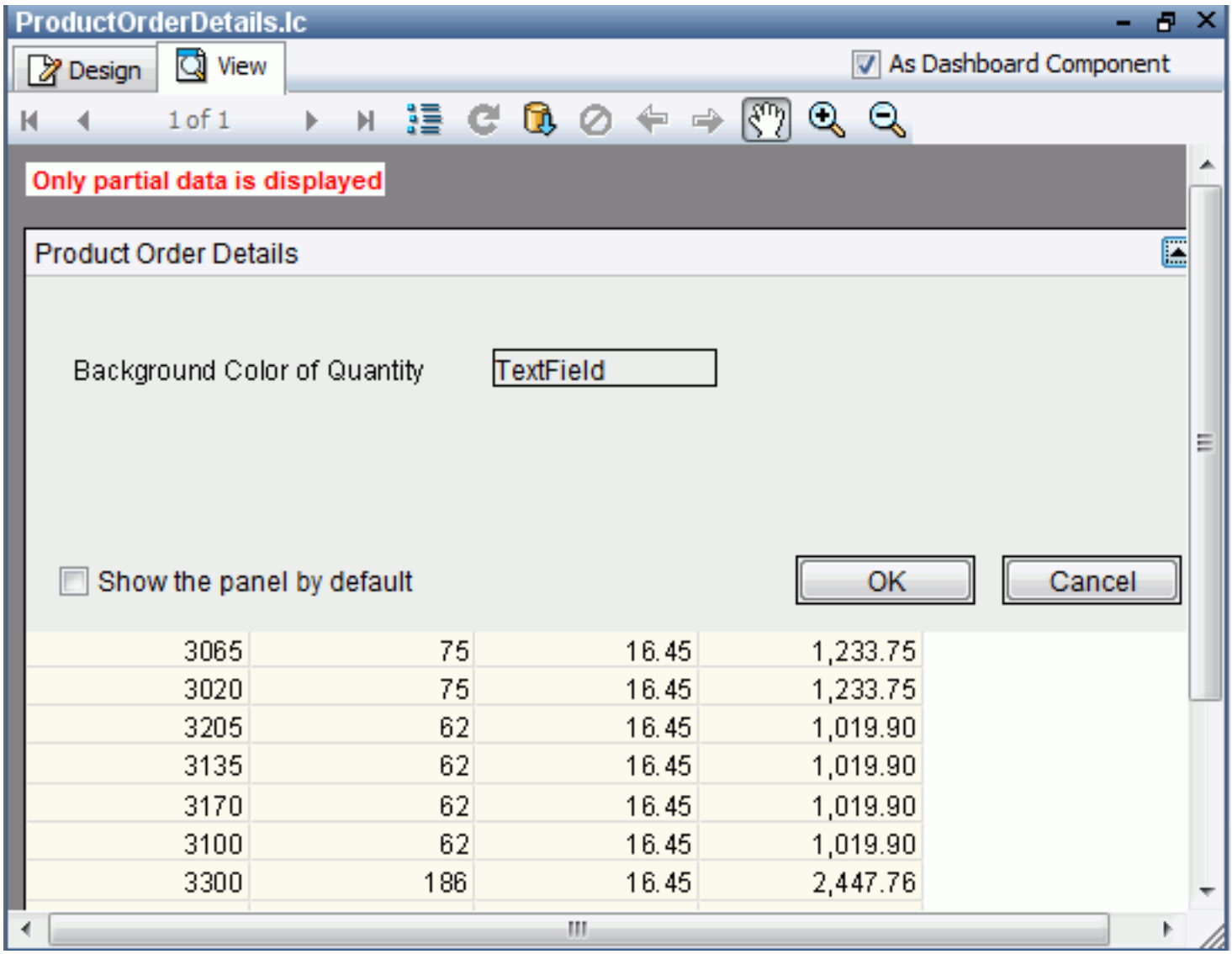
	Europe, Middl	Total
3006	41,797.91	41,797.91
3007	82,821.05	82,821.05
3008	207,433.29	207,433.29
3025	46,062.56	46,062.56
3027	47,933.49	47,933.49
3029	78,640.05	78,640.05
3030	7,200.00	7,200.00
3037	54,547.68	54,547.68
3053	41,573.07	41,573.07
3055	207,433.29	207,433.29
3060	102,250.80	102,250.80
3061	161,967.46	161,967.46
3070	6,852.61	6,852.61
3072	7,914.10	7,914.10
3074	9,243.21	9,243.21
3075	13,484.80	13,484.80
3080	19,021.05	19,021.05
3081	22,316.05	22,316.05
3088	17,812.02	17,812.02
3095	43,526.32	43,526.32

6. You can also click other assigned regions to see the records about each assigned region.

Previewing a library component

To preview a library component when designing it at any time, just click the **View** tab, which then causes JReport Engine to run the library component. However, in this way, you will find that only the result of the data component in the library component is displayed. The other parts like the configuration panel cannot be previewed. Therefore, JReport provides you with the option to preview the effect when the library component is published to JReport Server and inserted into a dashboard. To do this, check the **As Dashboard Component** checkbox on the top-right corner, then the title bar of the library component will be shown, and you can click on the title bar to display the configuration panel of the library component and preview the objects in it.

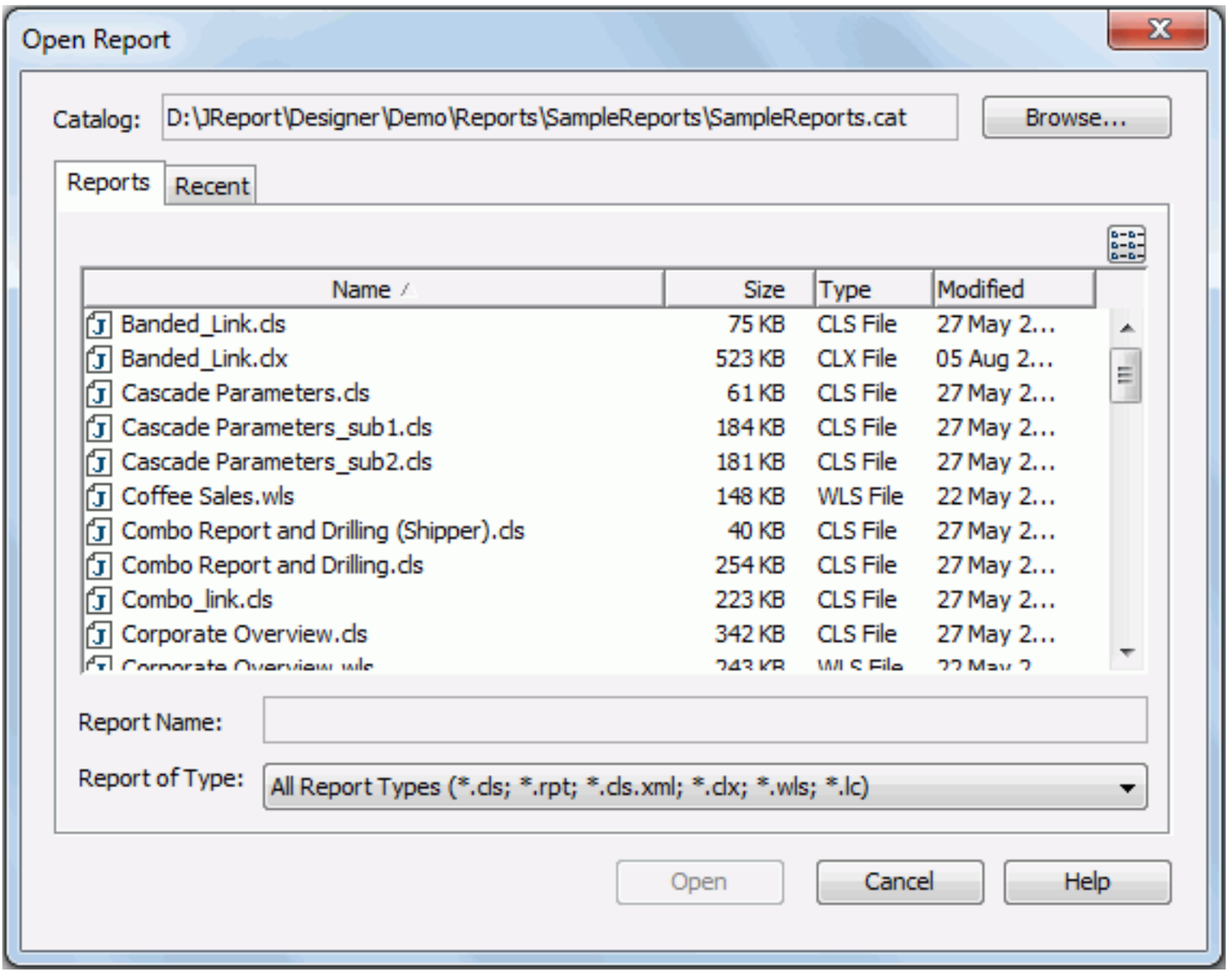
Moreover, when previewing a library component, if it is developed with National Language Support, you can specify the language in which you want the library component to display; if it contains parameters, you will be prompted to specify values for the parameters in the Enter Parameter Values dialog.



Opening a library component

To open a library component:

1. Click **File > Open** on the menu bar, or the **Open** button on the Standard toolbar. The Open Report dialog displays.



2. Browse to find the catalog that contains the library component you want to open.
3. In the Reports tab, select **Library Component (*.lc)** from the Report of Type drop-down list.
4. All the library components in the current catalog will be displayed. Select the desired library components.

   If you want to open library components that you have opened recently, switch to the **Recent** tab, and then select the library components you want to open.
5. Click **Open** to open the library components that you specified.

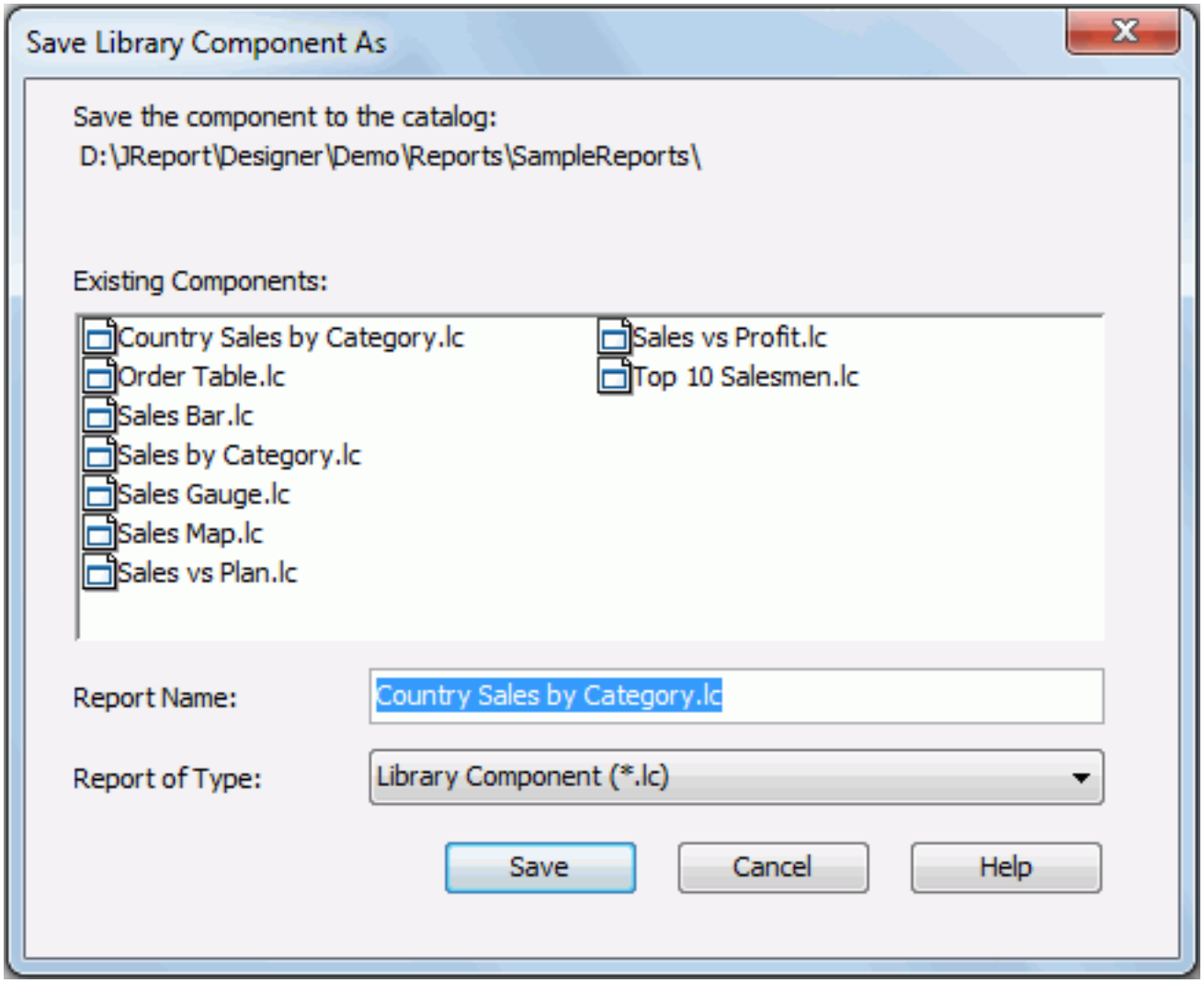
Tip: If you just want to access some recently open library components, click **File** > **Open Recent** and then select the library components one by one from the submenu.

Saving a library component

To save a library component, click **File** > **Save** on the menu bar, or the **Save** button on the Standard toolbar. This operation can also be accomplished by pressing the Ctrl + S keys. If the library component to be saved is new, the Save Library Component As dialog will appear for you to input a name.

To save a library component with a different name in the current directory:

1. Click **File** > **Save As** on the menu bar. The Save Library Component As dialog appears.



2. Type a new file name in the Report Name field.
3. The library component file type (*.lc) is the only default value in the Report of Type drop-down list.
4. Click **Save** to save the library component with the new name.

Report Bursting

In a large enterprise reporting deployment, it is important to handle both large amounts of data as well as a large number of users. Report bursting enables running a report once and distributing the report results to multiple recipients who each will receive a subset of the report results.

Bursting reports can be distributed to e-mail or FTP addresses, to disk, to the JReport versioning system, or to the security system members such as users, groups, and roles.

This chapter explains what a bursting report is and how to design a bursting report in JReport Designer. For details about how to submit a scheduled task which contains a bursting report and viewing the status of the bursting task and stopping it.

Pick a task from the following:

- What is a bursting report
- Designing a bursting report
- Built-in functions for bursting

What is a bursting report

A report applied with at least one bursting schema is called a bursting report. A bursting report can have multiple bursting schemas with each a unique schema name.

Bursting schema defines how data is split and who receives each subset of the split data. A valid bursting schema includes:

- Bursting key
- The dataset applied to the bursting key
- Bursting recipient
- Mapping relationship between bursting key and recipient

Notes:

- The bursting report always means the primary report. If both the primary report and its subreport are bursting reports, the system will only execute the bursting task according to the primary report, and the subreport will be processed as a normal report.
- If you load a report which was created in V8.2 containing only one bursting schema, JReport will create a default name for the schema. The default name is BurstingSchema.
- If the end user has applied security information or filters to the data and the system cannot obtain the bursting key, it will not send the result.

Bursting key

When running a bursting report, JReport will split the data according to a condition. The condition is the bursting key. The system will generate multiple results according to this key. For example, sending a credit card bill to every consumer according to the credit card ID. The credit card ID here is the bursting key.

Make sure the DB columns used for the bursting key are sortable in the DBMS, otherwise the bursting results may be unpredictable.

The following data types can be used for the bursting key:

- **Integer**: INT, SMALLINT, TINYINT, BIGINT
- **Float**: REAL, FLOAT, DECIMAL, NUMERIC
- **Character**: CHAR, NCHAR, VARCHAR, NVARCHAR
- **Date and Time**: DATETIME, SMALLDATETIME
- **Currency**: MONEY, SMALLMONEY

Bursting dataset

The bursting key must be created based on an existing dataset. The bursting key can be composed of a single DB column or multiple DB columns in the dataset. When multiple columns are selected, the system will split the data according to each unique combination of the columns.

Bursting recipient

The bursting task generates multiple results and sends them to different places. We define the places as the recipient. The recipient determines to which addresses or physical locations the report results will be sent.

In recipient, the following kinds of addresses are supported:

- Principals in JReport Server security system (user, role and group)

  You can specify a DB column that provides information to locate the users in the server security system.

  The report results can be sent to two places:

  - For user: Sending to e-mail addresses or private folders of the users defined in the DB column.
  - For role: Sending to e-mail addresses or private folders of all users belonging to the role.
  - For group: Sending to e-mail addresses or private folders of all users belonging to the group.
- E-mail: You can specify a DB column of e-mail addresses.
- Disk: You can specify a DB column that contains directories or full paths with file name. If it is a directory, the system will add the bursting result with the name generated by the system into the directory. If it is a full path, the system will generate the result with the specified file name and add it into the path.

  If the system finds that there is a file with the same name as the generated bursting result file, it will not send the result file into the path and will record a message into the log.
- FTP: You can specify a DB column of FTP addresses.
- Versioning system: You can specify a DB column that provides directories in the resource system of the JReport Server.

Default name for bursting result files

Sometimes you may not want to provide a file name for each bursting result when defining recipients. The bursting system will give it a name as generated by the system.

The default name format is: ReportName + "_" + BurstingKey + suffix (result format type). When there are multiple bursting key columns, connect each one by the character "_".

Converting to String

When a bursting key is of one of the following data types, it will be converted into String so as to make a valid result file name:

- Integer, Float, Character: Same as Java, these data types are transferred to String directly.
- Date and Time: All date and time formats will be transferred to the date format: yyyy-MM-dd hh:mm:ss.
- Currency: Currency will be transferred to the number without the currency mark ($ or others).

Name length

In the JReport Server resource system, the resource name only supports up to 64-character length. If a bursting result file name is longer than that, the system will trim it down automatically.

In order to avoid the same name in the same path, an index will be appended to the result name, for example: report1_USA_Maryland1.pdf, report1_USA_Maryland2.pdf.

Mapping relationship between bursting key and recipient

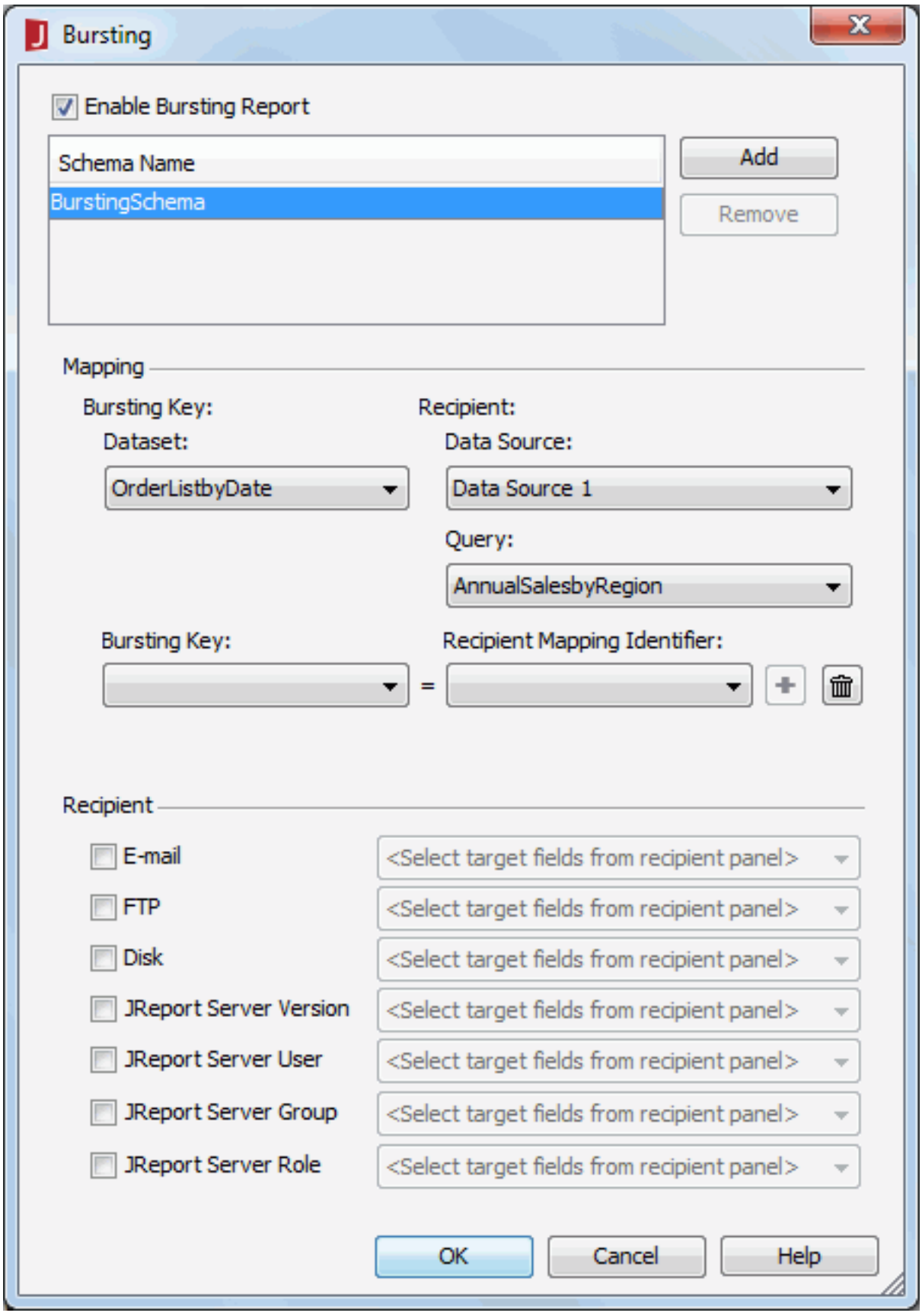
Relationships must be set up between bursting keys and bursting recipients in order to make clear which subset of report results will be sent to which addresses.

For each bursting key column, there must be only one unique recipient column to map to it. The data type of a recipient column must be consistent with the bursting key column it maps to. The user interface dialog only displays the matchable columns.

Designing a bursting report

To create a bursting report in JReport Designer, follow the steps below:

1. Design a report and then click **Report** > **Bursting** on the menu bar. The Bursting dialog appears.



2. Check the **Enable Bursting Report** option.
3. Click the **New** button next to the Schema Name box to add a bursting schema for the report. By double-clicking the schema name you can rename it as required.
4. In the Mapping panel, create the mapping relationship between the bursting key and the recipient according to your requirements.
5. In the Recipient panel, specify the recipient addresses.
6. If you want to create another schema for the report, repeat steps 3 - 5.
7. Click **OK** to apply the settings.
8. Save the report and publish the report to JReport Server.

Tip: When you finish designing a bursting report, you may want to check it with data. However, when you preview bursting report in JReport Designer, JReport runs the bursting report with all of the data, and if a large dataset is involved it may take a long time or run out of memory before completing. To avoid this from happening, you can limit the number of records by setting the Maximum Rows property of the query.

Reference: A sample report containing a bursting report tab named SaleStatForRegion.cls is provided in `<install_root>\Demo\Reports\SampleReports`.

Built-in functions for bursting

Two built-in functions are added in formulas for controlling bursting report result.

- **Boolean isRunBursting()**
  Returns true if the report is running based on bursting schema, else returns false.
- **String currentBurstingSchema()**
  Returns the names of the currently applied bursting schemas. When isRunBursting() returns false, the returned value for currentBurstingSchema() is NULL.

Note: Formulas referring isRunBursting() or currentBurstingSchema() are not allowed to be used in queries or datasets.

The following is an example of using the two functions.

Suppose that a report is first grouped by VP, and next by Manager as follows:

Group by VP,

Group by Manager,

detail

To distribute relative results to all VPs and managers, you can define two schemas as follows and apply them together in a task:

Bursting Schema	Bursting Key	Recipient
VP	VP	VPs
Manager	VP, Manager	Managers

Then write a formula:

```
if ( isRunBursting() && currentBurstingSchema() != "VP" )

return true

else

return false;
```

Next use the formula to control the Suppress property of the group headers and group footers in the VP group panel.

Then all results to the managers will hide the VP level.

National Language Support

JReport products implement a National Language Support (NLS) feature, which allows reports to be run in different languages.

In JReport Designer, you can use the two editors to edit NLS for reports: Data Mapping Editor and NLS Editor. With the Data Mapping Editor, you can create data mapping files to map data values for resources in a catalog, and the target language information of the resources will be stored in the data mapping files. Using the NLS Editor, you can translate the language-dependent text in a report or library component and the current catalog to any language you want, and modify the format and font properties for objects in the catalog and for objects used in the report or library component. Then when the report or dashboard that contains the library component runs, JReport Engine retrieves the language-dependent information according to the current locale in order to make the report result match the locale setting.

NLS can also be edited on JReport Server.

The following topics detail how to edit NLS at report level:

- NLS at application level
- NLS at report level

Note: NLS feature is not supported on shape map.

NLS at application level

All the JReport Designer UI text and message related files are packaged in the file resource_en_US.jar (if JReport Designer is Chinese version, the file name is resource_zh_CN.jar) in `<install_root>\lib`. To make JReport Designer displayed in your required language, you just need to translate the needed files in this package.

The following takes French as an example to explain how to make JReport Designer displayed in another language.

1. Backup the file resource_en_US.jar in `<install_root>\lib`, then rename it to **resource_fr_FR.jar**.

   The name of the file must follow the format below: resource_abbreviation of the language with lowercase_abbreviation of the country using the language with uppercase.jar. The language argument is a valid ISO Language Code as defined by ISO-639. You can find a full list of these codes at a number of sites, for example: http://www.loc.gov/standards/iso639-2/php/code_list.php. The locale argument is a valid ISO Country Code as defined by ISO-3166. You can find a full list of these codes at a number of sites, for example: http://www.chemie.fu-berlin.de/diverse/doc/ISO_3166.html.

2. Open the newly saved file resource_fr_FR.jar, and you can find it contains two folders, META-INF and resource.

3. In the resource folder, open the following files one by one and translate the property values in the files to French, then save and close the files:
   - JWColorChooser.properties
   - JWFileChooser.properties
   - JWMessage.properties
   - JWMisc.properties
   - JWNLSPropertyValue.properties
   - JWObject.properties
   - JWObjectToolTip.properties
   - JWPropertyGroup.properties
   - JWToolTip.properties

4. Open the folder images in the resource folder, translate all text that can be translated in the image files, and save them. The file names cannot be changed.

5. Copy the folder uidescriptors in the resource folder to a specified place (for example, the Desktop), then open it. The folder contains a lot of .xml files.

6. Right-click the file CatalogDoctorResource.xml in the uidescriptors folder and select **Edit** from the shortcut menu to open the file. The file contains some name properties. Translate their values to French, then save and close the file.

7. Translate the name property values in the following files as you do in the above step:
   - DataBrowser.xml
   - ReferenceEntityDialog.xml
   - MainMenu.xml
   - ToolBoxResource.xml
   - HelpLinkResource.xml

8. Translate the displayname property values in the following files as you do in step 6:
   - FormulaSpecialFieldsResource.xml
   - CrosstabFormulaFunctions.xml
   - FormulaFunctions.xml
   - FormulaOperators.xml
   - CTFormulaSpecialFieldsResource.xml
   - DynamicFormulaSpecialFieldsResource.xml

9. Translate the title, name, tooltip and text property values in the other .xml files except ComponentContainerRelationship.xml, GraphPanel.xml and the ones listed in step 7 and step 8 as you do in step 6.

10. After finishing the translation, copy the folder uidescriptors from the Desktop to the resource folder in resource_fr_FR.jar placed in `<install_root>\lib` to replace the original one.

11. Open the file **config.xml** in `<install_root>\bin`, and set the value of the property to **_fr_FR**. Make sure that JReport Designer is closed when you modify this file.

    The language name must follow the format below: abbreviation of the language with lowercase_abbreviation of the country using the language with uppercase.jar. If the name format is not correct, JReport Designer will start with the default language.

12. Right-click the file JReport.bat in `<install_root>\bin` and select **Edit** from the shortcut menu to open the file, then change the name of the language package resource_en_US.jar to **resource_fr_FR.jar**.

13. Save the file. Then when you run JReport Designer, the UI text will be displayed in the specified language.

NLS at report level

The NLS feature is supported for the language-dependent text in the label objects, DBFields, parameters, formulas, summaries, some special fields (Print Date, Fetch Date, Modified Date, Page N of M, Global Page N of M, group by field), table/view/synonym columns, imported SQL columns and so on. JReport Designer provides with two editors to edit NLS:

- **Data Mapping Editor**
  The Data Mapping Editor is used to map data values of resources in a catalog to a certain language, so that when you use the resources in reports or library components, you can choose to display their values in the mapped language.

- **NLS Editor**
  The NLS Editor is used to edit the language information of a report or library component and the current catalog and stores the information in a NLS property file.

Pick a task from below:

- Mapping data values to different languages
- Creating NLS property files
- Built-in functions for NLS
- Previewing an NLS report
- Localizing page navigation links in HTML report outputs
- Example of creating an NLS report

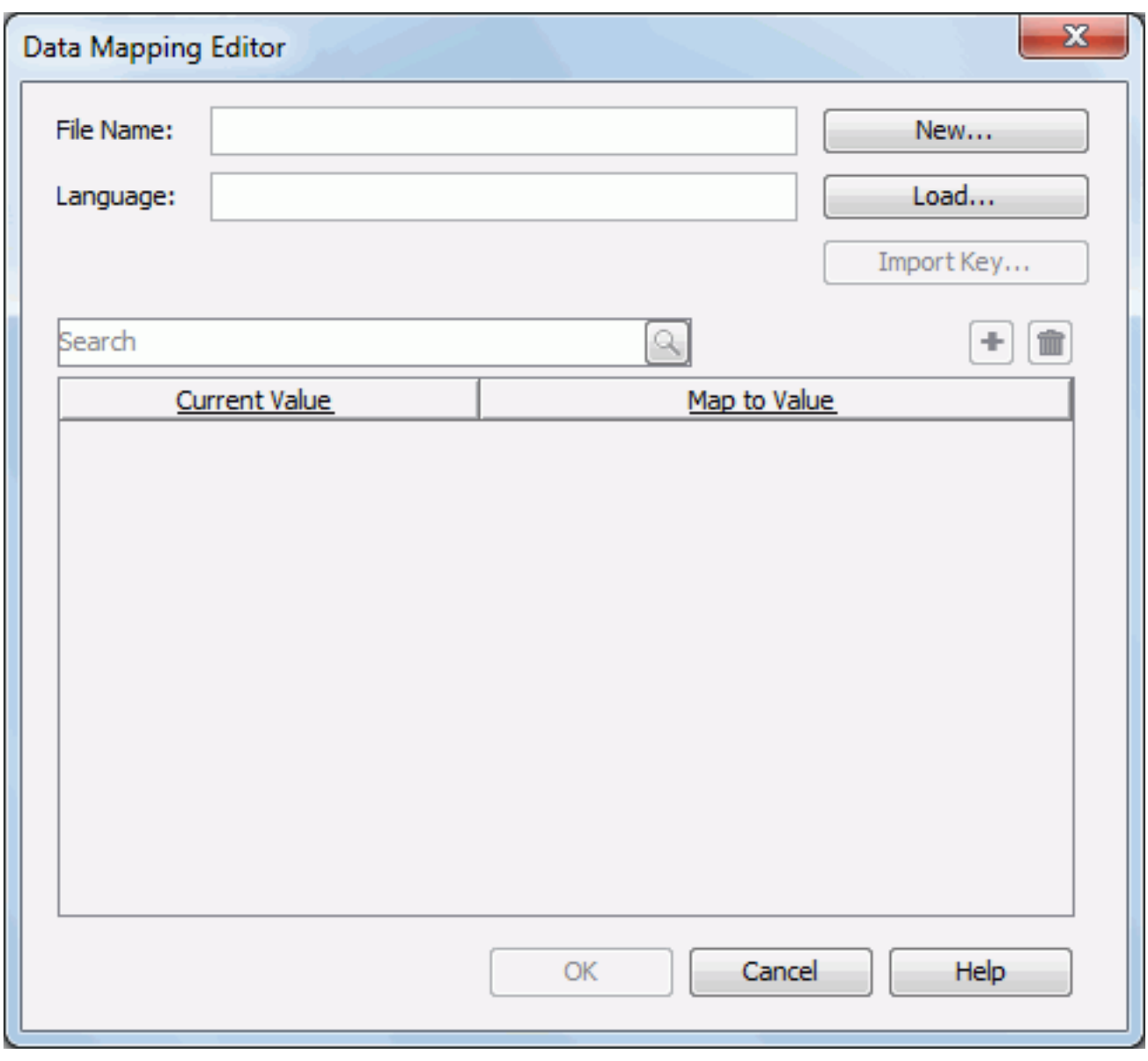
Mapping data values to different languages

With the Data Mapping Editor, you can create data mapping files to map data values of resources in a catalog to a specific language as you like. A data mapping file is a properties file used to store the target language information for resources in a catalog (DBFields, parameters, summaries, formulas, table/view/synonym columns, imported SQL columns and so on). One data mapping file stores one kind of language, and the name of the file is in the format "FileName_language_locale.properties". The abbreviations of the language and locale are used in the mapping file name. For example, the German mapping file name will be like this: FileName_de_DE.properties. The lower case "de" indicates the language is German, the upper case "DE" indicates the region is Germany. If all German regions can use the same file then the file name can be just FileName_de.properties.

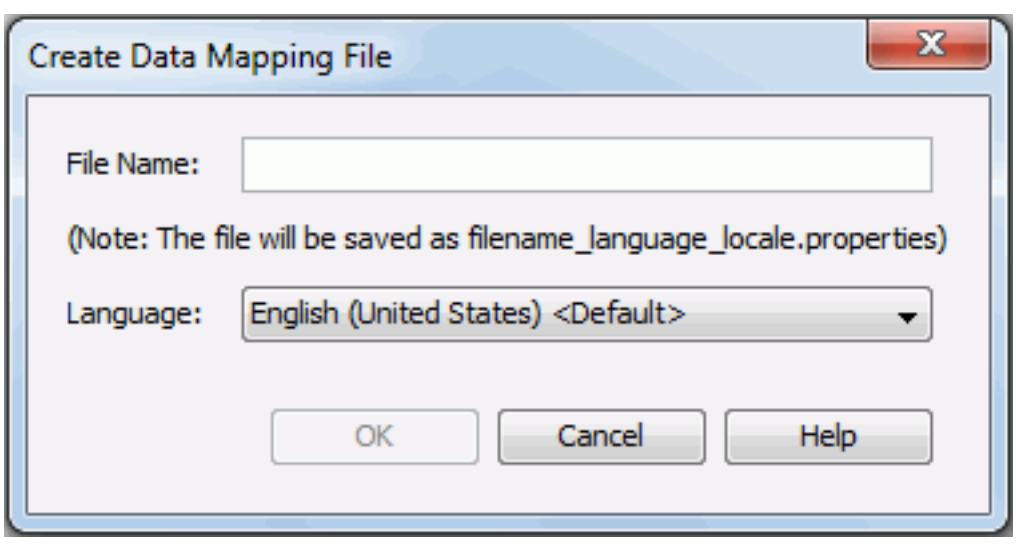
Creating a data mapping file

To create a data mapping file, that is to map the data values of resources in a catalog to a certain language, follow the steps below:

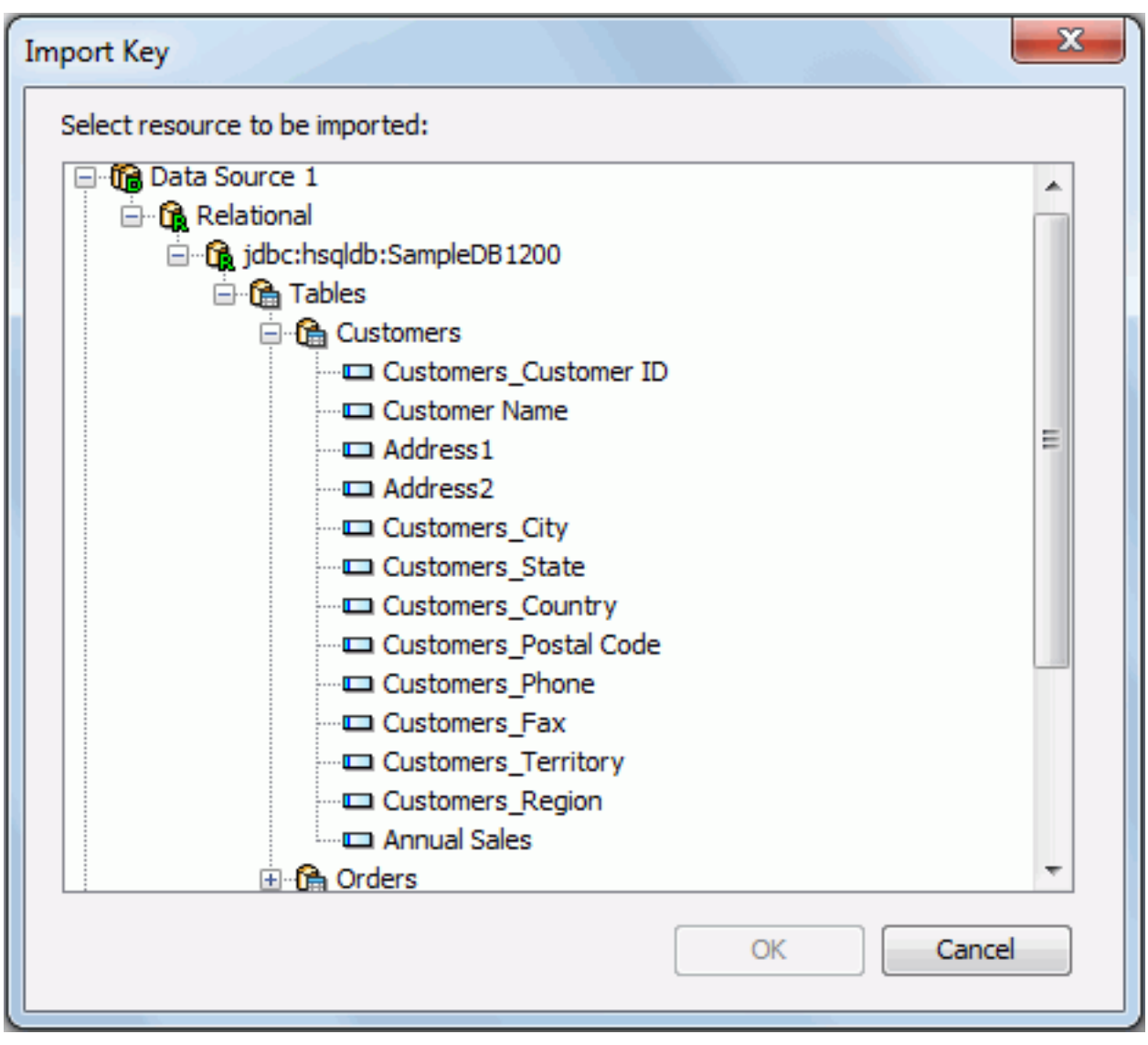
1. On the Catalog Browser toolbar, click the **Data Mapping Editor** button to display the Data Mapping Editor.



2. Click the **New** button and the Create Data Mapping File dialog appears.



3. In the File Name field, specify the name of the mapping file.
4. Click the **Language** drop-down list and select to which language the data values in the catalog will be mapped.
5. Click **OK**, and a property file will then be created in the folder where the catalog file is.
6. In the Data Mapping Editor, click the **Import Key** button to import the resources with the data values you want to map from the Import Key dialog.



7. Double-click cells in the Map to Value column to specify the mapping values for the imported resources according to the selected language. Use the quick search toolbar to locate the desired values in the value list if needed.
8. From the Import Key dialog, you can only import part of the resources in a catalog. If you want to map values of other resources like summaries, click to add a mapping line and specify the value in the Current Value column and Map to Value column accordingly.
9. When done, click **OK** to accept the changes and close the dialog.

Editing a data mapping file

To edit an existing data mapping file:

1. In the Data Mapping Editor, click the **Load** button.
2. In the Load Data Mapping File dialog, select the data mapping file you want to edit and click **Open**.
3. Information that has been saved in the file is then loaded in the Data Mapping Editor. You can import/add more values to map, or delete some by clicking .
4. When done, click **OK** to finish editing.

Applying a data mapping file

A data mapping file can be applied to the following objects:

- Objects in a catalog: table/view/synonym columns, aggregation objects, detail objects and group objects in business views, stored procedure columns, imported SQL columns, UDS columns, HDS columns, formulas, summaries and parameters.
- Objects in a report: DBFields, formulas, summaries, parameters and special fields.

To apply a data mapping file to objects in a catalog:

1. Make sure that the option Forbid editing data object properties in the Catalog category of the Options dialog is unchecked so that the object property values in the Catalog Browser can be changed.
2. Open the catalog and select the node that represents the object to which you want to apply the data mapping file.
3. In the Properties panel, set the data mapping file name (without the language and locale part) of the object as the value of the property Data Mapping File. For example, if the data mapping file name is Category_de_DE.properties, set the property value as Category.

To apply a data mapping file to objects in a report:

1. Open the report, then in Report Inspector, select the node that represents the object to which you want to apply the data mapping file.
2. In the Properties panel, set the data mapping file name (without the language and locale part) of the object as the value of the property Data Mapping File. For example, if the data mapping file name is Category_de_DE.properties, set the property value as Category.

   The data mapping file cannot take effect now because there is no NLS property file for the report. You have to create a corresponding NLS property file for the report using the NLS Editor. For details, see Creating NLS property files.

Note: When the same object is applied with a data mapping file both at the catalog level and report level, the file specified at the report level for the object has the higher priority. However, in the following two cases, if you want to apply a data mapping file to the object, you need to specify the file at the report level. The data mapping file specified at the catalog level cannot work under these two circumstances.

- When the object is used as group by field in a table or banded object and a special function or special group is applied to it.
- When the object is added as the category or series field in a chart and any text or mapping format is specified for it.

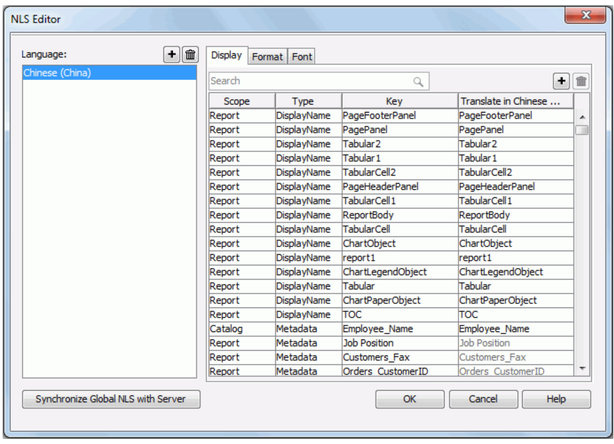
Creating NLS property files

The NLS Editor is used to create and edit NLS property files for reports and library components. With the NLS Editor, you can translate the language, and modify the format, font size and font face of the target language.

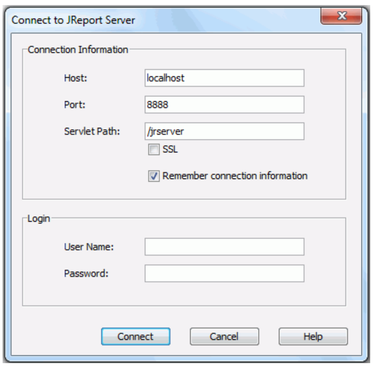
You can create NLS property files for a report/library component by using the report and catalog NLS resources, or by using NLS resources that are created on JReport Server and can be used globally for all reports in JReport.

To create NLS property files for a report:

1. Open the report, then click **Edit** > **NLS Editor** on the menu bar or click the **NLS Editor** button on the NLS Language toolbar. The NLS Editor appears.



2. Click above the Language box.
3. In the Add Language dialog, select the required languages in which you want the report to display, then click **OK**. The selected languages are now listed in the Language box of the NLS Editor.
4. When you open the NLS Editor for a report for the first time, by default, all the available report and catalog NLS resources are listed in the Display, Format and Font tabs of the NLS Editor. If you also want to use the global NLS resources created on JReport Server in the report, click **Synchronize Global NLS with Server**. The Connect to JReport Server dialog appears.



5. Type in the host and port of the remote JReport Server in the Host and Port text fields.
6. When using a standalone server, enter **/jrserver** in the Servlet Path text field or the context root of your web application if deployed as a war or ear fie such as **/jreport/jrserver**.
7. If the server has been integrated with another web server that supports SSL, select the **SSL** (Security Socket Layer) checkbox to create an SSL connection.
8. Check **Remember connection information** to enable JReport Designer to remember the connection information.
9. Provide the user name and password of the server. When the organization feature is enabled and when the user is an organization user, User Name should include the organization name. Use \ to separate the organization name and the user name, for example, org1\user1.
10. Click **Connect**. The global NLS resources defined for the selected languages are now displayed in the Display, Format and Font tabs together with the report and catalog NLS resources.
11. Select a target language from the Language box to edit NLS information for it.
12. In the Display tab, give the corresponding target language text for the keys in the Translate in <language> column. Use the quick search toolbar to locate the desired keys if needed.

    For any key, it may have three items with different scope: Global, Catalog and Report. If you want to use the global NLS definition for the key, leave its report and catalog scope items unchanged. If you edit the catalog scope item for the key, the definition can be used by all reports in the current catalog for the same language. To make the translation specific for the report, edit the report scope item.

    When you have edited some display text keys for certain language for a report, the next time you open the NLS Editor for the report to edit NLS for this language, only the edited keys are displayed by default.

    If some catalog scope display text keys are edited for a language, when you use the NLS Editor to edit NLS for another report in the same catalog for this language, by default only the edited catalog scope display text keys are listed for this language.

    To add more display text keys for a language, click above the key table, then in the Add Display dialog, specify the scope of the keys to be shown in the Available Display table, select the required keys and click **OK**.

    To remove any display text key, select it and click above the key table. You can select multiple keys and remove them at a time.
13. In the Format tab, provide the corresponding formats for the keys in the Format in <language> column for the target language. Use the quick search toolbar to locate the desired keys if needed.

    For any key, it may have three items with different scope: Global, Catalog and Report. If you want to use the global NLS definition for the key, leave its report and catalog scope items unchanged. If you edit the catalog scope item for the key, the definition can be used by all reports in the current catalog for the same language. To make the format specific for the report, edit the report scope item.

    When you have edited some format keys for certain language for a report, the next time you open the NLS Editor for the report to edit NLS for this language, only the edited keys are displayed by default.

    If some catalog scope format keys are edited for a language, when you use the NLS Editor to edit NLS for another report in the same catalog for this language, by default only the edited catalog scope format keys are listed for this language.

    To add more format keys for a language, click above the key table, then in the Add Format dialog, specify the scope of the keys to be shown in the Available Format table, select the required keys and click **OK**.

    To remove any format key, select it and click above the key table. You can select multiple keys and remove them at a time.

    Note: To edit the format of the special fields Page N of M or Global Page N of M, you should first edit its format too, for example, Page <N> of <M> or Global Page <N> of <M> in the Report Inspector, and then in the Format tab of the NLS Editor dialog, a new line Page <N> of <M> or Global Page <N> of <M> will be listed and you can translate Page, of, Global Page into the desired language. N and M must be formatted as <N> and <M> in order to avoid errors.
14. In the Font tab, give the font face and font size for the keys in the Font Face and Font Size columns for the target language. Use the quick search toolbar to locate the desired keys if needed.

    For any key, it may have three items with different scope: Global, Catalog and Report. If you want to use the global NLS definition for the key, leave its report and catalog scope items unchanged. If you edit the catalog scope item for the key, the definition can be used by all reports in the current catalog for the same language. To make the font properties specific for the report, edit the report scope item.

    When you have edited some font keys for certain language for a report, the next time you open the NLS Editor for the report to edit NLS for this language, only the edited keys are displayed by default.

    If some catalog scope font keys are edited for a language, when you use the NLS Editor to edit NLS for another report in the same catalog for this language, by default only the edited catalog scope font keys are listed for this language.

    To add more font keys for a language, click above the key table, then in the Add Font dialog, specify the scope of the keys to be shown in the Available Format table, select the required keys and click **OK**.

    To remove any format key, select it and click above the key table. You can select multiple keys and remove them at a time.
15. Select another language and edit NLS for it as shown above.
16. When done, click **OK** to close the dialog.

Notes:

- For the same key, its NLS definition can be specified at three level: global, catalog and report. Among them, the first has the lowest priority, and the third has the highest. Global NLS resources cannot be edited in JReport Designer.
- When the same key exists both in catalog NLS resources and report NLS resources and the translations at both levels are also the same, the one from report NLS resources will be automatically deleted after clicking OK in the NLS Editor.
- When you have added some languages to the NLS Editor, a set of NLS property files will be generated in the directory where the current catalog exists according to the languages you have selected. One file is for one language with the name ReportName_language_locale.properties. The abbreviations of the language and locale are used in the NLS property file name. For example, the German NLS property file name for Report1 will be like this: Report1_de_DE.properties.

Built-in functions for NLS

Two built-in functions are added in formulas for applying the NLS feature to the formula result.

- **Translate(String a)**
  This function searches for the NLS translation of the current field values in the bound data mapping file.
- **Translate(String a, String b)**
  This function accesses the corresponding data mapping file dynamically according to the language selected from the language drop-down list on the NLS language toolbar and searches for the NLS translation of String b in it.

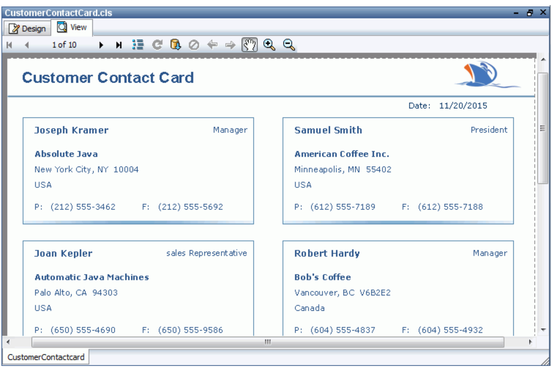
Notes:

- When either of the two functions is called in a formula, you are suggested not to use the formula to do further calculation unless you can make sure it is correct.
- When a formula references a parameter and the parameter's Allow Multiple Values property is true, the two functions are not available in the Formula Editor for editing the formula. In this case, if you want to apply NLS in the formula result, you can bind the data mapping file to the formula directly.
- The two functions can also be called in formulas on JReport Server, but they take effect only when the bound data mapping file exists on JReport Server. Moreover, only when the data mapping file is used in JReport Designer, can it be published to JReport Server together with the object it is bound to.

Example of calling the functions in a formula

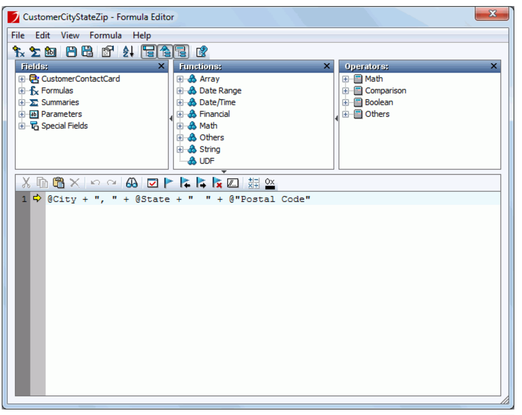
The following example is based on the report CustomerContactCard.cls in the SampleReports.cat catalog file.

1. In JReport Designer, click **File** > **Catalog Management** > **Open Catalog** on the menu bar to bring out the Open Catalog File dialog.
2. Browse to select the **SampleReports.cat** catalog file in `<install_root>\Demo\Reports\SampleReports`, click the **Open** button, and then click **Yes** in the Warning dialog.
3. Click **File** > **Open**. In the Open Report dialog, select **CustomerContactCard.cls** from the Reports tab and click the **Open** button to open the report.
4. Preview the report and it displays as follows:

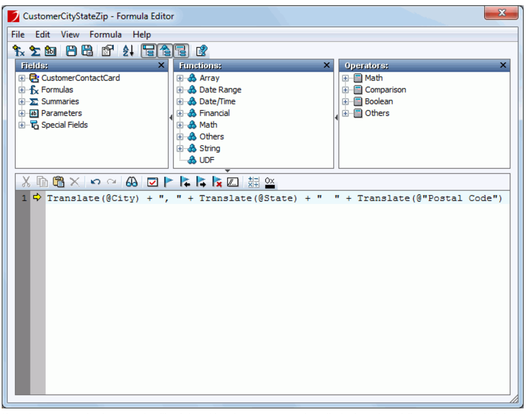


5. Click the **Design** tab to return to the design mode.
6. Click **Edit** > **NLS Editor**. The NLS Editor appears.
7. In the NLS Editor, click above the Language box, then in the Add Language dialog, select **Chinese (China)** from the Available Languages list and click **OK** to add it and return to the NLS Editor. Click **OK** in the NLS Editor.
8. Click on the Catalog Browser toolbar. The Data Mapping Editor apears.
9. Click the **New** button, then in the Create Data Mapping File dialog, input **DataMapping** in the File Name text field and select **Chinese (China)** from the Language drop-down list. Click **OK** to go back to the Data Mapping Editor.
10. Click the **Import Key** button, select the fields **City** and **State** in the Customers table of the displayed dialog and click **OK**.
11. Double-click cells in the Map to Value column to specify the mapping values for the imported resources in Chinese. Click **OK**.
12. Expand **Data Source 1** > **JDBC Connection** > **Tables** > **Customers** node in the Catalog Browser, select the **City** column, expand the Properties panel on the left, then select **DataMapping** from the Data Mapping File property value drop-down list.

    Make sure that the option Forbid editing data object properties in the Catalog category of the Options dialog is unchecked so that the object property values in the Catalog Browser can be changed.
13. Repeat the step above to set the Data Mapping File property of the column State in the same table to DataMapping. Click on the Catalog Browser toolbar to save the changes.
14. Right-click the formula field **CustomerCityStateZip** in the report and select **Edit Formula** from the shortcut menu to bring out the Formula Editor. The formula is written as below:



15. Edit the formula to **Translate(@City) + ", " + Translate(@State) + " " + Translate(@"Postal Code")**, then click on the toolbar to save the formula. Click **File** > **Close** to close the window.



If you have not set the data mapping file of the table columns, you can also edit the formula as follows to apply the NLS feature: Translate("DataMapping", @City)+ ", " +Translate("DataMapping", @State) + " " + Translate("DataMapping", @"Postal Code").

To make the Chinese characters displayed completely in the report, next we need to edit the font properties for the formula field CustomerCityStateZip.

16. Select the formula **CustomerCityStateZip** in the report, then set its Font Face property to **SimSun** and Font Size property to **12** in the Report Inspector. Click on the JReport Designer toolbar to save the report.
17. Select **Chinese (China)** from the language drop-down list on the NLS language toolbar.
18. Preview the report and you can find the formula result is displayed in Chinese:

Previewing an NLS report

After you develop a report with National Language Support, you can preview it by selecting the language in which you want the report to display.

To preview a report with NLS, you can choose either of the following two ways:

- **Method 1**

  1. In design mode, select a language from the language drop-down list

.
  2. Click the **View** tab to preview the report in the selected language. You can also click **View** > **Preview As** to preview the report in other format.

- **Method 2**

  1. In design mode, click the **View** tab.
  2. Select the language in which you want the report to display from the language drop-down list

.

Localizing page navigation links in HTML report outputs

When a report is exported to HTML format, the names of page navigation links in the report, such as First, Previous, Next, and Last, can be localized according to your requirements.

To localize the link names:

1. Create the sub directories in `<designer_install_root>` as follows: `<designer_install_root>\resources\report\languages\[language-locale]\properties`. For example, `C:\JReport\Designer\resources\report\languages\zh-cn\properties`.

   The language argument is a valid ISO Language Code as defined by ISO-639. You can find a full list of these codes at a number of sites, for example: http://www.loc.gov/standards/iso639-2/php/code_list.php. The locale argument is a valid ISO Country Code as defined by ISO-3166. You can find a full list of these codes at a number of sites, for example: http://www.chemie.fu-berlin.de/diverse/doc/ISO_3166.html.

2. Create a report.properties file in the properties directory.

3. Open the property file and copy the following contents to it:

```
# The following is the report properties file format that can localize the link names in HTML.
4000101=First
4000102=Prev
4000103=Next
4000104=Last
4000105=Back
4000106=Refresh
4000107=@CurrentPageNumber; of @TotalPageNumber;
```

4. Translate the text after = to the specified language.

   For the line "4000107=@CurrentPageNumber; of @TotalPageNumber;", you just need to translate "of" to the desired language. In the HTML outputs, @CurrentPageNumber will be replaced by the current page number, and @TotalPageNumber by the report total page number.

5. Save the property file with UTF-8 encoding.

6. Copy the property file to the `<jdk_install_root>\bin` directory.

7. Convert the contents in the property file into Unicode using native2ascii.exe in `<jdk_install_root>\bin` by running the following command:

   ```
   C:\jdk1.6.0_17\bin>native2ascii -encoding utf-8 report.properties > newreport.properties
   ```

   > **Note:** When you convert your property file to the same directory as the original one, you need to give it a new name instead of replacing the original in order to avoid problems.

8. Delete report.properties in `<designer_install_root>\resources\report\languages\[language-locale]\properties` and copy newreport.properties in `<jdk_install_root>\bin` to it, and then name the property file back to report.properties.

9. Start JReport Designer and open a multi-page report, select the language defined for the property file from the language drop-down list

on the NLS toolbar.

10. Click **File** > **Export** > **HTML** to export the report to HTML, and then in the exported HTML results, you will find that the page navigation links are displayed in the language you specified.

Styles

JReport style is one of the most important concepts in both JReport Designer and JReport Server. A style in JReport is a set of object attributes that you can assign to components in your report to alter their appearance and characteristics. When you apply a style, you apply a whole group of property entries at once. For example, you may want to format the headings of a report to make them stand out. Instead of formatting them step by step as 12pt, Arial, and center-aligned with a blue background, you can simply achieve the same goal in one step by applying a predefined style that contains the style properties you want.

JReport supports two kinds of styles, XSD style and CSS style. This chapter presents their usages respectively.

Pick a task from the following:

- XSD styles
- CSS style
- Applying styles
- Inheriting the style from a data container

Example of creating an NLS report

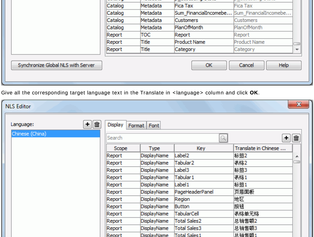
This example illustrates how to make the report Coffee Sales.wls in the SampleReports.cat catalog file in `<install_root>\Demo\Reports\SampleReports` displayed in Chinese. It contains the following tasks:

- Task 1: Create an NLS property file
- Task 2: Create a data mapping file
- Task 3: Publish the report to JReport Server
- Task 4: Apply NLS to the report at runtime

Task 1: Create an NLS property file

1. Open the report **Coffee Sales.wls** in the catalog file SampleReports.cat.
2. Select the **FilterControl** node representing the filter control Product Type in the Report Inspector and set its Map Name property to **true**. Select the nodes FilterControl1 and FilterControl2 representing the filter controls Category and Product Name respectively in the Report Inspector, you can find that their Map Name property is false.

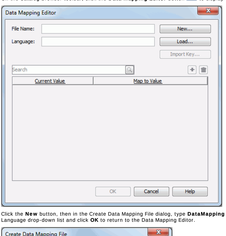
   If the Map Name property of a filter control is set to true, its title, namely, the name shown in the text box of the Text property is the name shown in the Catalog Browser and cannot be changed, whose type in the NLS Editor will be Metadata. Otherwise, its title shown in the text box of the Text property is editable, and its type in the NLS Editor will be Title.
3. Click **Edit** > **NLS Editor** to open the NLS Editor, then click the **Add** button above the Language box.
4. In the Add Language dialog, select **Chinese (China)** from the Available Languages list and click **OK** to add it and return to the NLS Editor.
5. In the Display tab, the scope and types of the objects in the report and items need to be translated are listed in the Scope, Type and Key columns. You can find that only the type of the filter controls Category and Product Name is Title.



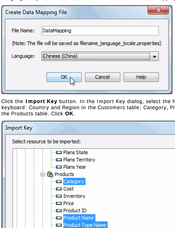
6. Give all the corresponding target language text in the Translate in <language> column and click **OK**.

Task 2: Create a data mapping file

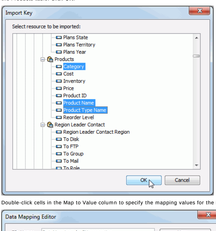
1. On the Catalog Browser toolbar, click the **Data Mapping Editor** button to display the Data Mapping Editor.



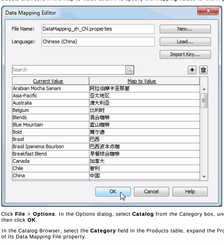
2. Click the **New** button, then in the Create Data Mapping File dialog, type **DataMapping** in the File Name text box, select **Chinese (China)** from the Language drop-down list and click **OK** to return to the Data Mapping Editor.



3. Click the **Import Key** button. In the Import Key dialog, select the following fields used in the report one by one by pressing the Ctrl button on the keyboard: Country and Region in the Customers table, Category, Product Name and Product Type Name (the Product Type field used in the report) in the Products table. Click **OK**.



4. Double-click cells in the Map to Value column to specify the mapping values for the imported fields to Chinese. Click **OK**.



5. Click **File** > **Options**. In the Options dialog, select **Catalog** from the Category box, uncheck the option **Forbid editing data object properties**, and click **OK**.
6. In the Catalog Browser, select the **Category** field in the Products table, expand the Properties panel on the left, then select **DataMapping** as the value of its Data Mapping File property.
7. Repeat the step above to set the Data Mapping File property of the fields Product Name and Product Type Name to DataMapping respectively. Click on the Catalog Browser toolbar to save the changes.
8. In the design area, select the chart legend and the Region field in the crosstab and set their Data Mapping File property to **DataMapping** in the Report Inspector respectively.

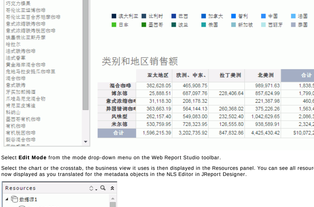
   You may notice that the Data Mapping File property can be specified in both the Catalog Browser and the Report Inspector. In JReport, you can specify the data mapping file of a field used in a report both at the catalog level and report level. When the property is specified at both levels, the data mapping file set in the Report Inspector has the higher priority.
9. Save the report.

Task 3: Publish the report to JReport Server

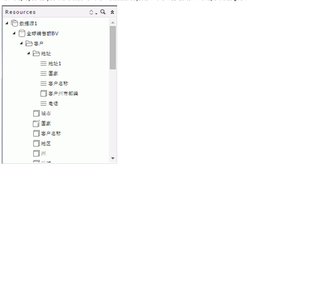
Now the NLS property file and corresponding data mapping file are created and applied to the report Coffee Sales.wls. Publish the report and the catalog it uses to JReport Server. For details, refer to Publishing resources remotely.

Task 4: Apply NLS to the report at runtime

1. Open a web browser and set the URL to http://localhost:8888, type **admin** in the User Name and Password text boxes respectively in the Welcome page, and click **Login** to open the JReport Console page.
2. Click **Profile** > **Customize Server Preferences** on the task bar.
3. Switch to the **Advanced** tab, check the **Yes** checkbox of the Enable NLS option, and select **Chinese (China)** from the Default Language drop-down list. Click **OK** in this dialog and then Click **OK** in the prompt message dialog.



4. Find the report published just now and click to run it in Web Report Studio. The report is displayed in Chinese.



5. Select **Edit Mode** from the mode drop-down menu on the Web Report Studio toolbar.
6. Select the chart or the crosstab, the business view it uses is then displayed in the Resources panel. You can see all resources in the resource tree are now displayed as you translated for the metadata objects in the NLS Editor in JReport Designer.

Publishing resources

JReport Designer, as a report design tool, allows you to publish your resources from JReport Designer to your local computer or to JReport Server.

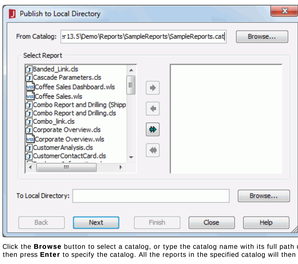
- Publishing resources locally
- Publishing resources remotely

Publishing resources locally

JReport Designer supports publishing resources locally. That is, publishing resources to any directory on your computer or across your intranet.

To do this, follow the steps below:

1. In JReport Designer, click **File** > **Publish and Download** > **Publish to Local Directory**. The Publish to Local Directory dialog appears.



2. Click the **Browse** button to select a catalog, or type the catalog name with its full path directly into the From Catalog text field, and then press **Enter** to specify the catalog. All the reports in the specified catalog will then be displayed in the Select Report box.
3. Select the reports that you want to publish from the Select Report box and click the arrow button to add them to the box on the right, or directly drag the reports you want to publish from the Select Report box to the box on the right. If no report is selected here, only the catalog will be published.
4. Specify a directory for the catalog and reports that are to be published.

   Click the **Browse** button to the right of the To Local Directory text field, and locate a directory in the Get Directory dialog. Or you can type a directory in the text box directly. If you type in a directory which does not exist, JReport Designer will automatically create it for you.
5. Click **Next** to view or modify properties of the data source connection.

   First, select a data source from the Catalog Data Source drop-down list, and then the connection properties of the selected data source will be displayed. If the selected data source is built on a JDBC connection, you can click the **Modify** button to modify properties of the data source connection in the Get JDBC Connection Information dialog.
6. Click the **Finish** button to publish the catalog and reports.

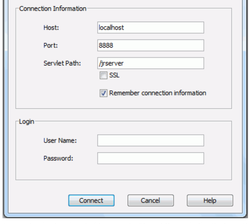
Publishing resources remotely

This section introduces how to publish different resources to JReport Server.

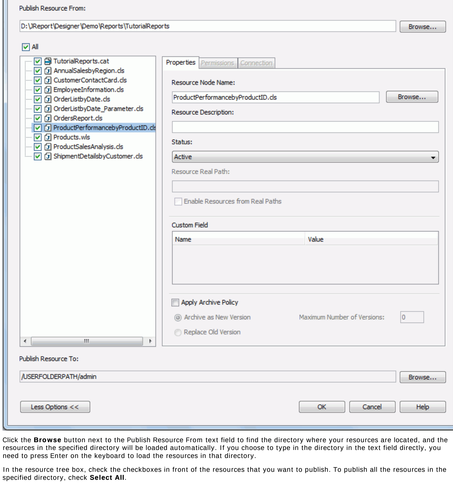
Publishing reports and library components

1. In JReport Designer:
   - Click **File** > **Publish and Download** > **Publish to Server** > **Publish Report to Server** if you want to publish reports.
   - Click **File** > **Publish and Download** > **Publish to Server** > **Publish Component to Server** if you want to publish library components.

   The Connect to JReport Server dialog appears.



2. Type in the host and port of the remote JReport Server in the Host and Port text fields.
3. When using a standalone server, enter **/jrserver** in the Servlet Path text field or the context root of your web application if deployed as a war or ear file such as //report//jrserver.
4. If the server has been integrated with another web server that supports SSL, select the **SSL** (Security Socket Layer) checkbox to create an SSL connection.
5. Check **Remember connection information** to enable JReport Designer to remember the connection information.
6. Provide the user name and password of the server. When the organization feature is enabled and when the user is an organization user, User Name should include the organization name. Use \ to separate the organization name and the user name, for example, org1\user1.
7. Click **Connect**. The Publish to JReport Server dialog appears.



8. Click the **Browse** button next to the Publish Resource From text field to find the directory where your resources are located, and the resources in the specified directory will be loaded automatically. If you choose to type in the directory in the text field directly, you need to press Enter on the keyboard to load the resources in that directory.
9. In the resource tree box, check the checkboxes in front of the resources that you want to publish. To publish all the resources in the specified directory, check **Select All**.
10. In the Publish Resource To text field, click the **Browse** button to specify the folder on the server where the resources are to be published.
11. Select a to-be-published resource in the resource tree box and then click the **More Options** button to set properties for the resource.
12. In the Properties tab,
    a. In the Resource Node Name text field, specify a name to be displayed in the server resource tree for the resource. You can type a name in the text field or click the Browse button to select a resource name from the Select Resource Name dialog, where all resource names of the same type as the selected resource, which are contained in the Publish Resource To path, are displayed.
    b. In the Resource Description text box, type a brief description to describe the resource.
    c. If a report is selected, specify the status of the report by selecting the required item from the Status drop-down list.
    d. If a folder is selected, type a local disk path in the Resource Real Path text field as the real path of the folder. If you would like to map the real path resources into the new node created for the folder in the server resource tree so that the server will always be able to get the resources and updates from the real path, check the option **Enable Resources from Real Paths**.
    e. Specify custom field values for the resource if there are custom fields enabled on JReport Server.
    f. Check **Apply Archive Policy** to apply an archive policy to the resource.
13. In the Permissions tab, check the **Enable Setting Permissions** option to specify permissions for the selected resource (this tab is available only when you have specified to publish the resources to the Public Reports folder on the server).
14. If the selected resource is a catalog, the Connection tab will be activated. You can view and modify properties of the selected data source connection as required.

    First, select a data source from the Catalog Data Source drop-down list, and then the connection properties of the selected data source will be displayed. If the selected data source is built on a JDBC connection, you can click the **Modify** button to modify properties of the data source connection in the Get JDBC Connection Information dialog.
15. Repeat steps 10 - 12 to set properties for other resources as required.
16. Click **OK** to publish the resources. If the reports or library components use special fonts, style groups, geographic information, or customized control files, the Publish Additional Resources dialog appears for you to specify them.

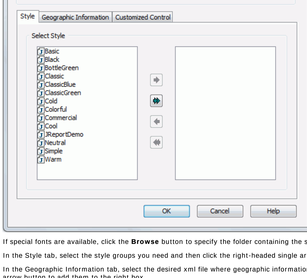
Notes:

- If you have made some changes to the to-be-published resources, but haven't saved them, before publishing starts, JReport will save the changes automatically to ensure the resources published to the server are the latest.
- After resources have been published to JReport Server, you may find that the publish time shown on the server UI is different from the client time. That is because the publish time takes the server time instead of the client time.

Publishing additional resources

Additional resources used by reports and library components can also be published to JReport Server. The resources include special fonts, style groups, geographic information, and customized control files.

1. On the File menu, click **Publish and Download** > **Publish to Server** > **Publish Other Files to JReport Server**.
2. In the Connect to JReport Server dialog, specify the information to connect to a started JReport Server. The Publish Additional Resources dialog is displayed.



3. If special fonts are available, click the **Browse** button to specify the folder containing the special fonts.
4. In the Style tab, select the style groups you need and then click the right-headed single arrow button to add them to the right box.
5. In the Geographic Information tab, select the desired xml file where geographic information is stored and click the right-headed single arrow button to add them to the right box.
6. In the Customized Control tab, select the customized control files you need and click the right-headed single arrow button to add them to the right box.
7. Click **OK** to publish the resources.

Publishing and Downloading Resources

JReport Designer is a 100% Java swing based report design tool. Developed page reports, web reports, library components, and catalogs can be published to run in distributed environments. You can publish your resources from JReport Designer to your local computer or to JReport Server. Meanwhile, resources managed on JReport Sever can also be downloaded to a specified folder on your local computer, and further modified in JReport Designer.

The following information is available:

- Publishing resources
- Downloading resources from JReport Server

To Applet

JReport Designer can collaborate with a web server by exporting a report as an applet to a server. After publishing it to the server, you can then retrieve it by accessing the server.

Exporting the report results as an Applet to a server

To export the results of a report as an applet to a server, follow the steps below:

1. In JReport Designer, open the web report or page report that contains the report tab you want to export.
2. When a page report is opened, on the report tab bar, click the tab that represents the report tab.
3. Click **File** > **Export To** > **Applet**.
4. In the Applet Setting dialog, key in the server name and port number in the Server Name and Port text boxes.
5. Check **Run Linked Report** if you want to include the linked reports in the exported applet result, and then click **OK**.
6. In the Export to Applet dialog, specify the directory and file name respectively.

   For example, browse to the location where you want to save the file, type exapplet.htm as the file name, and then click **Save**. The file exapplet.htm and the report result exapplet.rst will then be generated.

Viewing the exported result file

Before viewing the exported result file of JReport using a web browser, you first need to make sure that your web server supports applets. Then,

1. Copy the file viewer12.jar in `<server_install_root>\lib` to the document root directory of the web server.
2. Copy the result file `*.rst` and `*.html` generated when exporting to any directory of the web server. For example, copy A.rst and A.html into `<webserver_install_root>\test`.
3. Edit A.html (with a text editor) to make sure that all paths are valid. For example, change:
   - codebase="/" (which refers to the location of viewer12.jar)
   - value="/test/A.rst" (since we copied A.rst in `<webserver_install_root>\test`.)
   - width=800 height=600
4. Run the web browser with the web server to view the results.

Notes:

- Before you can view the results in a web browser, make sure you have enabled Additional Unsigned Permissions for the web browser and NPAPI is enabled for Google Chrome if you use this web browser. To do this, you need to type **chrome://flags/** in the address bar of the web browser, then scroll down and set Enable NPAPI Mac, Windows to **Enable**.
- JReport Server also supports requesting the report result in an applet from a web browser.

To fax

JReport Designer provides the feature of exporting the report results to fax either via a fax machine or a fax server. If you require this feature to be enabled, you will first need to configure your running environment.

Download the files listed in the tables below from our website (http://www.jinfonet.com/download/third party tool/JavaCommAPIV2 Win32.zip or http://www.jinfonet.com/download/third party tool/JavaCommAPIV2 Solaris.zip), and put them in the specified locations:

- For Windows

File Name	Location
comm.jar	<Java_install_root>\jre\lib\ext
javax.comm.properties	<Java_install_root>\jre\lib
Win32Com.dll	<Java_install_root>\jre\bin

- For Solaris

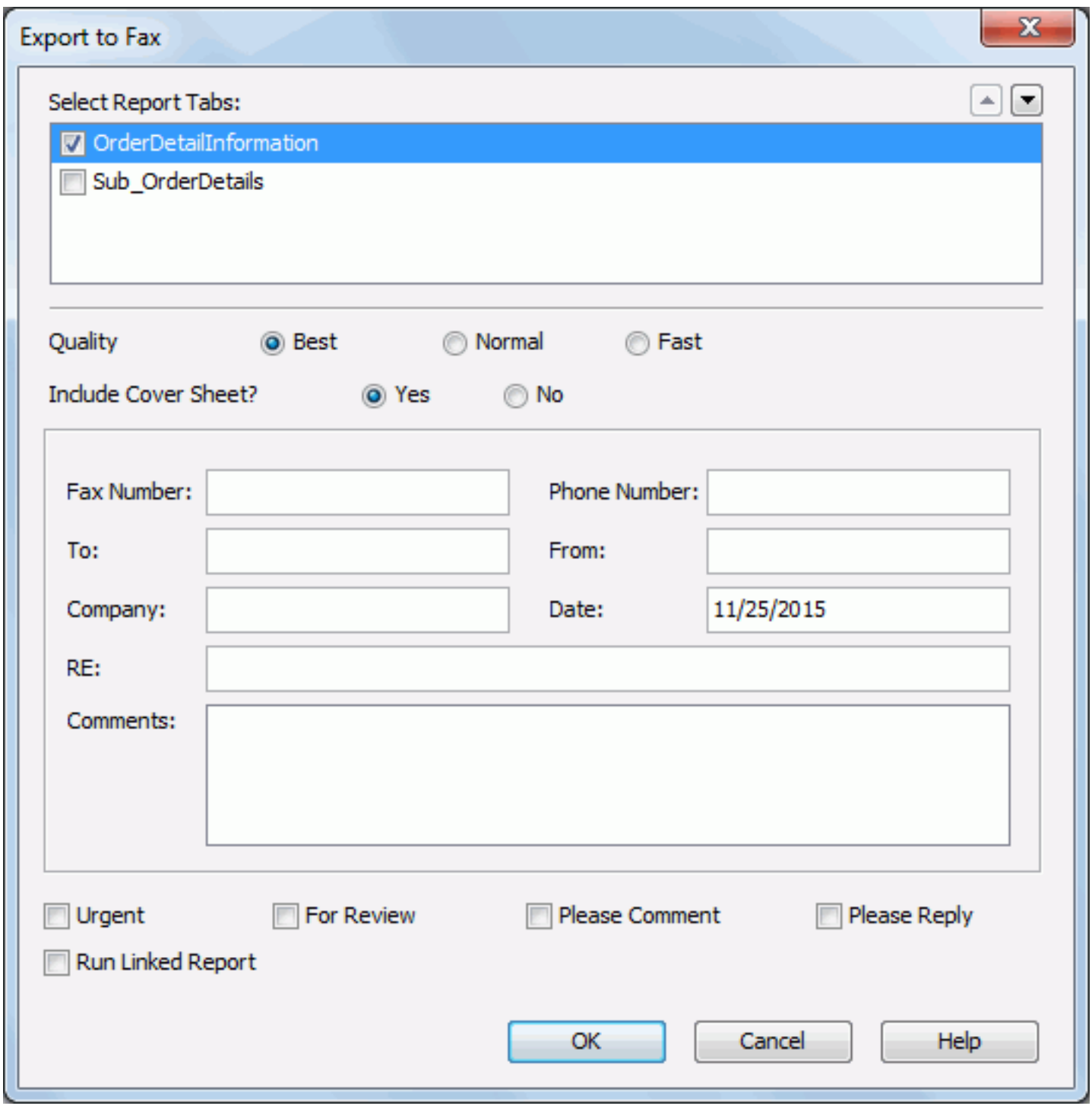
File Name	Location
comm.jar	<Java_install_root>/jre/lib/ext
javax.comm.properties	<Java_install_root>/jre/lib
libSolarisSerialParallel.so	LD_LIBRARY_PATH

Meanwhile, before you can export report results to fax, you need to have settings of the fax machine or fax server configured. Otherwise a warning message will be displayed when you try to export to fax. To do this:

1. In JReport Designer, click **File** > **Options** on the menu bar.
2. In the Options dialog, select the **Export to** category.
3. In the Fax tab, specify via which you want to fax the report results: fax machine or fax server.
4. Configure settings of the fax machine/server as required (For details about the settings, refer to Export to - Fax).
5. Click **OK** to accept the changes.

Now, you can export your report results to fax as follows:

1. Open the report you want to export.
2. Click **File** > **Export To** > **Fax** on the menu bar.
3. In the Export to Fax dialog, specify the settings as required.

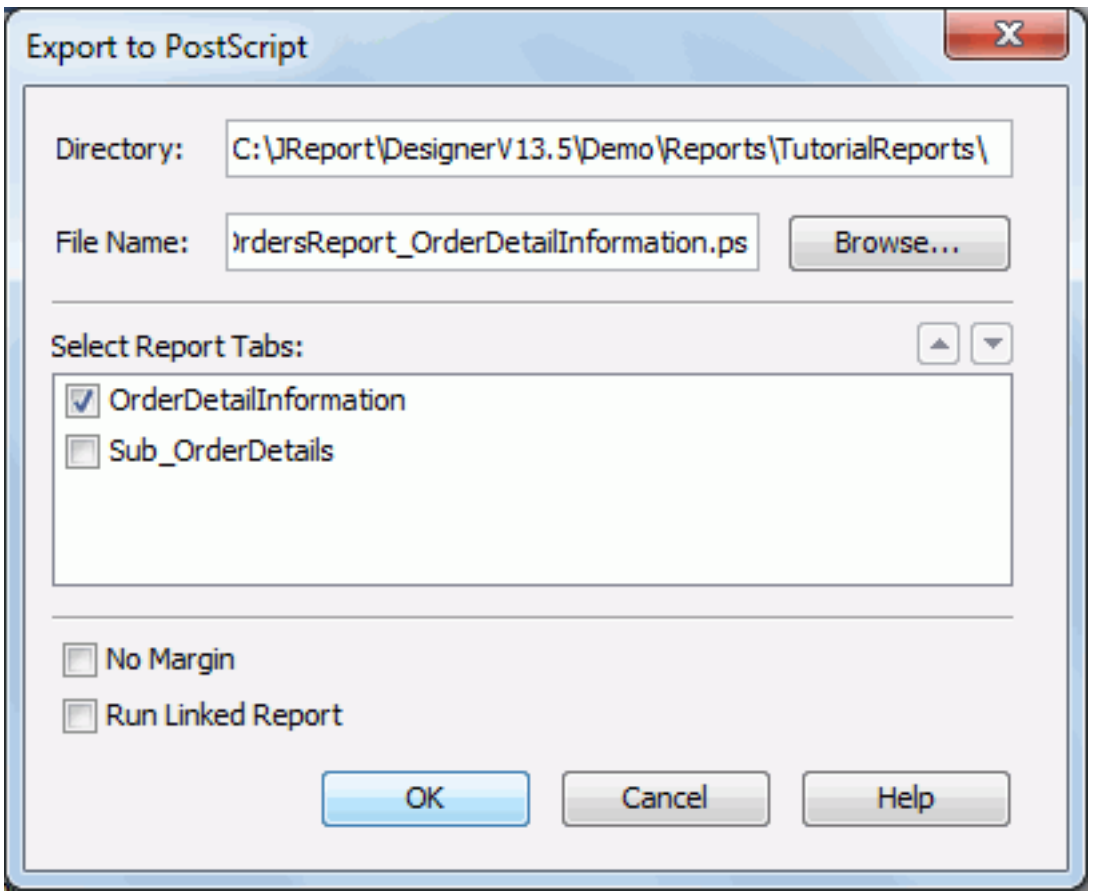


4. When done, click **OK** to generate the file.

To PostScript

To export the results of a report to a PostScript file, follow the steps below:

1. In JReport Designer, open the report you want to export.
2. Click **File** > **Export To** > **PostScript**. The Export to PostScript dialog appears.



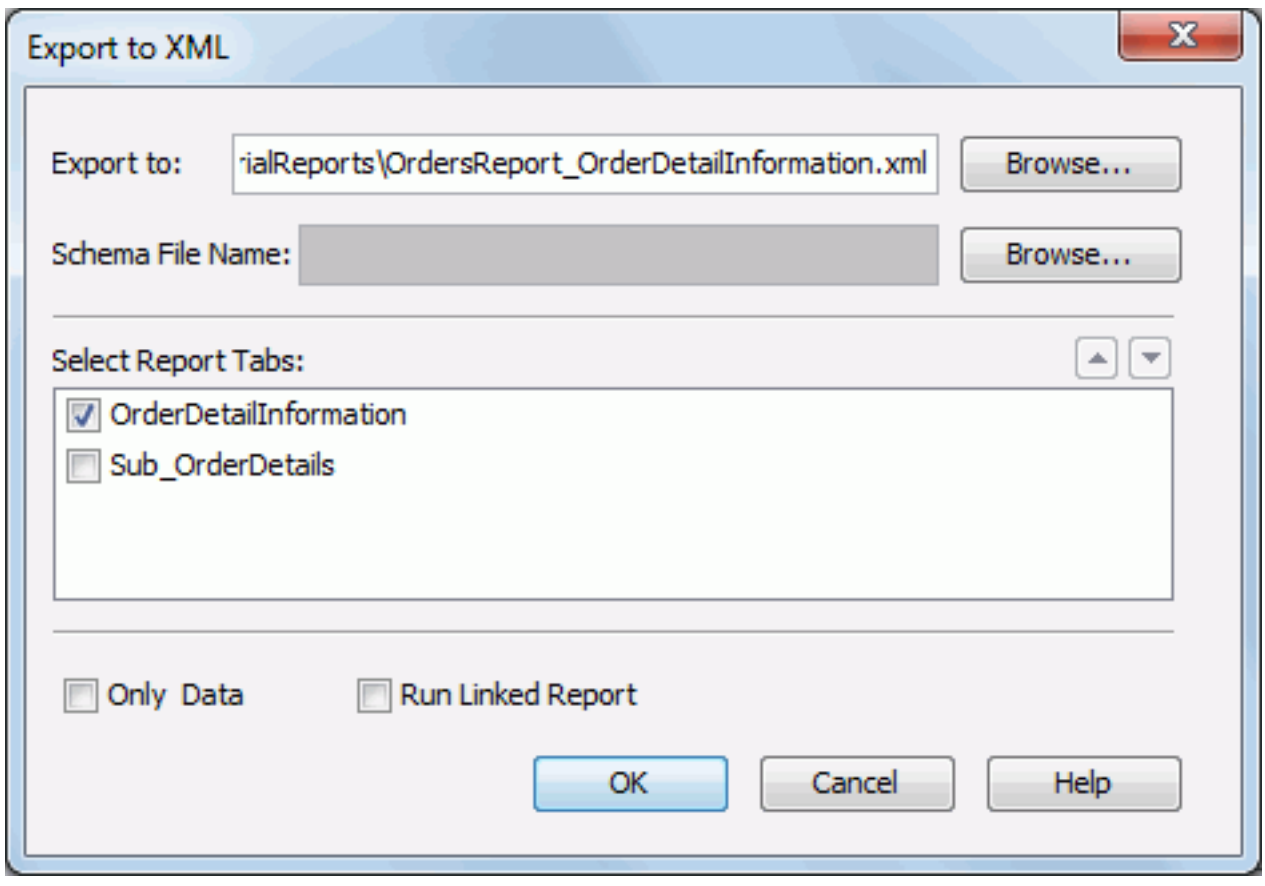
3. Specify the directory where the export PostScript file will be saved. If the directory does not exist, an error message will be produced.
4. Enter the name for the exported file.
5. If you are exporting a page report, in the Select Report Tabs box, select the report tabs in the page report you want to export. The selected report tabs will be exported in the list order. You can click ▲ or ▼ to change the order or the report tabs.
6. Check **No Margin** if you want to remove the margins that have been set to the report at design time.
7. Check **Run Linked Report** if you want to include the linked reports in the exported PostScript result.
8. When done, click **OK** to perform the export operation.

Note: When you export a report to a PostScript file, some objects in this report may be located imprecisely. You can print your report with PDF in order to get the accurate printed results. Therefore, you are recommended to export report to PDF rather than to PostScript.

To XML

To export the results of a report to an XML format file, follow the steps below:

1. In JReport Designer, open the report that you want to export.
2. Click **File** > **Export To** > **XML**. The Export to XML dialog appears.



3. Specify the destination of the exported file in the Export to field box.
4. In the Schema File Name field, specify an XSD file to export the web report or report tab to an XML file based on the specified XSD file.
5. If you are exporting a page report, in the Select Report Tabs box, select the report tabs in the page report you want to export. The selected report tabs will be exported in the list order. You can click or to change the order or the report tabs.
6. Check the **Only Data** checkbox to only export data of the web report or report tab.
7. Check **Run Linked Report** if you want to include the linked reports in the exported XML result.
8. When done, click the **OK** button, and a file with the extension .xml will be generated.

The following are some notes you should be aware of when exporting report results to XML:

- The existing schema file that you specify must also be generated from the web report or report tab. If you use other schema files, a NullPointerException exception will be thrown out.
- If the existing schema file that you specify contains only data, and you have cleared the Only Data checkbox in the Export to XML dialog, the exported XML file will then only contain database column information. The content of the exported XML file will be consistent with the schema file it is based on.
- If you delete an element node or an attribute node from the existing schema file, the exported XML file which is based on it, will not contain the corresponding column. In a not only data schema file, only the removal of an element used in a formula control property will affect an exported XML file which is based on it.

  Example: Deleting an element node from a schema file affects the exported XML file which is based on it

  Let's take EmployeeInformation.cls as an example.

  1. Click **File** > **Open** on the menu bar.
  2. In the Open Report dialog, click the **Browse** button to open the catalog file **SampleReports.cat** in `<install_root>\Demo\Reports\SampleReports`, then open the sample report **EmployeeInformation.cls**.
  3. Export EmployeeInformation.cls to an XML file. You will then get two files: EmployeeInformation_report1.xml and EmployeeInformation_report1.xsd.
  4. Open the **EmployeeInformation_report1.xsd** file, and delete EmoloyeeInformation_report1_ReportBody_PagePanel_PageHeaderPanel_ label3 from EmoloyeeInformation_report1_ReportBody_PagePanel_PageHeaderPanel.
  5. Export EmployeeInformation.cls to an XML file again. This time, specify the modified EmployeeInformation_report1.xsd as the file to base it on.

     The newly exported EmployeeInformation_report1.xml file will not contain information about EmoloyeeInformation_report1_ReportBody_PagePanel_PageHeaderPanel_ label3.

- If you do not want to see all the property values in a generated XML file, you can comment out any unwanted properties in this way:

  1. Open the file **JRXMLTag.properties** located in resource_en_US.jar in `<install_root>\lib`.
  2. Add a question mark in front of any unwanted properties to comment them out. See below:

     Change: Suppressed=Suppressed

     To: ?Suppressed=Suppressed
  3. When you export the web report or report tab to XML, the exported XML file will not contain the property Suppressed for each report object.
  4. Save the property file and refresh the one in resource_en_US.jar.

     In addition, when you uncheck Only Data, and add an @ sign in front of the value in JRXMLTag.properties, the property listed in the Element node will then move to the Attribute node.

     For example, in JRXMLTag.properties,

     Change: StartYPos=StartYPos

     To: StartYPos=@StartYPos

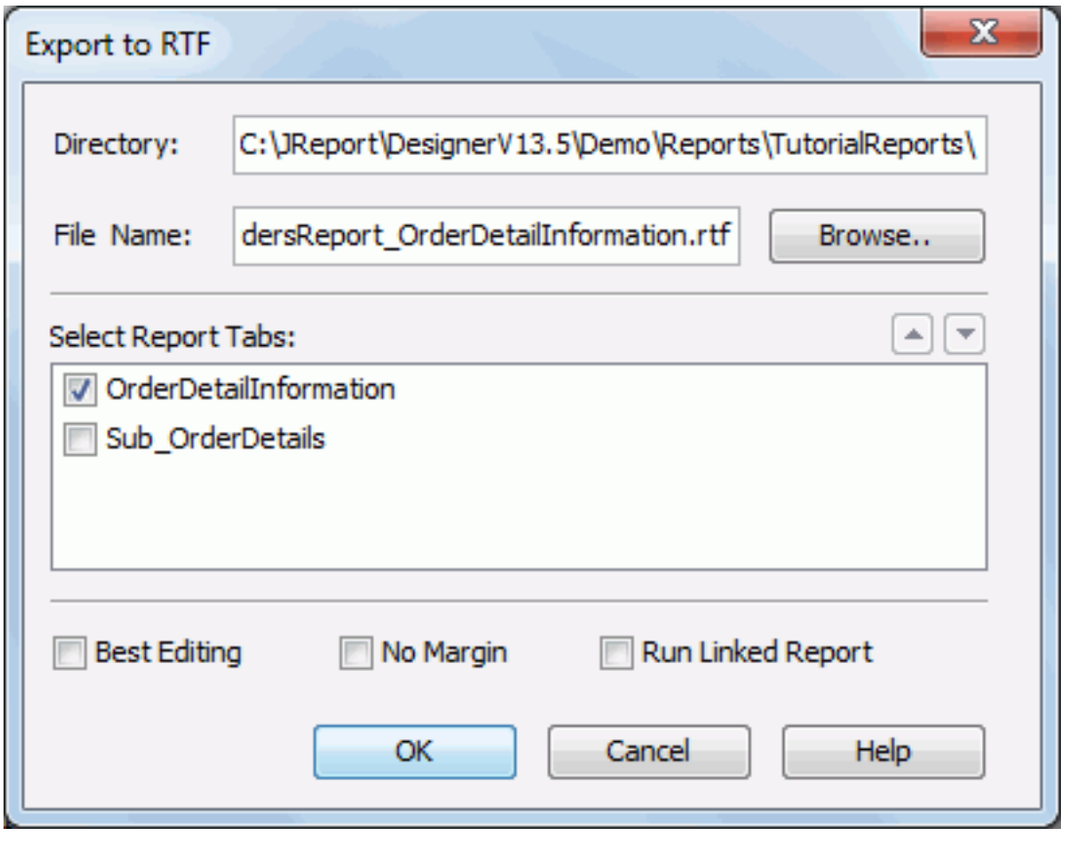
     You will now find that the StartYPos property is now listed in the Attribute node.

     Before adding the @ sign in front of the value, the StartYPos property will have been listed in the Element node.

To RTF

To export the results of a report to a RTF format file, follow the steps below:

1. In JReport Designer, open the report you want to export.
2. Click **File** > **Export To** > **RTF**. The Export to RTF dialog appears.



3. Specify the directory and file name respectively in the Directory and File name text boxes. If you don't specify the name, JReport will apply the report name as the default RTF file name.
4. If you are exporting a page report, in the Select Report Tabs box, select the report tabs in the page report you want to export. The selected report tabs will be exported in the list order. You can click ▲ or ▼ to change the order or the report tabs.
5. Check the **No Margin** checkbox to remove the margins that are originally set while you design the report.
6. Check **Best Editing** to apply the flow layout when exporting.
7. Check **Run Linked Report** if you want to include the linked reports in the exported RTF result.
8. Click **OK**, and a file with the extension .rtf will be generated.

Notes:

- If Best Editing is checked, the results can be edited easily, but the layout may be different from that you designed, because some flow layout properties will not be supported by JReport RTF exporter, such as clear, relative and so on.
- In the Best Editing mode, one report just has one default tab which is in the last paragraph, so others will be ignored.
- In order to get a better effect, Microsoft Office Word 2003 is recommended.

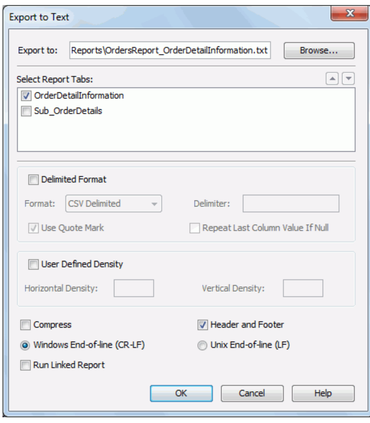
To Text

JReport supports exporting to both a regular Text file and a CSV file.

Exporting the report results to a Text file

To export the results of a report to a Text file, follow the steps below:

1. In JReport Designer, open the report that you want to export.
2. Click **File** > **Export To** > **Text**. The Export to Text dialog appears.



3. Specify the destination of the exported file.
4. If you are exporting a page report, in the Select Report Tabs box, select the report tabs in the page report you want to export. The selected report tabs will be exported in the list order. You can click ▲ or ▼ to change the order or the report tabs.
5. Make sure that the Delimited Format checkbox is unchecked.
6. Specify other settings according to your requirements. Check **Run Linked Report** if you want to include the linked reports in the exported Text result.
7. Click **OK** to perform the export operation.

Note: When exporting a report to a Text file, pay attention to the following:

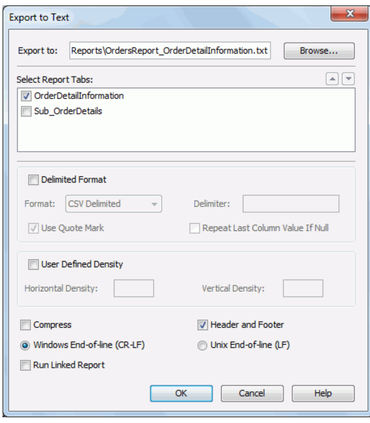
- The property Columned on the page report tabs or the web report will be ignored.
- Each object in the exported result file will be located according to its dimension values (x, y, width and height) you set in the Report Inspector.
- The list object could not be exported to a Text file.
- If the report contains drop-down list object, only the selected value in the list can be exported.
- The objects, which seem in the same line in other formats (such as PDF), may be in the different lines, or some unwanted blank lines will be generated. The exception is caused by the differences between the dimension values - y and height.

Exporting the report results to a CSV file

When exporting the results of a report to a CSV file, you can set the property Columned to decide whether to export the results to a general CSV file or a columned CSV file. If you set the property to false, the report will be exported to a general CSV file. If you want to export the report to a columned CSV file, set the property to true.

The following is an example about exporting a page report which contains only one report tab to a general CSV file:

1. In JReport Designer, open the page report.
2. In the Report Inspector, select the root node that represents the report tab and set the property Columned to false.
3. Click **File** > **Export To** > **Text**. The Export to Text dialog appears.



4. Specify the destination of the exported file as required.
5. Check **Delimited Format** checkbox, and then select **CSV Delimited** from the Format drop-down list.
6. To enable records to be aligned well in the exported file, you are recommended to check **Use Quote Mark**. Otherwise, the commas in data values such as $78,321.00 will conflict with the delimiter (comma), and may potentially cause the data to be separated incorrectly when the exported file is further processed by other applications. The following shows an example:

   Scarsdale,The Java King,(914) 657-9823,(914) 657-8745,$78,321.00

   As shown above, it is difficult to determine whether the value is $78321.00 or $78 and 321.00.
7. Click **OK** to start exporting.

Notes:

- No matter a report is exported to a general CSV file or a columned CSV file,
  - The data of each field should be in one column. If there is no data in a column, empty strings (" ") will be placed there automatically. In which case, the empty strings cannot be compressed. Only if there is no data in the entire line, can a row be compressed when you check the Compress option.
  - In the Export to Text dialog, if you select Custom Delimited, you should export the report to a .txt file instead of a .csv file. Then, open the exported file with Excel, and the Text Import wizard will be displayed. In this wizard, check the **Delimited** radio button, click **Next**, check **Other** on the Delimiters panel and key in the delimiter you specify in the Export to Text dialog, and then click **Finish** to view the results.
- When you specify to export a report to a general CSV file,
  - Objects in the exported file will be located according to its dimension values (X, Y, Width, Height) you set in the Report Inspector.
  - In the exported file, you may find that some column names are not aligned with their DBFields, so they may be displayed in different columns. To resolve this problem, make both the column names and its DBFields to be left aligned and adjust their location using the same Geometry X before exporting it to a general CSV file.
- When you specify to export a report to a columned CSV file,
  - Objects in the exported file will be located according to their Row Index and Column Index property values. When setting values for these two properties, pay attention to the following:
    - Make sure that the Row Index and Column Index property values for all objects in the report do not conflict with each other. Otherwise, some objects may be replaced.
    - If either the Row Index value or Column Index value of an object is less than 1, JReport will ignore this object and all its children.
  - In the exported file, you may find that some column names are not aligned with their DBFields. They may be displayed in different columns. To resolve this problem, set the Column Index value of the column name and its DBField to the same; if they are different, adjust it manually before exporting it to CSV format.
  - If the report contains drawing objects, you need to manually adjust these objects' geometry properties, which are Top Attach Column, Top Attach Row, Bottom Attach Column and Bottom Attach Row.
  - You are not recommended to export a mailing label report to a columned CSV file.

Improving the performance of running a report to Delimited Format on JReport Server

When exporting a page report to Text, if you want to improve the performance of running the page report to Delimited Format (such as the CSV format) on JReport Server, you are first required to set certain settings in JReport Designer. After which you can then publish the report to JReport Server for running to Delimited Format. See the example below:

1. Start JReport Designer and open the page report tab you want to run to Delimited Format on JReport Server.
2. In the Report Inspector, browse to the node of the report tab, find the two properties Fast Pass and Columned, and then set them to true.
3. Choose the object you want to display in the result file, and then set the location for where you want it to be shown in the result file.

   To do this, select the object, and set its Column Index and Row Index properties in the Report Inspector as required. ColumnIndex refers to the column where the object will be located, and RowIndex refers to the row where the object will be located in the result file. The numbers of the column and row start with 1, and are counted within its parent. For example, if you want to place the object in the first column and the second row in the detail section, set ColumnIndex = 1 and RowIndex = 2.
4. Go to the server.properties file in <server_install_root>\bin, and change the property engine.single_thread = false. This enables the property Fast Pass to take effect when running the report tab to Delimited Format.
5. Publish the page report containing the report tab to JReport Server. When you run or schedule the report, after you choose the report tab and then choose To Text format, select the **Delimited Format** box, and then click **Submit** to generate a Delimited Format file.

Notes:

- When the Fast Pass property is set to true, in the report header panel, the values of the objects that use global parameters to make calculations will be incorrect. Page level calculation, such as Page Number and Total Page Number, will also be incorrect. In order to avoid such errors, you should move these objects from the report header panel to the report footer panel.
- Remember to set the positions for objects with the Column Index and Row Index properties. Otherwise, all the objects will be located in the first column and the first row of the result file.

Improving the performance of exporting to Excel

Normally, when a report is being exported to an Excel file, the coordinates of the report objects are calculated by JReport Engine, and can require a great deal of processing time. However, you can arrange the coordinates of the report objects manually at design time, so that the extra position calculating time can be saved. To do this:

1. Select the node that represents the report in the Report Inspector.
2. In the Properties panel, set the Columned property for the report to true.
3. Set the values of Column Index and Row Index for each object to define the position of the object in the exported Excel file.
4. Click **File** > **Export To** > **Excel**.
5. In the Export to Excel dialog, select **Column Format**.
6. Set the other options as required.
7. Click **OK** to start exporting.

Controlling banded object panels in the exported Excel file

You can control panels of a banded object in the exported Excel file by settings the following three properties:

- Merge to Next Panel: Merges current panel to the next panel. It applied to BandedHeader, BandedPageHeader, GroupHeader, and GroupFooter panels.
- Repeat in Detail Panel: Repeats certain panel with the Detail panel. It applied to GroupHeader and BandedPageHeader panels.
- Remove Blank Row: Removes blank rows in the panel. It applied to BandedPageHeader, GroupHeader, and Detail panels.

Notes:

- The three properties take effect only when the Columned property of a report tab is set to true.
- There is an operation order when those three properties work together: first, Merge to Next Panel; second, Repeat in Detail Panel; third, Romove Blank Row.

Merging panels in the exported Excel file

You can specify to merge the BandedHeader, BandedPageHeader, GroupHeader, or GroupFooter panel in a banded object into the panel next to it in the exported Excel file. To do this:

1. Make sure the Columned property of the report that contains the banded object is set to true.
2. In the Report Inspector, select the node that represents the panel you want to merge.
3. Set the value of the Merge to Next Panel property to true.
4. Click **File** > **Export To** > **Excel**.
5. In the Export to Excel dialog, select **Column Format**.
6. Set the other settings according to your requirements and click **OK** to start exporting.

In the exported Excel file, you can find that the specified panel has been merged to the panel next to it.

Notes:

- When two or more objects in the to be merged panels have the same Row Index and Column Index property values, after merging the panels, in the exported Excel file, the objects will be merged into one cell, and they will be located in the cell based on this rule: the object whose X/Y property value is smaller than others will be put to front, and Y has higher priority than X. However, you can improve the appearance of the report by relocating the objects.
- Do not set the property Merge to Next Panel to true for the last GroupFooter panel. That will make the exported Excel file incorrect.
- Do not set the property Merge to Next Panel to true for the panel that contains crosstab or subreport.

Example

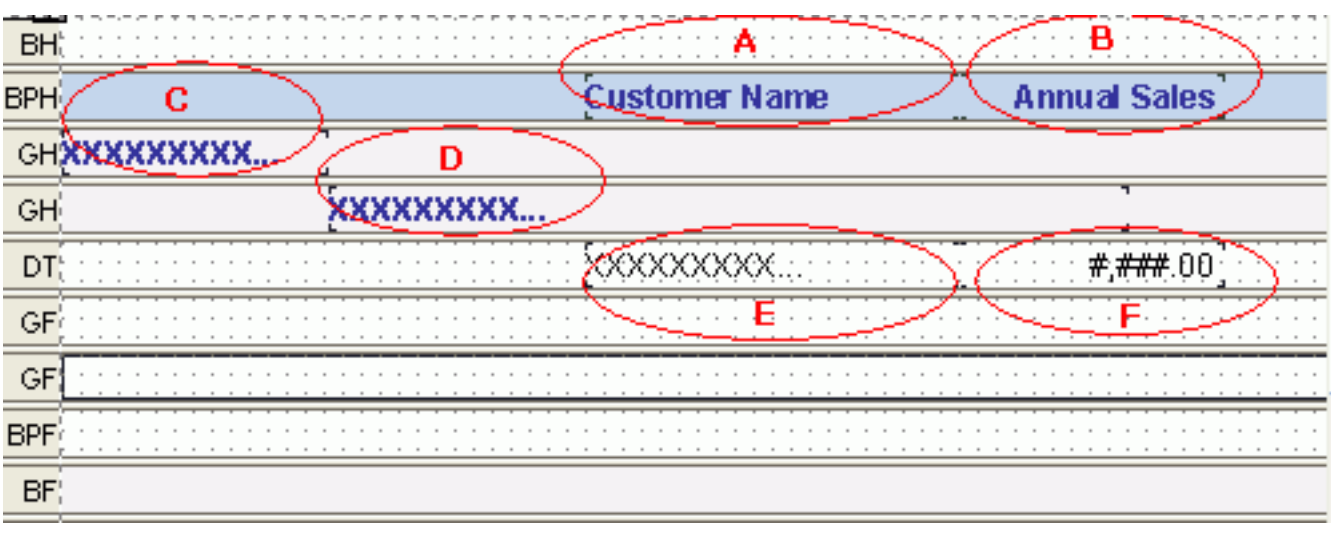
Suppose there is a standard banded report which displays the fields Customer Name, Annual Sales, and is grouped by Customer_Country and Customer_Region.

	Customer Name	Annual Sales
Australia		
	Asia-Pacific (APAC)	
	Cafe Brazil - Sydney	84,124.00
Belgium		
	Europe, Middle East, Africa (EMEA)	
	Coffee of Belgium	98,760.00
Brazil		
	Latin America (LATAM)	
	Casa do Roaster de café	88,757.00
Canada		
	North America (NA)	
	Mocha Coffee House	50,232.00
	Kona Premium	142,896.00
	Bob's Coffee	45,897.00

Then we set the Columned property of the report to true and Merge to Next Panel of the first GroupHeader panel to true as well, save the report and export it to Excel. We can see that the two group by panels are merged into one and fields in the two panels are put in one cell in the exported Excel file.

A	B
Customer Name	Annual Sales
Australia Asia-Pacific (APAC)	
Cafe Brazil - Sydney	84,124.00
Belgium Europe, Middle East, Africa (EMEA)	
Coffee of Belgium	98,760.00
Brazil Latin America (LATAM)	
Casa do Roaster de café	88,757.00
Canada North America (NA)	
Mocha Coffee House	50,232.00
Kona Premium	142,896.00
Bob's Coffee	45,897.00

In order to improve the layout of the Excel result, now we specify the Column Index and Row Index properties of the objects as follows:



Object	Column Index	Row Index
A	3	1
B	4	1
C	1	1
D	2	1
E	3	1
F	4	1

Save the report and export it to Excel again. We can see that fields in the Excel file are shown more clearly this time.

A	B	C	D
		Customer Name	Annual Sales
Australia	Asia-Pacific (APAC)		
		Cafe Brazil - Sydney	84,124.00
Belgium	Europe, Middle East, Africa (EMEA)		
		Coffee of Belgium	98,760.00
Brazil	Latin America (LATAM)		
		Casa do Roaster de café	88,757.00
Canada	North America (NA)		
		Mocha Coffee House	50,232.00
		Kona Premium	142,896.00
		Bob's Coffee	45,897.00

Repeating objects in the GroupHeader or BandedPageHeader panel in the Detail panel

If you want to repeat objects in the GroupHeader panel or BandedPageHeader panel in the Detail panel of a banded object in the exported Excel file, follow the steps below:

1. Make sure the Columned property of the report that contains the banded object is set to true.
2. In the Report Inspector, select the node that represents the panel you want to repeat.
3. Set the value of the Repeat in Detail Panel property to true.
4. Click **File** > **Export To** > **Excel**.
5. In the Export to Excel dialog, select **Column Format**.
6. Set the other settings according to your requirements and click **OK** to start exporting.

In the exported Excel file, you can find that objects in the specified panel are repeated in the Detail panel.

Notes:

- When the objects in the GroupHeader/BandedPageHeader panel have the same Row Index and Column Index property values with those in the Detail panel, and you specify to repeat the GroupHeader/BandedPageHeader panel in the Detail panel, in the exported Excel file, the repeated objects will be put in the same cells with the ones in the Detail panel based on this rule: the object whose X/Y property value is smaller than others will be put to front, and Y has higher priority than X. However, you can improve the appearance of the report by relocating the objects.
- When you specify to repeat a GroupHeader panel, if there is a chart in the panel, the chart cannot be repeated.
- If a banded object contains more than one group by fields, that is to say, it has several GroupHeader panels, when you specify to repeat the first GroupHeader panel in the Detail panel, it will be repeated in the second GroupHeader panel as well.

Removing blank row in a panel

The property Remove Blank Row is provided to make the report look compact in the exported Excel file. It applied to BandedPageHeader, GroupHeader, and Detail panels of a banded object.

To remove blank rows in the exported Excel file:

1. Make sure the Columned property of the report that contains the banded object is set to true.
2. In the Report Inspector, select the node that represents the panel where blank rows may appear in the Report Inspector.
3. Set the value of the Remove Blank Row property to true.
4. Click **File** > **Export To** > **Excel**.
5. In the Export to Excel dialog, select **Column Format**.
6. Set the other settings according to your requirements and click **OK** to start exporting.

In the exported Excel file, you can find that there will be no blank rows in the specified panel.

Specifying subreport location in the exported Excel file

If a report contains a subreport, you can specify where the subreport where be put in the exported Excel file in the following way:

1. Open the report you want to export, which contains a subreport.
2. Select the node that represents the subreport in the Report Inspector.
3. In the Properties panel, set the On New Sheet property as required.
   - If you want to put the subreport in the same worksheet together with the primary report, set the value to false.
   - If you want to put the subreport in a new worksheet in the exported Excel file, set the property value to true, then enter a name for the new worksheet in the Sheet Name property's value cell. Make sure the name you specify here isn't used by any other worksheets.
4. Click **File** > **Export To** > **Excel**.
5. In the Export to Excel dialog, make sure Report Format is selected.
6. Set the other options as required.
7. Click **OK** to start exporting.

   If you set On New Sheet to true, after exporting the report successfully, you will find that the primary report is exported into the first worksheet, and there are some links on it. Click the links to go to the subreport in the second worksheet of the Excel file.

Note: If the subreport are cut into several pages, that is to say you have defined some page settings on the subreport, the links in the primary report will be subreport1_0, subreport1_1, subreport1_2 and so on. You can click the link to see the corresponding part you want.

Realigning objects for a better appearance in the exported Excel file

Normally, when a report is exported to an Excel file, the position and size of the objects in the original report are automatically rearranged by JReport Designer. However, you can also arrange most objects manually, especially for exporting to Excel, so that they can be well-aligned in the exported Excel file.

Before you can resize or reposition an object, you should set the Columned property of the report to true, then:

Relocating an object for exporting to Excel

1. Select the node that represents the object in the Report Inspector.
2. In the Properties panel, set the Column Index and Row Index properties. The values you specify here determine the object's location in the exported Excel file.

   Note: If the Position property of an object is set to static, its Column Index and Row Index properties will not take affect when the report is exported to Excel result.
3. Click **File** > **Export To** > **Excel**.
4. In the Export to Excel dialog, select **Column Format**.
5. Set the other options as required and click **OK** to start exporting.

The object in the exported Excel file will then be located according to its Column Index and Row Index property values.

Alignment for drawing objects is different from other objects. The following explains the difference in detail:

1. Select the node that represents the drawing object in the Report Inspector.
2. In the Properties panel, set the following properties as required: Top Attach Column, Top Attach Row, Bottom Attach Column, Bottom Attach Row. These four properties together determine the coordinates and size of the drawing object.

   The first pair of the properties (Top Attach Column, Top Attach Row) locates a cell and uses its upper left corner as the upper left corner of the drawing object, while the second pair (Bottom Attach Column, Bottom Attach Row) locates another cell and uses its upper left corner as the lower right corner of the drawing object.

   For example, if the properties for an oval are set as:

   Top Attach Column: 1, Top Attach Row: 2, Bottom Attach Column: 3, Bottom Attach Row: 4

   Then, the upper left cell will be A2 and the lower right cell will be C4.

   Notes:

   - The value you set for the Top Attach Column (Top Attach Row) property should be always larger than that for the Bottom Attach Column (Bottom Attach Row) property.
   - The arc object cannot be shown in Excel and therefore these four properties for arc does not work.
3. Click **File** > **Export To** > **Excel**.
4. In the Export to Excel dialog, select **Column Format**.
5. Set the other options as required and click **OK** to start exporting.

The drawing object in the exported Excel file will then be aligned according to the four property values.

Resizing an object for exporting to Excel

1. Select the node that represents the object in the Report Inspector.
2. In the Properties panel, set the Column Number and Row Number properties as required. The values you specify here determine the object's size in the exported Excel file, and only a few objects, such as images, charts, and user-defined objects, have these two properties.
3. Click **File** > **Export To** > **Excel**.
4. In the Export to Excel dialog, select **Column Format**.
5. Set the other options as required and click **OK** to start exporting.

The object in the exported Excel file will then be resized according to its Column Number and Row Number property values.

Setting the Excel column width

You can set the width of each column for the exported Excel file in JReport Designer. To do this:

1. Select the node that represents the report in the Report Inspector.
2. In the Properties panel, set the Column Width List property as required.

   Type the width for the columns in turn, using a semicolon (;) to separate each value. If you don't want to specify the width for a certain column, omit the value. JReport will then use the default value (8) for the column.

   For example, the value list 12;15;;20 makes the first column width 12, the second 15, the third 8, which is the default value, and the fourth 20.
3. Click **File** > **Export To** > **Excel**.
4. In the Export to Excel dialog, select **Column Format**.
5. Set the other options as required.
6. Click **OK** to start exporting.

Notes:

- The Column Width List property takes effect only when the report's Columned property is set to true and the Column Format option in the Export to Excel dialog is selected.
- If a report contains a subreport, the Column Width List property for the subreport is disabled when exporting the report to Excel.

Setting the maximum number of rows for every worksheet

You can control the maximum number of rows in every worksheet in the exported Excel file. To do this:

1. Select the node that represents the report in the Report Inspector.
2. In the Properties panel, set the Rows per Sheet property as required.
3. Click **File** > **Export To** > **Excel**.
4. In the Export to Excel dialog, set the options as required.
5. Click **OK** to start exporting.

Notes:

- For the Excel version Excel 97-2003 Workbook (*.xls), the value ranges from 0 to 65000. For the Excel version Excel Workbook (*.xlsx), the value ranges from 0 to 1048000. If you set the value of the Rows per Sheet property beyond 65000 and choose the Excel version Excel 97-2003 Workbook (*.xls) in the Export to Excel dialog, 65000 will be used as the maximum number of rows in every worksheet.
- When exporting a report to Excel, rows in the same page of the report cannot be separated and exported to different worksheets, thus the number of rows that is exported to a worksheet may be greater than the maximum number of rows that you set for the property Rows per Sheet. For example, if you set the value of Rows per Sheet to 100, all the contents of the first page will be exported to the first worksheet. After that, the rows of the first worksheet will be tested automatically by JReport Designer. If the number of rows is less than 100, the next page will still be exported to the first worksheet, even if the number of rows in the first page and the next page is greater than 100; If the number of rows in the first page is equal to or more than 100, a new worksheet will be generated and the second page will be exported to the new worksheet.

Setting the Excel buffer size

You can store the report result in Excel buffer sheets when exporting to Excel format. To do this:

1. Select the node that represents the report in the Report Inspector.
2. In the Properties panel, set the Excel Buffer Size property as required.

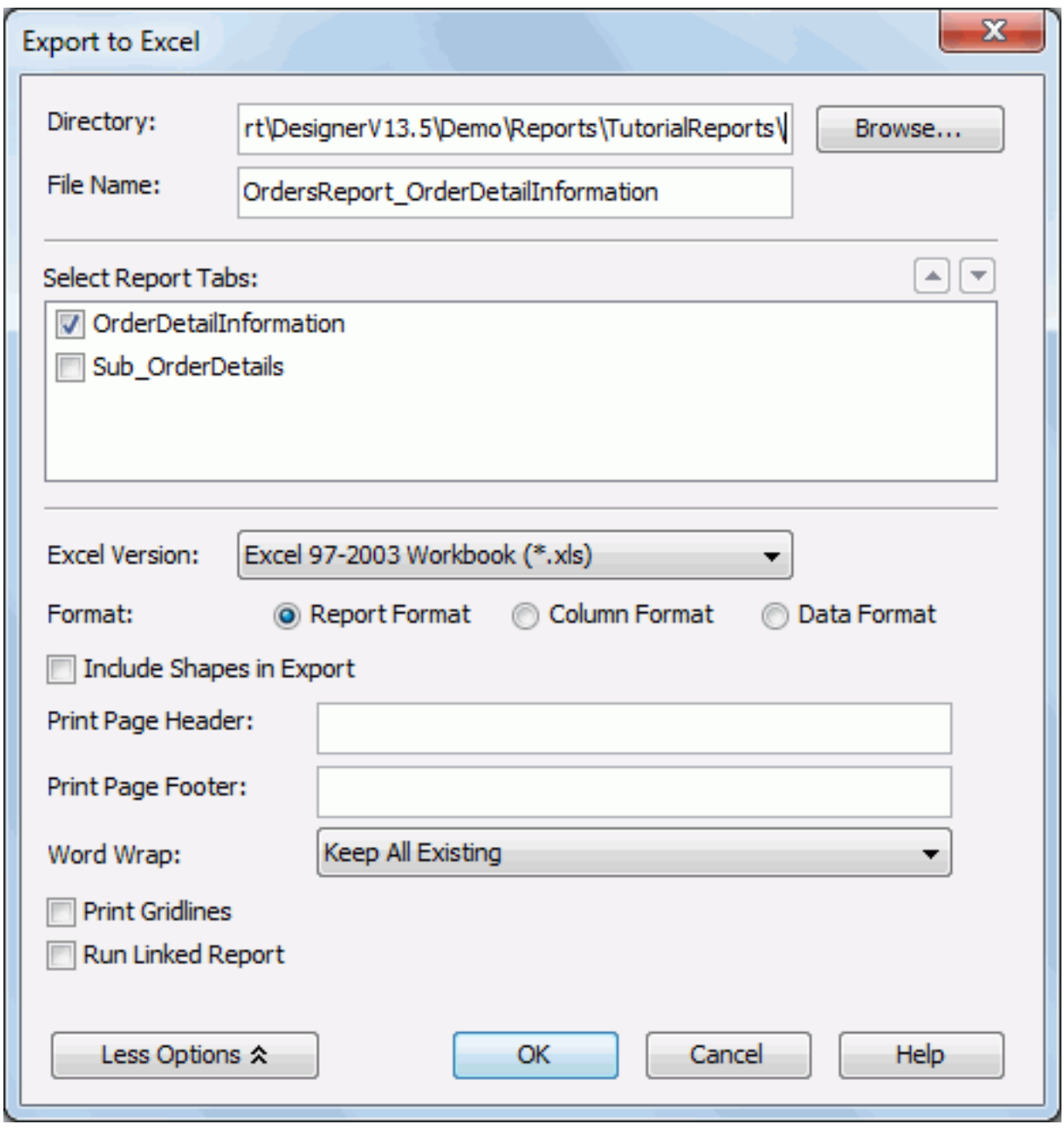
   The default size is 1, which indicates that 1 sheet of the report result will be allocated to the result buffer, while the other sheets will be stored onto disk. If you have enough memory, to improve performance, you can increase the Excel buffer size to store more sheets of the excel format report result.
3. Click **File** > **Export To** > **Excel**.
4. In the Export to Excel dialog, make sure Data Format is not checked and set the other options as required.
5. Click **OK** to start exporting.

Note: The Excel Buffer Size property takes effect only when the Data Format option in the Export to Excel dialog is not selected.

To Excel

To export the results of a report to an Excel file, follow the steps below:

1. In JReport Designer, open the report that you want to export.
2. Click **File** > **Export To** > **Excel**. The Export to Excel dialog appears.



3. Click the **Browse** button to specify the destination directory where the exported Excel file will be placed, or type the destination directory into the Directory text field directly (make sure that the folder you specify here does exist).
4. Specify the name of the exported Excel file in the File Name text field. If you do not type the name, JReport will use the report name as the Excel file name by default.
5. If you are exporting a page report, in the Select Report Tabs box, select the report tabs in the page report you want to export. The selected report tabs will be exported into multiple sheets of the Excel file in the list order. You can click ▲ or ▼ to change the order or the report tabs.
6. From the Excel Version drop-down list, select the version for the exported Excel file.
7. Specify the format of the exported Excel file.
   - To make the format of the result file match the format as designed in the template, check **Report Format**, and then specify the available settings according to your requirements.
   - To export the report results to a general Excel file, check **Column Format**, in which case, you can set the Columned property value for the report tab in the Report Inspector to decide the calculation method used for all components' row/column values in the report when exporting (due to the particularity of mailing label report, do not set the Columned property to true when you want to export it in this format).
   - To export only the report data without format, check **Data Format**. This option is only available for the Excel version Excel 97-2003 Workbook (*.xls).
8. Check **Run Linked Report** if you want to include the linked reports in the exported Excel result.
9. When done, click **OK** to start exporting.

Notes:

- Multiple colors and languages in a report can be exported to an Excel file, but multiple pie charts and the arc drawing object cannot be. For example, Excel only supports 56 kinds of colors, if there are more than 56 kinds of colors in a report, when you export the report to Excel, some colors will be merged.
- When exporting a report to Excel,
  - Paragraphs and texts in the report will be exported as objects.
  - If the data type of the field on the category axis of a chart is Date, DateTime or Time whose format confirms to the date format of the Excel, the axis will be exported as a date axis, otherwise, it will not be exported as a date axis.
  - Gauge and surface chart in the report will be exported as bar chart while stock chart will be exported as image.
  - The minor tick marks on a chart will not be exported.
  - If the field on the category axis of a chart is of Number type and you use a constant interval to customize the tick marks on the axis, the settings will not take effect.
  - If the report contains a drop-down list, only the value selected in the list will be exported.
  - If you set an object's Export to Excel property to false, the object and other objects contained in it will not be exported.
- When Data Format is selected in the Export to Excel dialog, formulas, barcodes, images, HyperLink property, charts and all drawing objects will not be supported, therefore, we recommend not using this format when exporting report results to Excel.

The following are some specific topics related with exporting the report results to Excel:

- Setting the Excel buffer size
- Setting the maximum number of rows for every worksheet
- Setting the Excel column width
- Realigning objects for a better appearance in the exported Excel file
- Specifying subreport location in the exported Excel file
- Controlling banded object panels in the exported Excel file
- Improving the performance of exporting to Excel

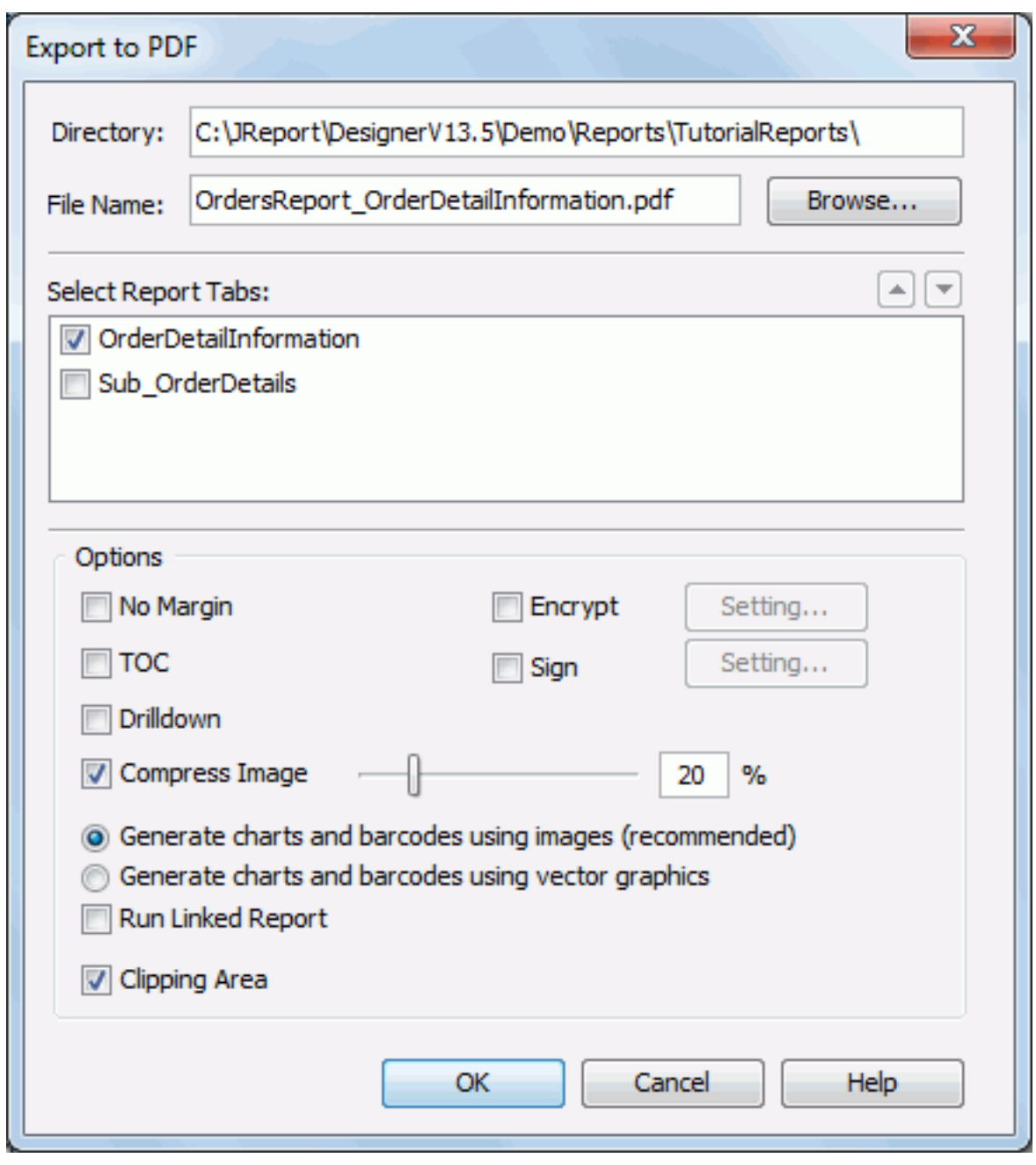
Related topics:

- Creating dynamic charts in Excel

To PDF

To export the results of a report to PDF file, follow the steps below:

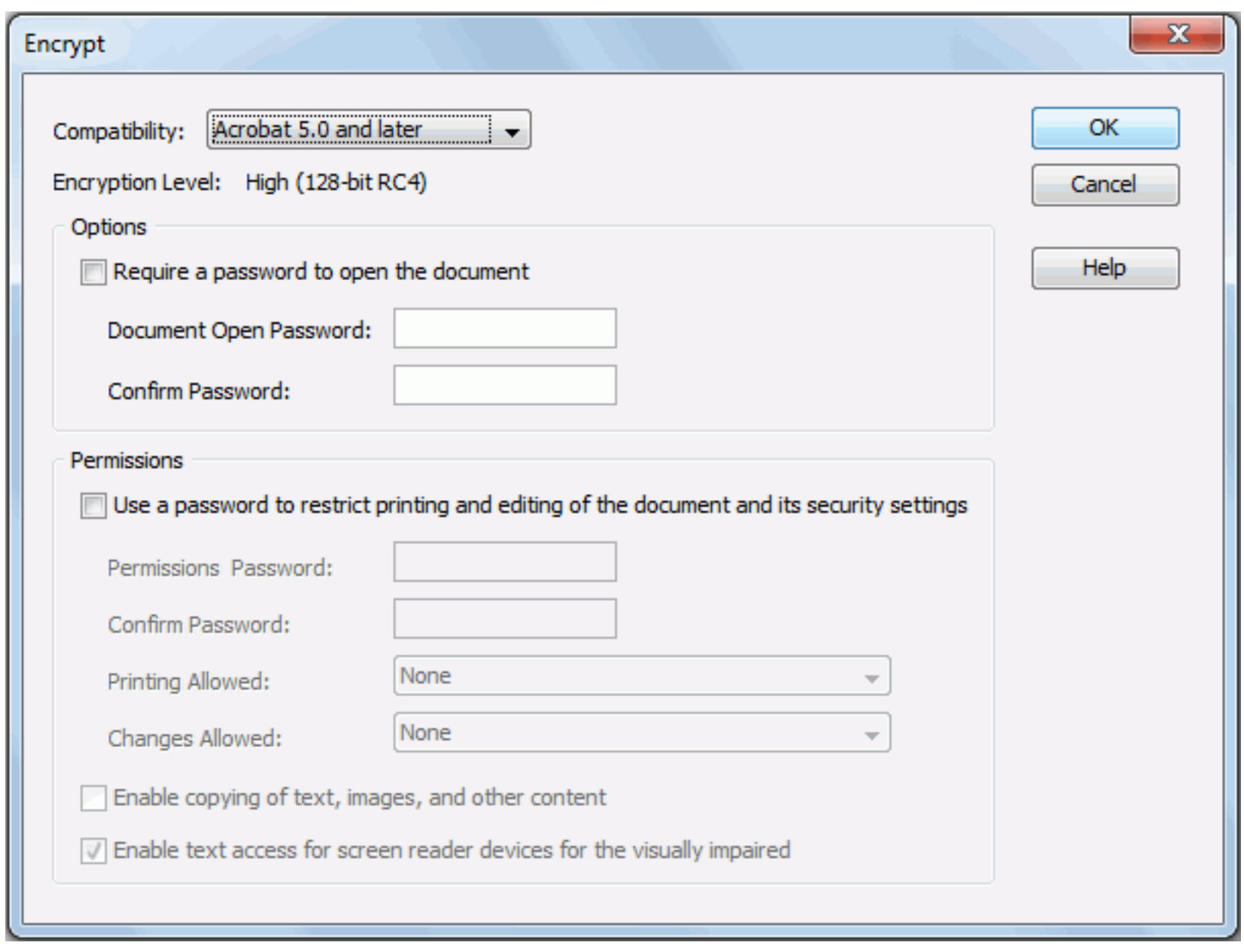
1. In JReport Designer, open the report that you want to export.
2. Click **File** > **Export To** > **PDF**.
3. In the Export to PDF dialog, specify the settings as required.



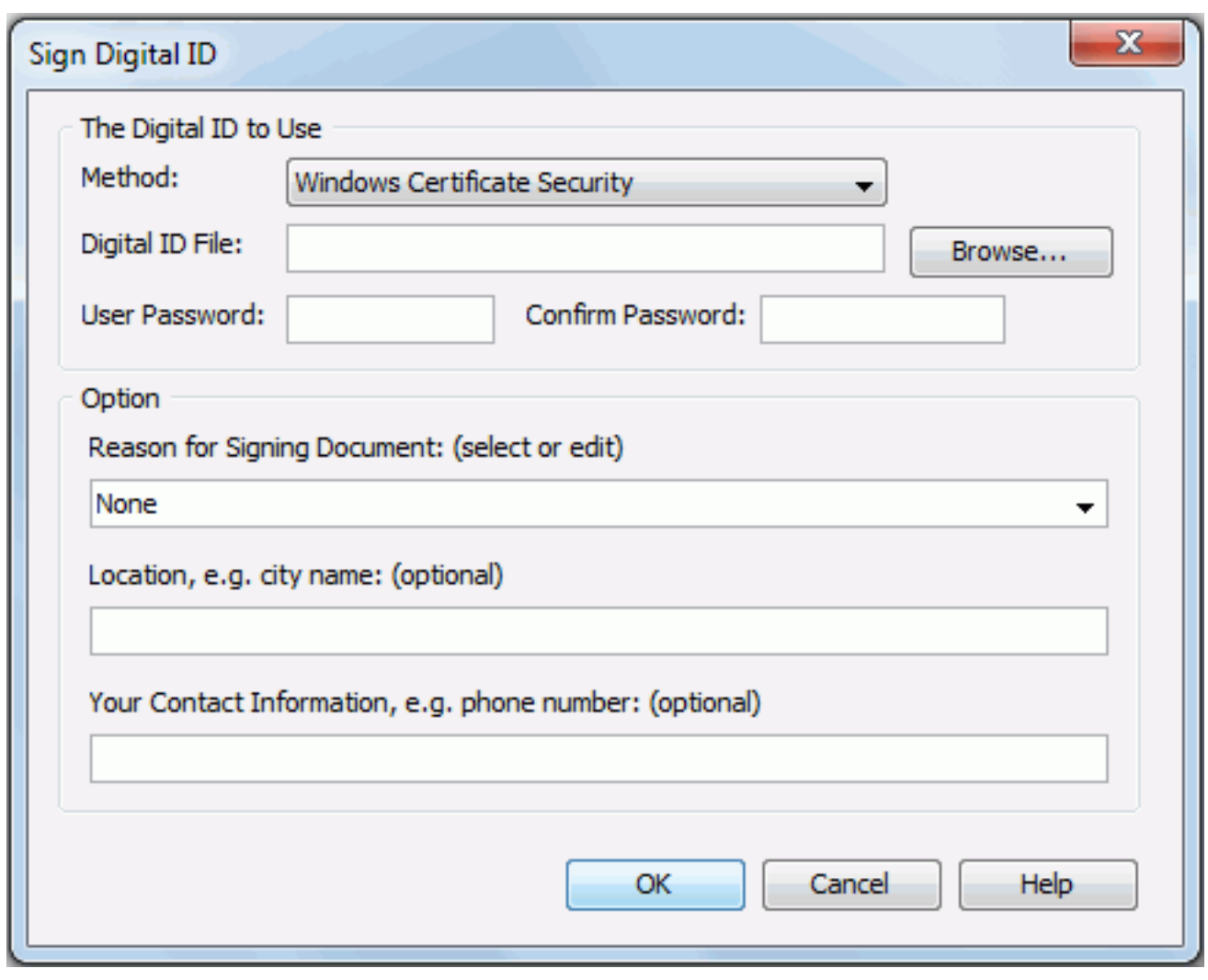
4. When done, click **OK** to generate the file.

The following example shows how to export the results of a report to PDF format and view the exported result file, taking the sample report CustomerAnalysis.cls in SampleReports.cat as an example:

1. Click **File** > **Open** on the menu bar.
2. In the Open Report dialog, click the **Browse** button to open the catalog file **SampleReports.cat** in `<install_root>\Demo\Reports\SampleReports`, then open **CustomerAnalysis.cls** in the catalog.
3. Click **File** > **Export To** > **PDF** to open the Export to PDF dialog.
4. Specify the path where the files will be exported and the file name as required. Here we use the default path `C:\JReport\Designer\Demo\Reports\SampleReports\` and default name CustomerAnalysis_CustomerAnalysis.pdf.
5. Make sure that **TOC**, **Drilldown** and **Generate charts and barcodes using vector graphics** are checked.
6. Check the **Encrypt** option to set the document open password and permissions password in the Encrypt dialog.



7. Check the **Sign** option to set a digital ID for the PDF file in the Sign Digital ID dialog.



8. Check **Run Linked Report** if you want to include the linked reports in the exported PDF result.
9. When done, click **OK** to generate the files.
10. Open the directory `C:\JReport\Designer\Demo\Reports\SampleReports\` and fine the exported files.
11. Open the file **CustomerAnalysis_CustomerAnalysis.pdf** with the inputted document open password or permissions password using the Adobe Acrobat software.
12. The Table of Contents (TOC) is displayed in the left Bookmarks panel, and includes the names of both groups and subgroups. If you click a group, the contents will be shown in the right pane.
13. The signature is displayed in the left Security Settings panel, and all information you set can be found there.
14. Place the mouse pointer over a summary field. When it becomes a hand, click to view the details of the summary.

    Note that summaries should be inserted into the report in advance. They cannot be hidden or suppressed; otherwise you will not be able to get the drill-down files.

    Also, since the option Generate charts and barcodes using vector graphics is checked when exporting the report, when zooming in the chart, you will find that the chart is a vector graphic which never distorts.

Notes:

- If you check Compress Image in the Export to PDF dialog, the transparent image will be compressed, but the transparent attribute will be ignored.
- JReport stores the finished pages in RAM before the pages are finally written to a PDF file. When the result file is too large, you may find that you are low on RAM, leading to an OutOfMemoryError. You can solve this problem by changing the maximum heap size (-Xmx100m). Also, you can use the old method (Simulated Printing).
- If the color mode of your display adapter is less than 256 colors, when you export to PDF file using the standard method, charts cannot be compressed. This is because JReport catches an image with less than 256 colors as a GIF image. A GIF image cannot be compressed.
- When you export a report which has summary fields, if the Drill Down checkbox is checked, every summary field in this report will generate a new file which is linked in the main file.
- In the Sign Digital ID dialog, by default, the information of Digital ID File and Password are checked. If there are any problems, the error messages will prompt you, such as Unable to import the chosen Digital ID file! or Incorrect password! and so on.
- Before exporting a page report tab or a web report using true type fonts, you should select the report tab name or the web report in the Report Inspector, and then set Embedded Fonts equal to all true type fonts that you have used in the report tab or web report. Otherwise the PDF files may not be displayed correctly.
- When you export a web control object, barcode or UDO to PDF, if you select Generate charts and barcodes using vector graphics, the results will be the vector graphic; if you select the Generate charts and barcodes using images, the results will be the bitmap.
- When a report, in which there is an object beyond its container, is exported to PDF, the overstepped part will be ignored in PDF file, but the overstepped part of objects in a paragraph can still be exported to PDF.
- In order to get a better effect, Acrobat Reader 6.0 is recommended.

To HTML

To export the results of a report to an HTML file, follow the steps below:

1. In JReport Designer, open the report that you want to export.
2. Click **File** > **Export To** > **HTML**.
3. In the Web Wizard, specify the settings as required.
4. When done, click **Finish** to generate the file.

The following example shows you how to export the results of a report to HTML format and view the exported file via a web browser, taking the sample report CustomerAnalysis.cls in SampleReports.cat as an example:

1. Click **File** > **Open** on the menu bar.
2. In the Open Report dialog, click the **Browse** button to open the catalog file **SampleReports.cat** in `<install_root>\Demo\Reports\SampleReports`, then open **CustomerAnalysis.cls** in the catalog.
3. Click **File** > **Export To** > **HTML** to open the Web Wizard.
4. In the Report tab, check **Multiple Files** radio button.
5. Specify the web page name as required. Here we use the default one CustomerAnalysis_CustomerAnalysis. Make sure that **Drilldown** and **TOC** are checked.
6. In the Directory tab, specify where the exported result files will be saved. Here we save them in the default directory `C:\JReport\Designer\Demo\Reports\SampleReports\`.
7. In the Chart Applet tab, select **Image Chart** > **GIF**.
8. Click **Finish** to start generating the files.
9. Open the directory `C:\JReport\Designer\Demo\Reports\SampleReports\` and find the exported files.
10. Open the file **CustomerAnalysis_CustomerAnalysis_1.html**.
11. The Table of Contents (TOC) is displayed in the left pane, including the names of groups and subgroups. If you click a group, the relative contents will be shown in the right pane.
12. Since Drilldown is checked when exporting the report, you can drill down on a summary by placing the mouse pointer over the summary and clicking it when the mouse pointer becomes a hand.

Note that summaries should be inserted into your report in advance. They cannot be hidden or suppressed; otherwise you will not be able to get drilldown files.

Notes:

- When exporting a multi-page report to HTML, you can localize the page navigation link names. For detailed information, refer to Localizing the page navigation links in HTML report outputs.
- Multimedia and shapes (except for straight lines and boxes) in reports cannot be exported to HTML.
- For mainly meeting the requirements from voice agents used by people with eyesight disabilities, JReport supports exported tables/crosstabs in the HTML data table format with SCOPE=Row/Column to get a section 508 compliant result. For detailed descriptions, refer to Accessibility.
- After you export a report to an HTML file, you may view it with charts in a web browser. If that, you must ensure that the chart can be displayed correctly, make sure that you have enabled Additional Unsigned Permissions in the web browser. Moreover, if Applet Chart in the Web Wizard is checked and you use Google Chrome, it can be viewed only when NPAPI is enabled for the web browser. To do this, you need to type **chrome://flags/** in the address bar of the web browser, then scroll down and set Enable NPAPI Mac, Windows to **Enable**.

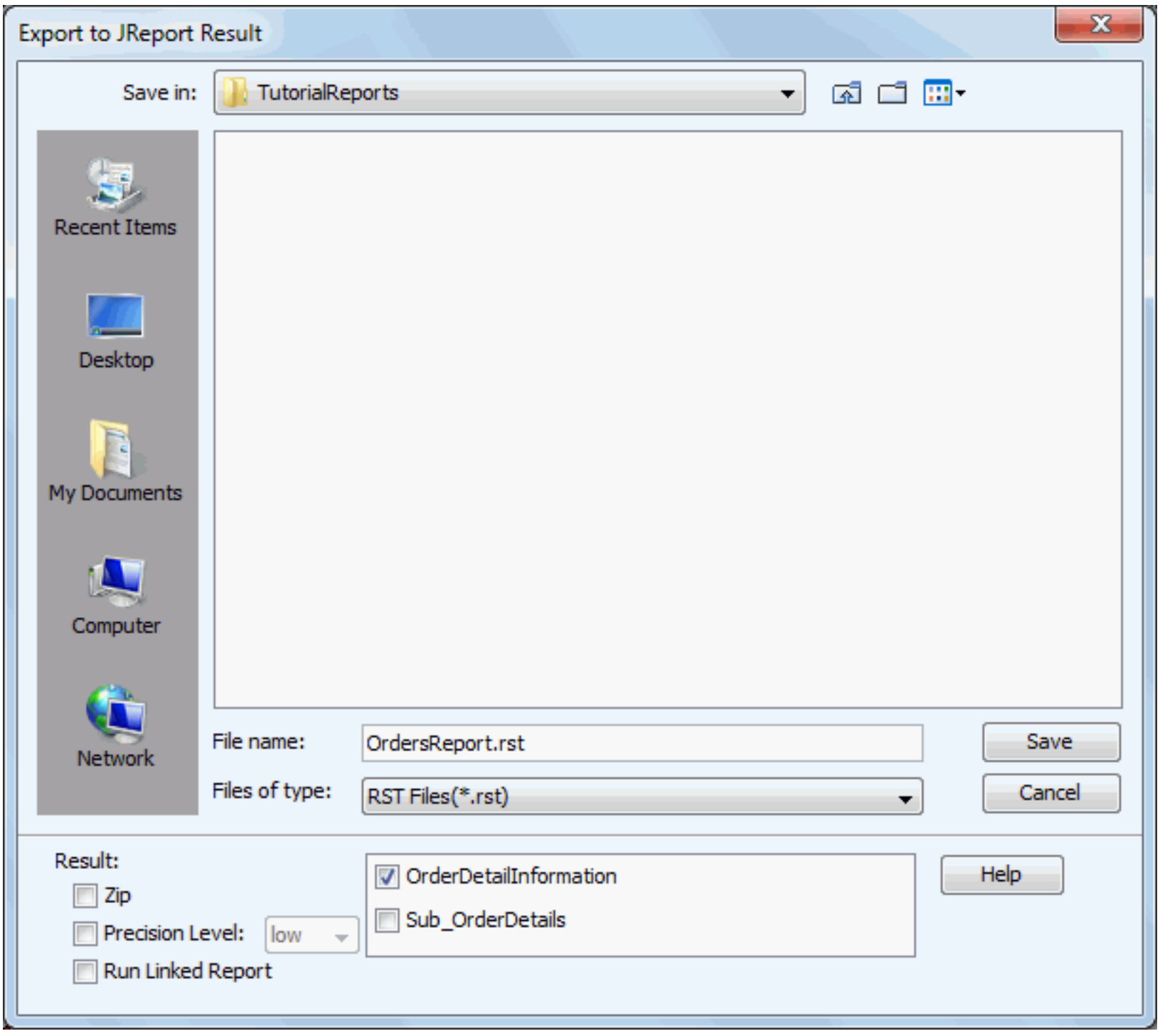
•html>

To JReport result

A JReport result file is JReport's proprietary version of the report results. This file can only be viewed by JReport Server.

To export the results of a report to JReport result:

1. In JReport Designer, open the report that is to be exported.
2. Click **File** > **Export To** > **JReport Result**. The Export to JReport Result dialog appears.



3. Type a name and specify a directory for the result file.
4. If you are exporting a page report, in the Result box, specify the report tabs in the page report you want to export.
5. If you want to export the report to a zip file, check the **Zip** option.
6. Specify the precision level with which you want to export the report from the Precision Level drop-down list.
7. Check **Run Linked Report** if you want to include the linked reports in the exported RST file.
8. Click **OK**, and a .rst file will then be generated in the specified directory.

Notes:

- To make the specified precision take effect, you need to make sure that the Precision Sensitive property of the selected page report tabs is set to true. For details, see Report Property - Precision Sensitive.
- The precision you specify in the Export to JReport Result dialog has higher priority than the one in the Export to category of the Options dialog.

To mail

If you want to send the results of a report to other people, you can attach it to a mail. Be aware that before you can export the report results via e-mail, you must first have your report mail system configured. Otherwise, a warning message will be displayed when you try to export to mail. Here when exporting a page report to mail, only the current specified report tab of the report in JReport Designer will be exported.

Configuring the report mail system

To configure the report mail system, follow the steps below:

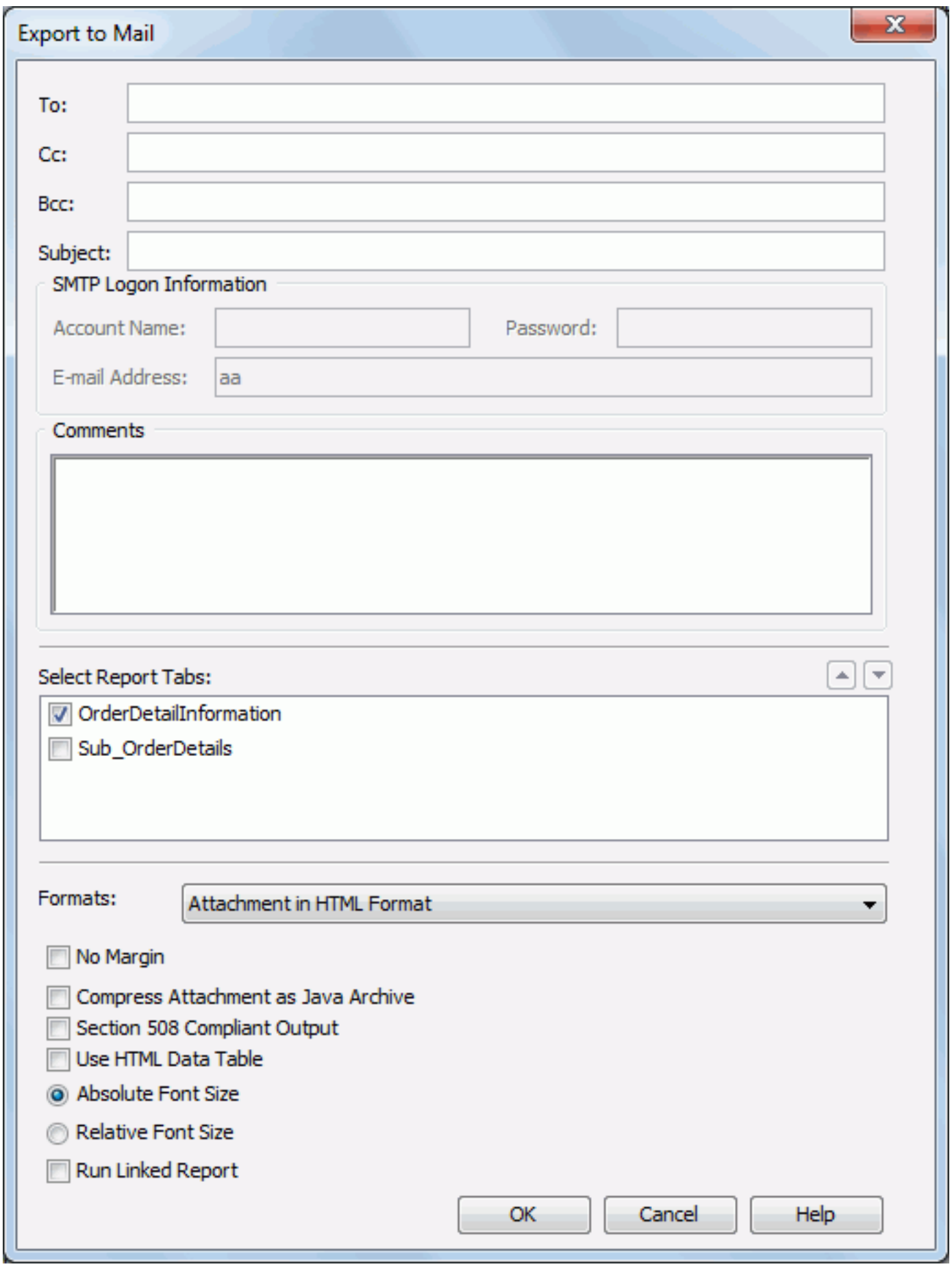
1. In JReport Designer, click **File** > **Options**.
2. In the Options dialog, select the **Export to** category.
3. In the E-mail tab, set up the SMTP Server, Port, Mail Box and Default Mail Format respectively.
4. Check the **Server requires authentication** if the mail server requires authentication.

   Note: If the mail server does not need authentication but you check the Server requires authentication, the mail may not be sent successfully due to the mail server.
5. Check **Compress Attachment as Java Archive** to compress the mail before sending out.
6. Click **OK** to finalize the changes.

Exporting the report results to mail

Now you can export your report results to mail as follows:

1. Open the report that you would like to export.
2. Click **File** > **Export To** > **Mail** to access the Export to Mail dialog.



3. Specify addresses for TO, CC, and BCC respectively.
4. Input the mail subject in the Subject field.
5. If your SMTP server requires authentication, specify the required information in the SMTP Logon Information box.
6. Enter some comments for the mail in the Comments text box, and what you write here will be shown in the text part of your mail.
7. If you are exporting a page report, in the Select Report Tabs box, select the report tabs in the page report you want to export. The selected report tabs will be exported in the list order. You can click or to change the order or the report tabs.
8. Select the required mail format and set the parameters for the format as required (for details about the parameter settings, refer to Export to Mail dialog).
9. Check **Run Linked Report** if you want to include the linked reports in your exported mail.
10. Click **OK**, and the report results will then be sent to the specified mail address. The recipient can open the mail normally, using Outlook Express, or any other mail processing product.

Note: If you select E-mail Result in HTML E-mail Format as the mail format, the comments that you input will be overwritten by the report.

Exporting report results

You can export report results to the following file formats:

- To mail
- To JReport result
- To HTML
- To PDF
- To Excel
- To Text
- To RTF
- To XML
- To PostScript
- To fax
- To Applet

Notes:

- You can export the report data in either design or view mode. However, if you modify the report after opening it, it is recommended to use view mode to export the report data. To do this, click the **View** tab, and then click the **Refresh Data** button on the toolbar. The data will then be re-fetched from the database. After which you can export the report data to the required format.
- When exporting a page report to JReport result, you can specify which report tabs in the current page report you want to export, however, when you export to other formats, only the current open report tab will be exported.

Printing report results

JReport relies on the JDK to create the printable version of report results. By default the JDK 1.4 printing method is used, but you can use other versions if needed.

Specifying printing options

Before you can print reports, you need to first specify the printing options in the Options dialog.

To specify printing options:

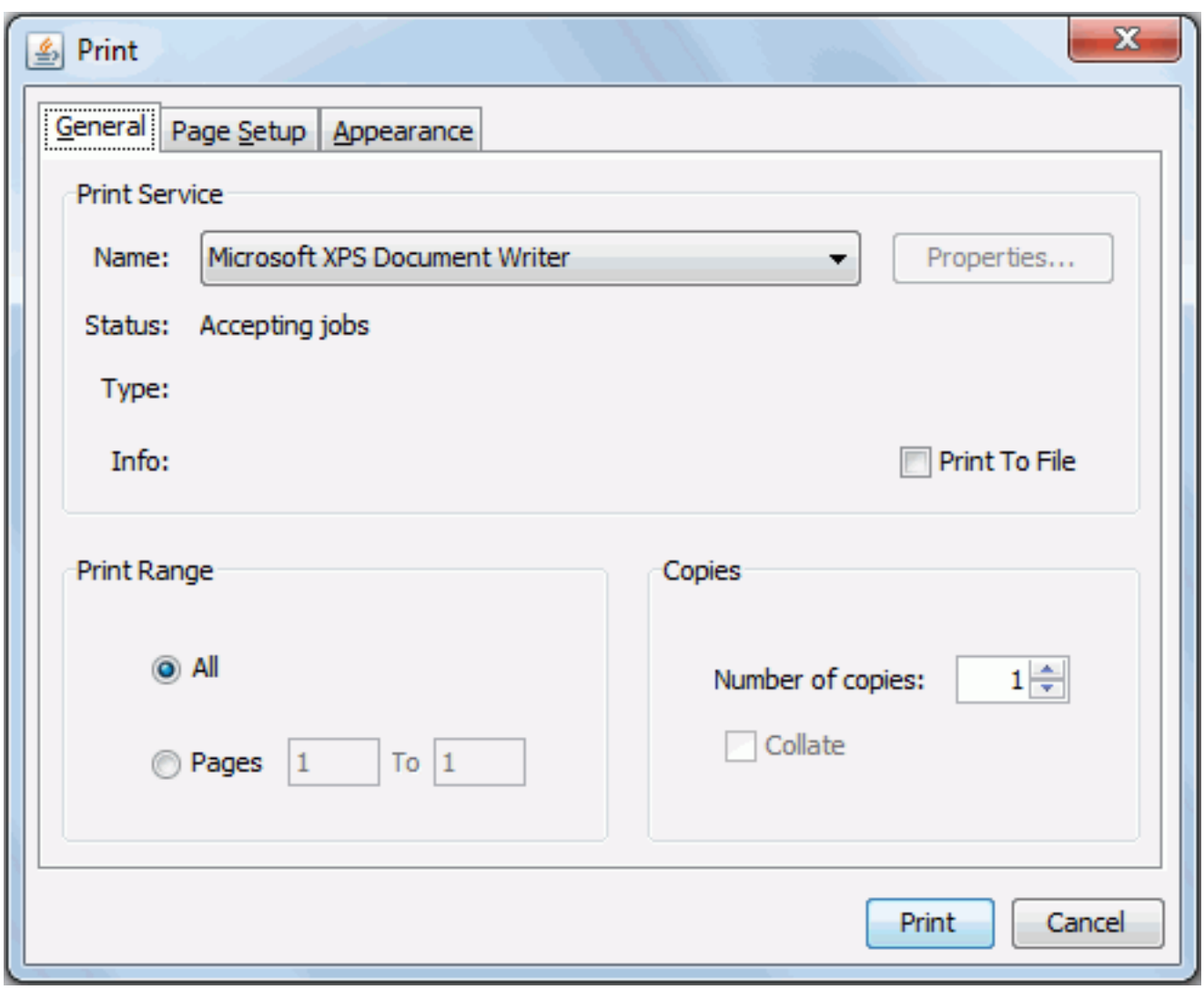
1. In JReport Designer, click **File** > **Options**. The Options dialog appears.
2. Select **Print** in the Category box, and select a method of your choosing from the Print panel. By default, the **Use JDK1.4 printing method** is selected.
3. If you want to separate a large page during printing, select **Separate large pages**. When you print a report with a report page size which is larger than the print paper size, the report will be printed in multiple pages serially. That is, JReport will automatically separate a large page of a report.
4. Click the **Apply** button to save the changes, and then click the **OK** button to close the dialog.

Printing with the JDK 1.4 printing method

Compared with JDK 1.1 and JDK 1.2, the JDK 1.4 printing method provides you with a more diverse selection of options for printing reports, such as paper tray, and color appearance.

To print a report with the JDK 1.4 printing method:

1. Specify the printing method as **Use JDK1.4 printing method** in the Print category of the Options dialog.
2. Open the report you want to print.
3. Click **File** > **Print** on the menu bar, or click the **Print** button on the Standard toolbar. The Print dialog appears.



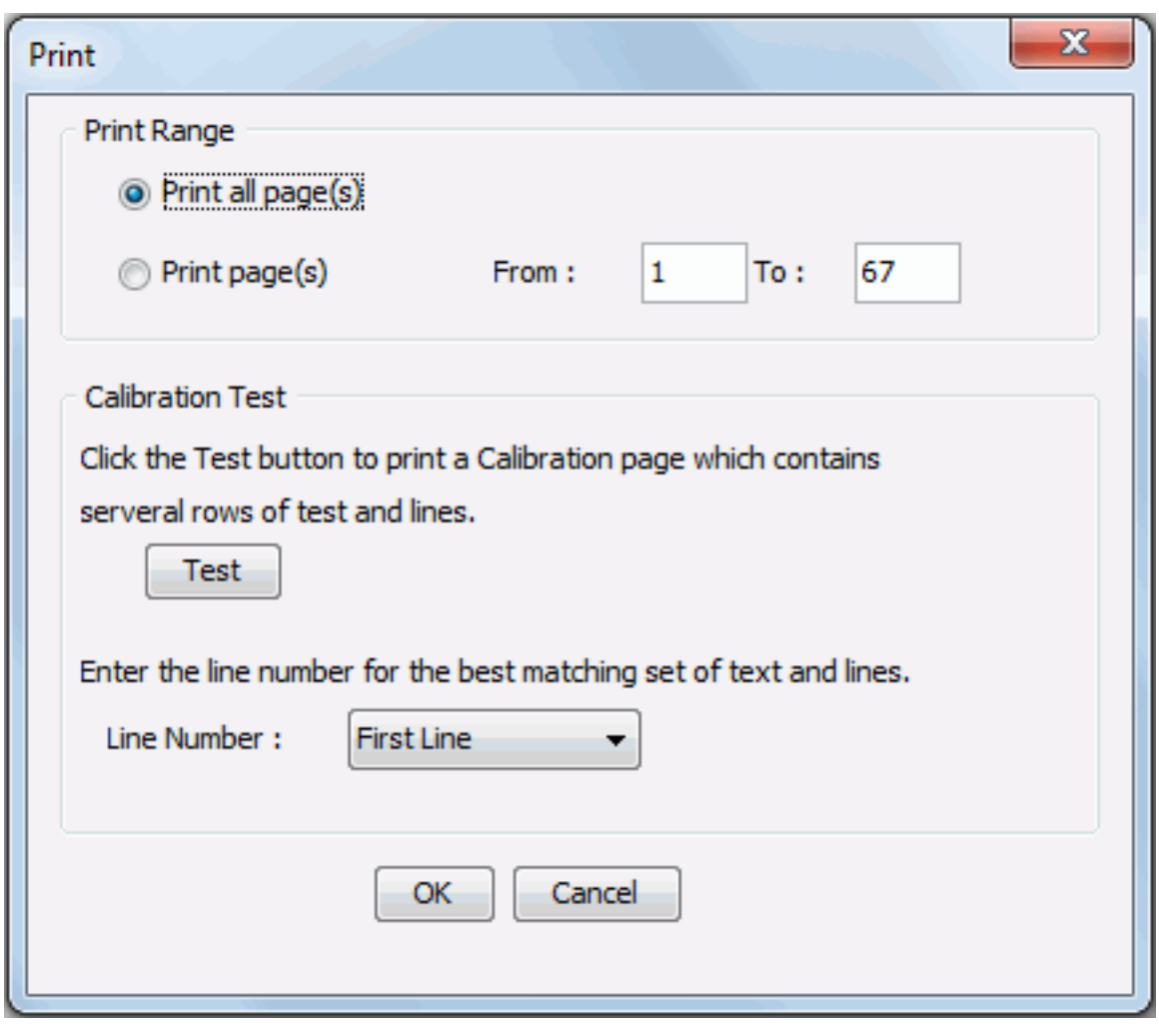
4. Follow the window instructions to specify the settings.
5. Click **Print** to print the report.

Printing with the JDK 1.1 printing method (deprecated)

The JDK 1.1 printing method is quick in speed, but the print quality cannot be guaranteed.

To print a report with the JDK 1.1 printing method:

1. Specify the printing method as **Use JDK1.1 printing method** in the Print category of the Options dialog.
2. Open the report you want to print.
3. Click **File** > **Print** on the menu bar, or click the **Print** button on the Standard toolbar. The Print dialog appears.



In order to overcome the limitation of JDK 1.1.x, JReport is required to be calibrated on the very first print so that the lines and text can be printed correctly. You will only need to calibrate once for each printer that JReport uses. Click the **Test** button, and follow the window instructions.

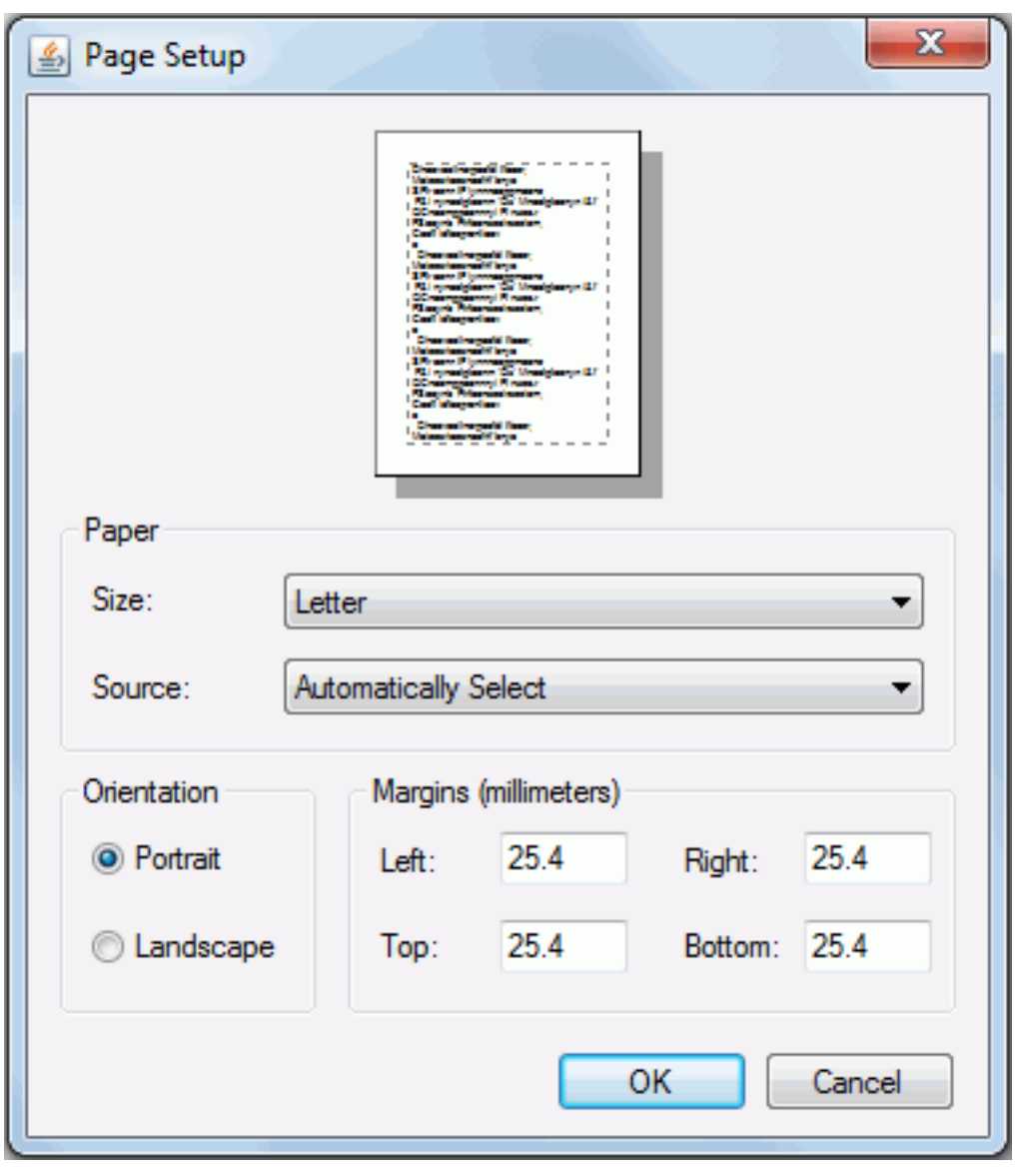
4. Click **OK** to print the report.

Printing with the JDK 1.2 printing method (deprecated)

The JDK 1.2 printing method will provide you with a satisfactory result, even for .gif files, but is slow in speed.

To print a report with the JDK 1.2 printing method:

1. Specify the printing method as **Use JDK1.2 printing method** in the Print category of the Options dialog.
2. Open the report you want to print.
3. Click **File** > **Print** on the menu bar, or click the **Print** button on the Standard toolbar. The Page Setup dialog appears.



4. Specify the settings according to your requirements.
5. Click **OK** to print the report.

Report Result Formats

As you develop reports in JReport Designer, you can print the report results or export them to a file without publishing to JReport Server. For information about publishing reports, see Publishing resources.

This chapter describes the following topics:

- Printing report results
- Exporting report results

Note: When exporting or printing a report which contains parameters, you will be prompted to specify values for the parameters in the Enter Parameter Values dialog.

Using Security Context

Using in the Formula Editor

The Formula Editor now allows you to pass the Security Context to a user-defined formula (UDF) function.

You can call getSecurityContext() in your formula, which will pass the returning object to your UDF function. The method getSecurityContext() returns a DbSecurityContext type object, which provides the method get() to get the Security Context instance transfer from Server or Designer.

DbSecurityContext is a new data type for the SecurityContext object, since the data type of all the variables used in UDF functions should be DbValue.

The following is an example.

```
Import userClass from "UserFunction";
userClass.getData(getSecurityContext(), @country, …);
```

Using in the Deployment Wizard

JReport provides you with two API methods which allow you to bring out your own publishing tool.

DeploymentWizard

- **Package**: com.jinfonet.designer.deployment.api
- **Usage**: DeploymentWizard is an interface used for implementing the customized publishing wizard.
- **Method**: deploy (Frame frame, String catFullPathName, String reporthome).
- **Parameters**:
  - frame - The frame of the deployment box.
  - catalogFullPath - The full path name of the catalog file.
  - reporthome - The report home path.

DeploymentWizardFactory

- **Package**: com.jinfonet.designer.deployment.api
- **Usage**: DeploymentWizardFactory is an interface used for implementing the customized deployment wizard.
- **Method**: getDeploymentWizard()
- **Method description**: Returns the DeploymentWizard.

Working principle

- Loading DeployWizardFactory class name from configuration information:

  In JReport Designer, there is a config.xml file in `<install_root>\bin`. This file corresponds with the Options dialog. You can make it configurable by modifying the XML file. Designer loads information from the XML file and configures the Options dialog when it starts up.

- Getting DeployWizardFactory class name and showing the customized deployment wizard:

  When you click File > Publish and Download > Publish to Server in JReport Designer, it will first check if there is deployWizardFactory in the ReportOption, and will then try to create new instance by the class name. If the class of the customized deployWizard has been successfully loaded, its wizard will be shown. Otherwise, the default publish wizard will be shown.

Using in viewing report

When you click on the View tab to view a report that requires the security context to run, JReport Designer will use the implemented JRSecurityContextFactory to instantiate the SecurityContext instance which will be used to execute the report.

Security Context APIs

JReport provides a set of APIs to support the Security Context feature, including JReport Designer API and JReport Server API.

JReport Designer API

JRSecurityContextFactory

Package: jet.acl.api

Usage: This interface is used to help the user to create an instance of SecurityContext.

Method: getSecurityContext(javax.swing.JFrame frame, java.util.Vector roles)

Parameters:

- frame - The parent window.
- roles - The role list defined by RLS (record-level security).

Returns: The SecurityContext object.

JRSecurityUserDataSource

Package: jet.datasource

Usage: A JRSecurityUserDataSource provides data to JReport for generating reports. The JRSecurityUserDataSource class is developed by users of JReport. It can provide data from a flat file, non-relational database, or application data.

The data returned by this class is in ResultSet object, so users will need to create a ResultSet instance. Then, JReport will use the instance to fetch the data. Users can also create their own ResultSet class.

Method: getResultSet(SecurityContext sc, java.lang.String param)

Method description: Gets the data in ResultSet according to parameters.

Parameters:

- sc - A SecurityContext object, which is implemented by the user.
- param - The parameter that is to be used for UDS.

Returns: The ResultSet object.

JRSecurityHierarchicalDataSource

Package: jet.datasource

Usage: A JRSecurityHierarchicalDataSource provides data to JReport for generating reports. The JRSecurityHierarchicalDataSource class is developed by users of JReport. It can provide data from a flat file, non-relational database, or application data.

The data returned by this class is in JRHierarchicalDataset object, so users will need to create a JRHierarchicalDataset instance. Then, JReport will use the instance to fetch the data. Users can also create their own JRHierarchicalDataset class.

Method: getHierarchicalDataset(SecurityContext sc, java.lang.String param)

Method description: Gets the data in JRSecurityHierarchicalDataset according to the security context and parameters

Parameters:

- sc - A SecurityContext object, which is implemented by the user.
- param - The parameter that is to be used for HDS.

Returns: The JRHierarchicalDataset object.

JReport Server API

SecurityContextFactory

Package: jet.server.api

Usage: SecurityContextFactory is an interface that is used to help you implement security context.

Method: getSecurityContext(java.lang.String realmName, java.lang.String userName, java.lang.String resource, int versionNumber)

Method description: Returns the SecurityContext of a specified resource.

Parameters:

- realmName - The realm name of the current active realm of the server.
- userName - The user name, for a schedule task it is submitter, for an advance run task, it is login user.
- resource - The resource of the server. Now, it is only limited to report resource in the server. For example, `/SampleReports/CustomerAnalysis.cls`
- versionNumber - The version number of the resource.

Returns: The SecurityContext of the specified resource.

SecurityContext

Package: jet.server.api

Usage: SecurityContext is an interface that is used to help you implement security context.

Methods:

- getEmail() - Returns the e-mail address of the user.
- getRoles() - Returns the roles the user belongs to.
- getSalutation() - Returns the salutation of the user (for example, Mr., Mrs., and so on).
- getUserName() - Returns the user name.

Configuring JReport products

Configuring JReport Designer

By default, JReport Designer uses its own security system. If you want to enable the Security Context Support, you will first need to configure the config.xml file in `<designer_install_root>\bin`.

1. Add the elements **`<SecurityContextFactory value=""/>`** and **`<DeployWizardFactory value=""/>`** as the child of the element `<Option>`.

2. Implement classes to the attribute values. For example, part of the config.xml file might be as follows:

```
<Option>
    …

    <SecurityContextFactory value="com.utify.JRSecurityContextFactoryImpl"/>
    <DeployWizardFactory value="com.utify.JRDeployWizardFactoryImpl"/>
</Option>
```

3. Add the root path of the implementing class packages to the ADDCLASSPATH variable of setenv.bat in `<install_root>\bin`.

Configuring JReport Server

Security Context interfaces in JReport Server are different from those of JReport Designer, and you will have to implement the Server interfaces separately.

1. Before you can use the Security Context feature, first configure the server.properties file in `<server_install_root>\bin`.

   Open the server.properites file, and then modify (add if it doesn't exist) the value of the property com.jinfonet.securityContextFactory to be that of your SecurityContextFactory implementing class package and name as follows:

   `com.jinfonet.securityContextFactory = mypackage.myclass`

   > **Note:** Alternatively, you can add the parameter `-D com.jinfonet.securityContextFactory = mypackage.myclass` to the file JRServer.bat with which you start your JReport Server.

2. Add the root path of the implementing class packages to the class path of the file JRServer.bat.

Security Context Support

Security Context Support refers to a new set of functionalities that JReport products provide to support Unify NXJ Security Context. Other than using the security system of JReport, Security Context Support allows Unify NXJ developers to write code that obtains user profile information from the security database. Security Context Support may be useful for any developer who uses Java EE Security.

Click a link from the following to get details:

- Configuring JReport products
- Security Context APIs
- Using Security Context

Example of using member-level security

This example is based on the business view WorldWideSalesBV in the SampleReports.cat catalog file. It is assumed that the server administrator has already created two users on JReport Server: User1 and User2, with the passwords abc and xyz respectively. We will create a member-level security policy on SampleReports.cat to limit the two users' access to different group members of WorldWideSalesBV.

The example is divided into the following tasks:

- Task 1: Set member-level security on business view groups
- Task 2: Create a crosstab and publish it to JReport Server
- Task 3: View the crosstab

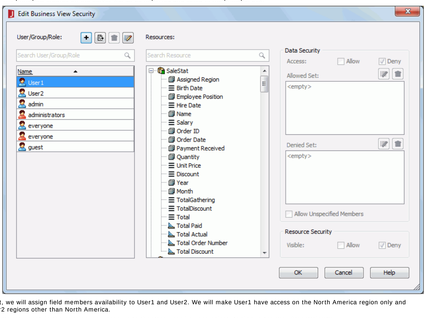
Before starting the example, make sure JReport Server is started.

Task 1: Set member-level security on business view groups

In this task, we will import the two server users to JReport Designer and limit their access to different members of the Region group in WorldWideSalesBV.

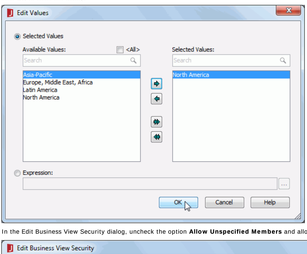
1. In JReport Designer, click **File** > **Catalog Management** > **Open Catalog** on the menu bar.
2. In the Open Catalog File dialog, browse to select the **SampleReports.cat** catalog file in `<install_root>\Demo\Reports\SampleReports`, then click the **Open** button. The Catalog Browser shows up.
3. In the Data tab, go to the **Data Source 1** > **Security** > **Business View Security** node, right-click and select **Edit Business View Security**.
4. In the Edit Business View Security dialog, click and select **Import from JReport Server**.
5. In the Connect to JReport Server dialog, specify the required connection information, then click **Connect**.

   The principals created on the server will all be imported and listed in the principal box.

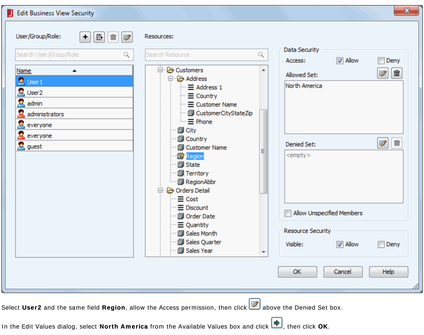


Next, we will assign field members availability to User1 and User2. We will make User1 have access on the North America region only and User2 regions other than North America.

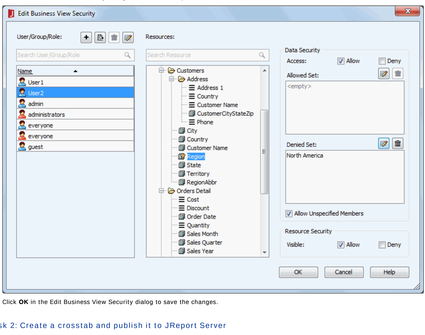
6. Select **User1** in the principal box, then go to the Resources box and select **Region** in the WorldWideSalesBV.
7. In the Data Security box, allow the Access permission.
8. Click above the Allowed Set box.
9. In the Edit Values dialog, select **North America** from the Available Values box and click to add it to the Selected Values box, then click **OK** to close the dialog.



10. In the Edit Business View Security dialog, uncheck the option **Allow Unspecified Members** and allow the Visible permission.



11. Select **User2** and the same field **Region**, allow the Access permission, then click above the Denied Set box.
12. In the Edit Values dialog, select **North America** from the Available Values box and click , then click **OK**.
13. In the Edit Business View Security dialog, allow the Visible permission for User2.

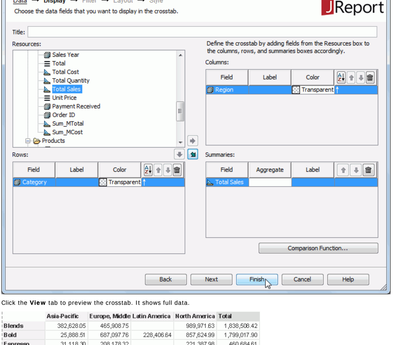


14. Click **OK** in the Edit Business View Security dialog to save the changes.

Task 2: Create a crosstab and publish it to JReport Server

In this task, we will create a crosstab using WorldWideSalesBV and then publish it to JReport Server.

1. In JReport Designer, click **File** > **New** > **Web Report** on the menu bar. A blank web report is created.
2. From the Toolbox panel on the left, drag **Crosstab** into the web report. The Create Crosstab wizard is displayed.
3. In the Data screen, select **WorldWideSalesBV** under Data Source 1, then click **Next**.
4. In the Display screen, select **Region** as the column field, **Category** as the Row field, and **Total Sales** as the summary field, then click **Finish** to create the crosstab.



5. Click the **View** tab to preview the crosstab. It shows full data.

	Asia-Pacific	Europe, Middle	Latin America	North America	Total
Blends	382,628.05	465,908.75		989,971.63	1,838,508.42
Bold	25,888.51	687,097.76	228,406.64	857,624.99	1,799,017.90
Espresso	31,118.30	208,178.32		221,387.98	460,684.61
Exotic	363,663.19	564,144.13	260,368.02	375,226.26	1,563,401.60
Flavored	262,157.40	549,083.00	232,502.40	942,828.65	2,086,372.45
Mild	530,759.95	728,323.95	126,555.80	938,589.91	2,324,229.60
Total	1,596,215.39	3,202,736.92	847,832.86	4,425,430.42	10,072,214.59

6. Go back to the Design tab and click the **Save** button to save the crosstab as **mlscrstab.wls**.

Next, we will publish the crosstab to JReport Server.

7. Click **File** > **Publish and Download** > **Publish to Server** > **Publish Report to Server** on JReport Designer menu bar.
8. In the Connect to JReport Server dialog, specify the required connection information, then click **Connect**.
9. In the Publish to JReport Server dialog, unselect the **All** checkbox and then select the catalog and mlscrstab.wls only.
10. Click the **Browse** button next to the Publish Resource To text field, then in the Select Folder dialog, select **Public Reports** in the Folder box and click **OK**.
11. In the Publish to JReport Server dialog, click the **More Options** button, select **mlscrstab.wls** in the resource box, then on the right panel, switch to the **Permissions** tab.

We will grant the two users the necessary permissions on the web report, so that after the report is published to JReport Server, they are able to run them.

12. Select the **Enable Setting Permissions** checkbox, click the **User** sub tab, select **User1** in the User box and check the **Visible** and **Execute** checkboxes; select **User2** and do the same.
13. Click the **OK** button to publish the resources to JReport Server.

Task 3: View the crosstab

Now, we can log onto JReport Server as User1 and User2 separately to view the crosstab.

1. Log onto the JReport Console page as User1 with the password abc.
2. On the Resources page, open the Public Reports folder and run the web report mlscrstab.wls. The result will be:

	North America	Total
Blends	989,971.63	989,971.63
Bold	857,624.99	857,624.99
Espresso	221,387.98	221,387.98
Exotic	375,226.26	375,226.26
Flavored	1,042,629.65	1,042,629.65
Mild	938,589.91	938,589.91
Total	4,425,430.42	4,425,430.42

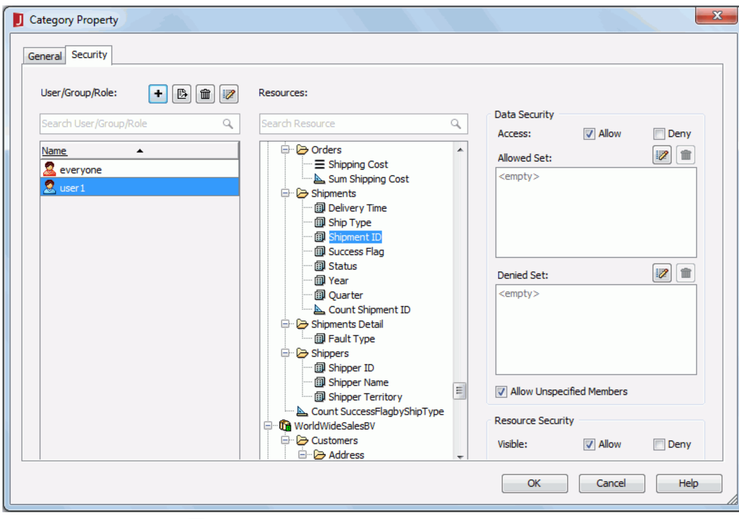
3. Log out and then log in again as User2 with the password xyz. Run the report and the result will be:

	Asia-Pacific	Europe, Middle	Latin America	Total
Blends	382,628.05	465,908.75		848,536.80
Bold	25,888.51	687,097.76	228,406.64	941,392.91
Espresso	31,118.30	208,178.32		239,296.62
Exotic	363,663.19	564,144.13	260,368.02	1,188,175.34
Flavored	262,157.40	549,083.00	232,502.40	1,043,742.80
Mild	530,759.95	728,323.95	126,555.80	1,385,639.69
Total	1,596,215.39	3,202,736.92	847,832.86	5,646,784.17

Configuring member-level security

This document shows how to configure member-level security on business views in a catalog.

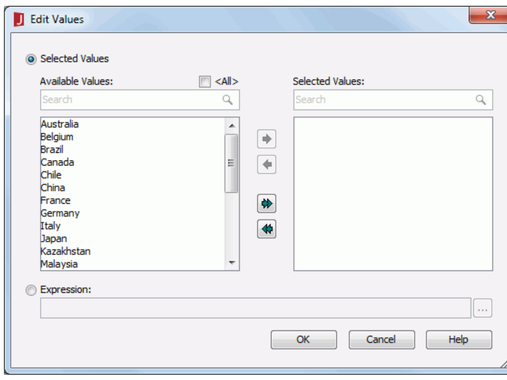
1. In the Catalog Browser, do any of the following:
   - Expand the data source > **Security** node, right-click on **Business View Security** and select **Edit Business View Security** on the shortcut menu. The Edit Business View Security dialog is displayed.
   - Select a business view, expand it, then right-click on the **Business View Security** node and select **Edit Business View Security** on the shortcut menu. The Edit Business View Security dialog is displayed.
   - When creating or editing a business view, click **Tools** > **Security Configuration** on the menu bar. The Edit Business View Security dialog is displayed.
   - When adding or editing a business view element using dialog, switch to the Security tab.



2. In the User/Group/Role panel, click to add users, roles, and groups manually or by importing from an XML file or from JReport Server. If you choose to add users manually in JReport Designer, the users should have been added in the JReport Server security system before the resources involving the member-level security are published to JReport Server so as for the member-level security to take effect.

   To edit a user/role/group, select it and click . In the corresponding edit dialog, edit it as required.

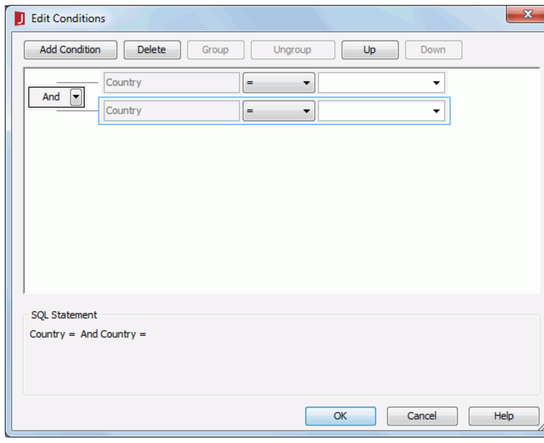
   To delete a user/role/group that is not wanted, select it and click .
3. Make a user/role/group selected.
4. In the Resources box, select the field to which you want to set the principal permission.
5. In the Data Security box, specify whether the principal can access values of the field.
6. If the selected field is a group field, you can specify which members of the field are allowed and which are denied for the principal.
   a. Click above the Allowed Set or Denied Set box. The Edit Values dialog appears.



   b. Choose a way of specifying the members: select members from the available list, or compose an expression to retrieve members. Only one can be used.
      - To select members, first check the **Selected Values** radio button.

        If you would like all the possible members of the field to be selected, select **<All>**.

        If you just want to select some of the members, leave <All> unchecked. Add them one by one by selecting one and then clicking . You can make use of the quick search toolbar to search for the required members.
      - To compose an expression, check the **Expression** radio button, then click . The Edit Conditions dialog appears.



        Click the **Add Condition** button to add a condition line. Choose the operator with which to compose the condition expression from the operator drop-down list. From the value drop-down list, specify the value of how to build the condition. You can also type in the value manually. Click **Add Condition** to add more condition lines and define the relationship between the condition lines.

        To make some conditions grouped, select them and click the **Group** button, then the selected conditions will be added in one group and work as one line of filter expression. Conditions and groups together can be further grouped. To take any condition or group in a group out, select it and click **Ungroup.**

        To adjust the priority of the conditions, select it and click the **Up** or **Down** button.

        To delete a condition line, select it and click the **Delete** button.
   c. Click **OK** to save the values.
   d. If a user is selected, check whether the unspecified members are available to the user.
7. In the Resource Security box, specify whether you want the field to be visible to the principal.
8. Repeat steps 4 to 7 to customize the principal's permissions on other fields.
9. Repeat steps 3 to 7 to customize other principals' permissions on different fields.
10. When finished, click **OK** in the dialog.
11. Save the catalog.

    When you save the catalog, the permission settings are also saved and they are saved in an authorization file in the same folder as the catalog file. The catalog and authorization files have the same file name but different extensions, for example, if the catalog file is test.cat, the authorization file will be named test.auth. The authorization file is loaded by the view authorization manager of its catalog during runtime.

See also Edit Business View Security dialog for addition information about options in the dialogs.

Note: When a catalog with member-level security is published to JReport Server, only the principals on the server which match the principals defined in the member-level security will maintain the member-level security setting. When a principal is deleted from the server security system, the related member-level security setting in all catalogs will be removed.

Permission logic

The relationship between a principal (user, role or group) and the members of a group are classified into three groups:

- **Allowed Set**
  The members of a group visible to a principal, which are directly specified to the principal, excluding those inherited from the parent roles or groups.
- **Denied Set**
  The members of a group invisible to a principal, which are directly specified to the principal, excluding those inherited from the parent roles or groups.
- **Unspecified members**
  The members of a group that are not specified in a user's allowed/denied set or in the inherited allowed/denied set from the user's parent roles/groups. Report designer can determine whether the unspecified members are visible to a user.

A principal can have its own allowed/denied set and inherit the allowed/denied sets from its parent roles or groups. The parent allowed/denied sets will be calculated first and it is a recursive process.

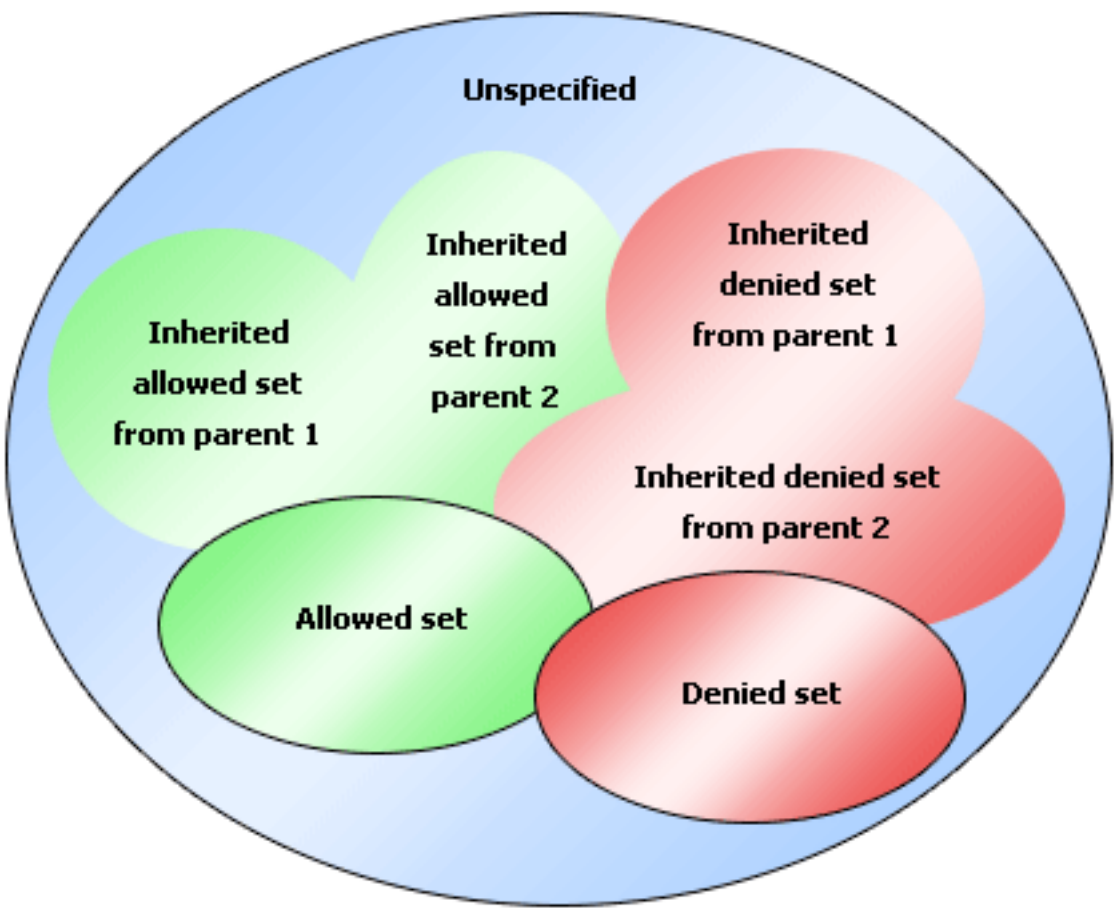
The following are the detailed working logic:

- The directly defined permission on a principal always overwrites the permission inherited from its parents. That is, for a member of a group, if its permission has not been defined on a given user/role/group, the permission will inherit from the parents of the user/role/group; if it is defined, the permission will be used directly.
- If for a user/role/group a member exists in both the allowed set and denied set, the denied set has higher priority than the allowed set. In this case the member is not available to the user/role/group.
- Inheritance rule:
  - If a member is denied by any of a principal's parents, it will be denied, or else if a member is allowed by any of a principal's parents, it will be allowed.
  - If a member is neither denied nor allowed by any parent, the permission is regarded as unspecified.
- After inheritance rule is applied to a specific user, if some members are still unspecified, the status of the members will depend on the option setting whether to make them allowed or denied.

Here is the priority order:

Denied Set > Allowed Set > Inherited Denied Set > Inherited Allowed Set > Unspecified (no matter whether it is allowed or denied).

See the diagram below:



The final result of the members of a group that a principal is allowed to access will be:

({Allowed Set} - {Denied Set}) U
(({Inherited Allowed Set from parent 1} U {Inherited Allowed Set from parent 2} U ...) - ({Inherited Denied Set from parent 1} U {Inherited Denied Set from parent 2} U ...)) U
{Unspecified if allowed}

The final set of the unspecified will be:

{All} - ({Allowed Set} U {Denied Set} U {Inherited Allowed Set from parent 1} U {Inherited Denied Set from parent 1} U {Inherited Allowed Set from parent 2} U ...)

For a user, the member-level security in a business view would be like this:

{Accessible members of group1} And {Accessible members of group1} And...

Note: If all members are denied in one group regardless of whether it is set to a principal directly or to the parents indirectly, no data will be retrieved from this business view to the principal.

We will take some examples to further demontrate the relationship between a principal and the group members.

Example 1

Here we use a simple sample to describe a case when a user belongs to multiple roles.

Assume there is a group Order ID={1,2,3,4,5,6,7,8,9}, and we set member-level security on this group for a user (user1) and two roles (role1, role2) separately.

	Belong to	Allowed Set	Denied Set
user1	role1, role2	1	
role1		2,3	4,5
role2		3,4,5	1,2

The unspecified members will be {1,2,3,4,5,6,7,8,9} - ({1}U{2,3}U{4,5}U{3,4,5}U{1,2}) = {6,7,8,9}.

Assume we set the property "Allow Unspecified Members" to true, which means the unspecified members {6,7,8,9} are allowed to the user.

The final result that user1 can see when only this member-level security is taking effect will be:

{1} U (({2,3}U{3,4,5}) - ({4,5}U{1,2})) U {6,7,8,9}={1,3,6,7,8,9}

Example 2

This example is more complex. It contains four properties of the member-level security setting.

Assume there is a summary table with three groups and summary.

Region	Country	City	Summary (the count of Order_ID)
APAC			41
	Australia		20
		Sydney	20
	China		21
		Beijing	9
		Hongkong	4
		Shanghai	8

And in below table, we list the member-level security setting in the left cells, and the final result a user will get are listed in the right cells.

Member-level security specified as below

Group	Allowed Set	Denied Set	Allow Unspecified Members
Region	<empty>	<empty>	True
Country	<empty>	{China}	True
City	<empty>	<empty>	True

User will get such result

Region	Country	City	Summary (count Order_ID)
APAC			20
	Australia		20
		Sydney	20

Member-level security specified as below

Group	Allowed Set	Denied Set	Allow Unspecified Members
Region	<empty>	<empty>	True
Country	{China}	<empty>	False
City	<empty>	{Beijing,Shanghai}	True

User will get such result

Region	Country	City	Summary (count Order_ID)
APAC			4
	China		4
		Hongkong	4

Member-level security specified as below

Group	Allowed Set	Denied Set	Allow Unspecified Members
Region	<empty>	<empty>	True
Country	{China}	<empty>	False
City	<empty>	{Beijing,Shanghai}	False

User will get such result

Region	Country	City	Summary (count Order_ID)
			0

Member-level security

Member-level security (MLS) enables report designers to limit user access to the specific members of groups in business views. By defining which members of a group are available to which users, groups, or roles, report results are created for each user, role and group. When a user accesses a report, JReport checks the user, group and role of the user and merges the data in the report the user is authorized to see and displays it to the user.

Note: JReport Designer does not check member-level security when getting data from business views.

This section explains member-level security in the following topics:

- Permission logic
- Configuring member-level security
- Example of using member-level security

Report-scope security

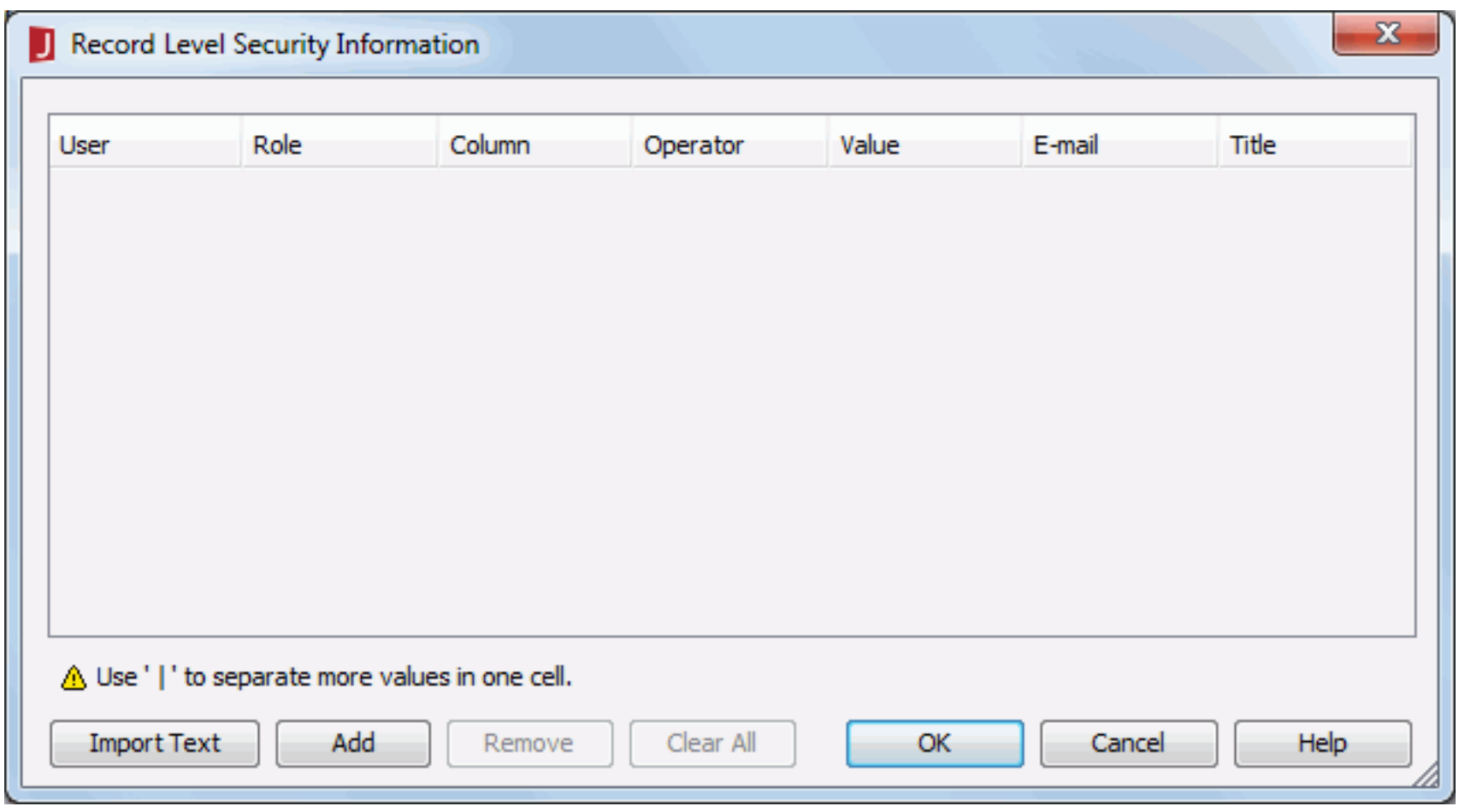
The record-level security (RLS) can be of report scope, which is based on the security information file. That is, you can use the security information file to set the security policies for a report. Report-scope security policy doesn't support column-level security (CLS).

Report-scope security has a higher priority than connection-scope security. That is, if in a report, both connection-scope and report-scope security have been applied, only the report-scope security will take effect.

Setting up a RLS policy for a report

The following example shows you how to set up a record-level security policy for a report:

1. Click **File** > **Open** in JReport Designer.
2. In the Open Report dialog, click the **Browse** button to open the catalog file **SampleReports.cat** in `<install_root>\Demo\Reports\SampleReports`, then open the sample report **EmployeeInformation.cls**.
3. In the Report Inspector, select **EmployeeInformation** in the Datasets node.
4. In the Security section of the Properties panel, click ... to the right of the Record Security property entry to open the Record Level Security Information dialog.



5. Click the **Add** button to add a condition line and edit the security information as required by clicking corresponding cells.

   Since the record-level security policies are file-based, you can just create a new text file, add the security settings, and then link the file to your report. Below are steps detailing how to import the security information from a text file.

   a. In the Record Level Security Information dialog, click the **Import Text** button.
   b. In the Select Record Level Security File dialog, specify the text file that contains the predefined security information and click **Open**.
   c. Information set in the text file will be displayed in the Record Level Security Information dialog. Click the cells to edit them if required.
   d. Click **OK** to make the security policy applied.
6. Click **OK** to close the dialog.
7. Click the **View** tab to view the report. You will then be prompted to provide security ID (SID), which refers to the user name. Type the SID, and then click **OK**. Only the records which satisfy the condition will be displayed.

After you have set up RLS for reports in JReport Designer, you can then publish them to JReport Server as normal. Then, when you log onto JReport Server as different users, you will find that the record-level security settings have been applied to the report. Different users will only see the data that they are supposed to see.

Notes:

- In the Record Level Security Information dialog, the data values provided in the drop-down list may not be valid for your database, because they are data values that have already been reformatted using your date format settings in JReport. For detailed information on how to set the date format in JReport Designer, see the Date Format explanation of the document Get JDBC Connection Information dialog.
- If you want to use more values in one cell, use "|" to separate them (applies to User, E-mail, and Title column). This is useful when you want to apply the same conditions to multiple users. For example, if you want user1, user2, and user3 to share the same security setting, list user1, user2, and user3 in the same cell, separated by "|", and then define the security conditions.
- Currently, you cannot save the security information to a file.
- If you want to apply more than one condition expression to a user, you can edit the condition expressions in several individual lines, typing in the same user name for each line. For such compound conditions, the relationship among them is logical OR.

  For example, both Customer ID >= 10 and Customer Name = 'Absolute Java' together will cause JReport Designer to retrieve records where customer ID is larger than or equal to 10 and the record with customer name equal to Absolute Java.

  If you want to use logical AND for the relationship between each condition expression, use a formula that returns a Boolean value to filter the records. To specify the formula name for a report, in the Properties panel, select or type the formula name in the Function property entry.

  For example, if user1 should only see records which satisfy the condition State = 'CO' and Customer ID >= 10, the formula should be:
  `if ( @State == 'CO' && @"Customer ID">= 10 ) return "user1"`

  If both the Record Level Security Information dialog and a formula have been used to control the security information, both will take effect. The relationship between them is logical OR.
- When creating the text file to include the security information, use TAB to separate each column, and always keep the headings (User, Role, Column, and so on) in the first line of the text file. For example,

User	Role	Column	Operator	Value	E-mail	Title
admin	admin					
user1		Customer ID	>=	10	user1@yoursite.com	Mr.
user1		Customer Name	=	'Absolute Java'		
user2		Customer Name	IN	('Absolute Java','American Coffee Inc.')	user2@yoursite.com	Miss
user3		Phone	IN	('(212) 555-3462','(317) 555-1274')	user3@yoursite.com	Mrs.

Applying a connection-scope security policy to a report

To apply a connection-scope security policy to a report, follow the steps below:

1. In JReport Designer, open the report to which you want to apply the security policy.
2. In the Report Inspector, select the node that represents the dataset the report uses in the Datasets node.
3. In the Security section of the Properties panel, select a required security policy for the Security Policy Name property.
4. Preview the report and the Security Identifier dialog will pop up.
5. Type the user SID, then click **OK** to see the result. You will not be able to view any records if the user SID you enter doesn't exist in the defined security policy.

After you have applied a connection-scope security policy for a report in JReport Designer, you can then publish it to JReport Server as normal. Then when you log onto JReport Server as different users, you will find that the security settings have been applied to the report. That is, different users will only see the data they are supposed to see. However, as the user defined users and roles in a connection-scope security policy may not be recognized by JReport Server, if your security policy contains such users/roles, you need to first create these users and roles respectively on JReport Server, and then in JReport Designer, synchronize the security information with the server by means of importing security information from the sever with the Merge option checked.

Notes:

- Column-level security (CLS) is not supported on library component at present, so for a library component, only the data source security policies which have not been defined with CLS will be available in the value list of the Security Policy Name property.
- Security policies for the primary report and subreport can be set and applied individually, with different combinations causing different results. The relationships between the security policy settings and the try-viewing results are shown in the following table:

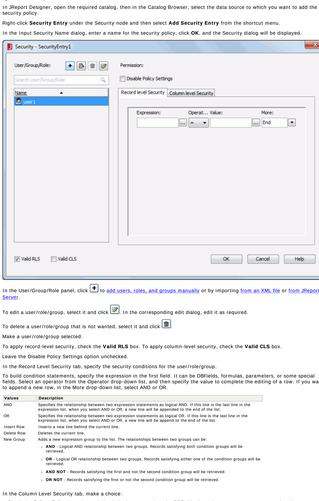
main	sub	Users in Policy	Users NOT in Policy
Ð	Ð	No security policy is applied to either the primary report or the subreport. Users can therefore view all records in both reports.	
Ï	Ð	Can view specified records in the primary report and all records in the subreport.	Can view no records in the primary report and all records in the subreport.
Ð	Ï	Can view all records in the primary report and specified records in the subreport.	Can view all records in the primary report and no records in the subreport.
Ï	Ï	Can view specified records in both the primary report and the subreport.	Can view no records in either the primary report or the subreport.

Ï - Security policy has been set and applied.
Ð - Security policy has not been set and applied.

Setting up a connection-scope security policy

To set up a connection-scope security policy, or in other words, to add a data source security entry to a data source connection in a catalog, follow the steps below:

1. In JReport Designer, open the required catalog, then in the Catalog Browser, select the data source to which you want to add the security policy.
2. Right-click **Security Entry** under the Security node and then select **Add Security Entry** from the shortcut menu.
3. In the Input Security Name dialog, enter a name for the security policy, click **OK**, and the Security dialog will be displayed.



4. In the User/Group/Role panel, click to add users, roles, and groups manually or by importing from an XML file or from JReport Server.

   To edit a user/role/group, select it and click in the corresponding edit dialog, edit it as required.

   To delete a user/role/group that is not wanted, select it and click .
5. Make a user/role/group selected.
6. To apply record-level security, check the **Valid RLS** box. To apply column-level security, check the **Valid CLS** box.
7. Leave the Disable Policy Settings option unchecked.
8. In the Record Level Security tab, specify the security conditions for the user/role/group.

   To build condition statements, specify the expression in the first field. It can be DBfields, formulas, parameters, or some special fields. Select an operator from the Operator drop-down list, and then specify the value to complete the editing of a row. If you want to append a new row, in the More drop-down list, select AND or OR.

Values	Description
AND	Specifies the relationship between two expression statements as logical AND. If this line is the last line in the expression list, when you select AND or OR, a new line will be appended to the end of the list.
OR	Specifies the relationship between two expression statements as logical OR. If this line is the last line in the expression list, when you select AND or OR, a new line will be appended to the end of the list.
Insert Row	Inserts a new line behind the current line.
Delete Row	Deletes the current line.
New Group	Adds a new expression group to the list. The relationships between two groups can be: - **AND** - Logical AND relationship between two groups. Records satisfying both condition groups will be retrieved. - **OR** - Logical OR relationship between two groups. Records satisfying either one of the condition groups will be retrieved. - **AND NOT** - Records satisfying the first and not the second condition group will be retrieved. - **OR NOT** - Records satisfying the first or not the second condition group will be retrieved.

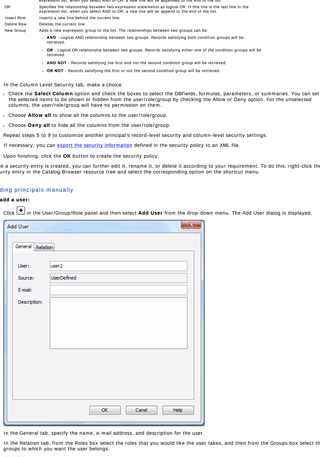
9. In the Column Level Security tab, make a choice:
   - Check the **Select Column** option and check the boxes to select the DBfields, formulas, parameters, or summaries. You can set the selected items to be shown or hidden from the user/role/group by checking the Allow or Deny option. For the unselected columns, the user/role/group will have no permission on them.
   - Choose **Allow all** to show all the columns to the user/role/group.
   - Choose **Deny all** to hide all the columns from the user/role/group.
10. Repeat steps 5 to 9 to customize another principal's record-level security and column-level security settings.
11. If necessary, you can export the security information defined in the security policy to an XML file.
12. Upon finishing, click the **OK** button to create the security policy.

Once a security entry is created, you can further edit it, rename it, or delete it according to your requirement. To do this, right-click the security entry in the Catalog Browser resource tree and select the corresponding option on the shortcut menu.

Adding principals manually

To add a user:

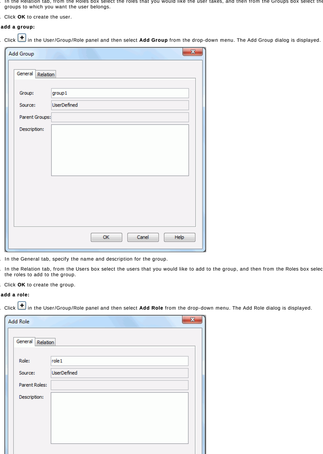
1. Click in the User/Group/Role panel and then select **Add User** from the drop-down menu. The Add User dialog is displayed.



2. In the General tab, specify the name, e-mail address, and description for the user.
3. In the Relation tab, from the Roles box select the roles that you would like the user takes, and then from the Groups box select the groups to which you want the user belongs.
4. Click **OK** to create the user.

To add a group:

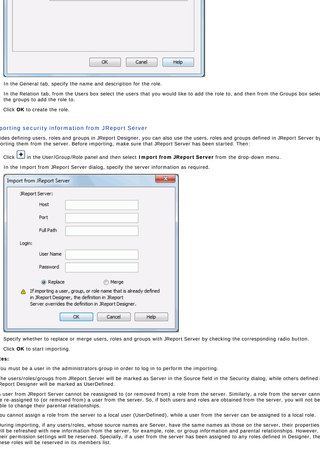
1. Click in the User/Group/Role panel and then select **Add Group** from the drop-down menu. The Add Group dialog is displayed.



2. In the General tab, specify the name and description for the group.
3. In the Relation tab, from the Users box select the users that you would like to add to the group, and then from the Roles box select the roles to add to the group.
4. Click **OK** to create the group.

To add a role:

1. Click in the User/Group/Role panel and then select **Add Role** from the drop-down menu. The Add Role dialog is displayed.

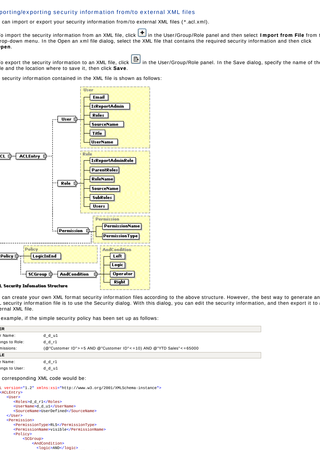


2. In the General tab, specify the name and description for the role.
3. In the Relation tab, from the Users box select the users that you would like to add the role to, and then from the Groups box select the groups to add the role to.
4. Click **OK** to create the role.

Importing security information from JReport Server

Besides defining users, roles and groups in JReport Designer, you can also use the users, roles and groups defined in JReport Server by importing them from the server. Before importing, make sure that JReport Server has been started. Then:

1. Click in the User/Group/Role panel and then select **Import from JReport Server** from the drop-down menu.
2. In the import from JReport Server dialog, specify the server information as required.



3. Specify whether to replace or merge users, roles and groups with JReport Server by checking the corresponding radio button.
4. Click **OK** to start importing.

Notes:

- You must be a user in the administrators group in order to log in to perform the importing.
- The users/roles/groups from JReport Server will be marked as Server in the Source field in the Security dialog, while others defined in JReport Designer will be marked as UserDefined.
- A user from JReport Server cannot be reassigned to (or removed from) a role from the server. Similarly, a role from the server cannot be re-assigned to (or removed from) a user from the server. So, if both users and roles are obtained from the server, you will not be able to change their parental relationships.

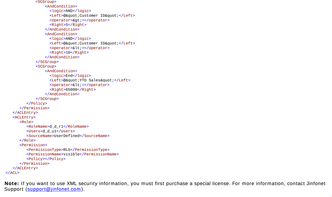
  You cannot assign a role from the server to a local user (UserDefined); while a user from the server can be assigned to a local role.
- During importing, if any users/roles, whose source names are Server, have the same names as those on the server, their properties will be refreshed with new information from the server, for example, role, or group information and parental relationships. However, their permission settings will be reserved. Specially, if a user from the server has been assigned to any roles defined in Designer, then these roles will be reserved in its members list.

Importing/exporting security information from/to external XML files

You can import or export your security information from/to external XML files (*.apt.xml):

- To import the security information from an XML file, click in the User/Group/Role panel and then select **Import from File** from the drop-down menu. In the Open an xml file dialog, select the XML file that contains the required security information and then click **Open**.
- To export the security information to an XML file, click in the User/Group/Role panel. In the Save dialog, specify the name of the file and the location where to save it, then click **Save**.

The security information contained in the XML file is shown as follows:



XML Security Information Structure:

You can create your own XML format security information files according to the above structure. However, the best way to generate an XML security information file is to use the Security dialog. With this dialog, you can edit the security information, and then export it to an external XML file.

For example, if the simple security policy has been set up as follows:

```
USER
User Name:                    d_d_s5
Belongs to Role:              d_d_r5
Permissions:                  (@"Customer ID"+5 AND @"Customer ID"<10) AND @"YTD Sales"<=65000
ROLE
Role Name:                    d_d_r5
Belongs to User:              d_d_s5
```

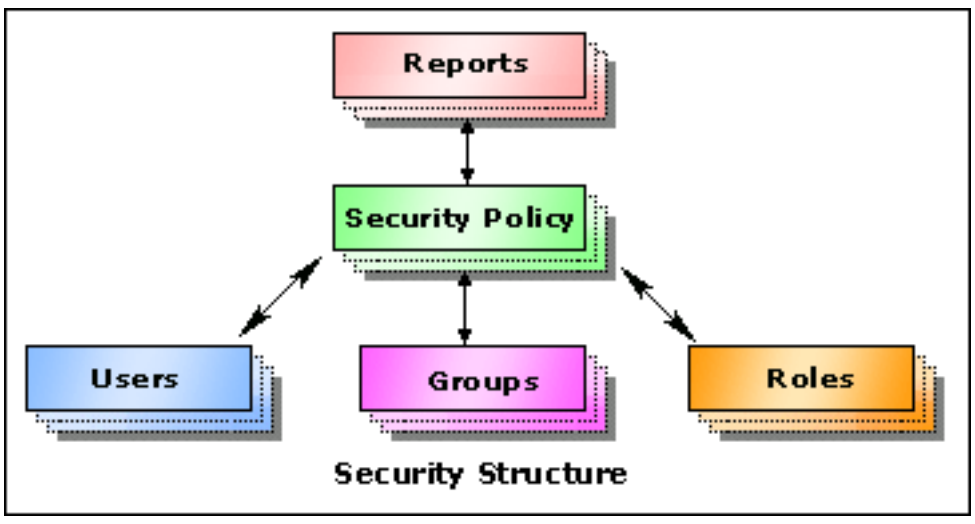
The corresponding XML code would be:

```
<?xml version="1.2" xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">
  <User_Role_Set>
    <User>
      <Selected_d_r5></Selected>
      <Name>d_d_s5</Name></User>
      <SourceName>UserDefined</SourceName>
    </User>
    <Permission>
      <PermissionTypeID_RLS></PermissionTypeID>
      <PermissionName>yzzb15a</PermissionName>
      <Policy_Set>
        <ACL_Set>
          <ACLEntry_Set>
            <ACLEntry_Id_d_r5></ACLEntry_Id_d_r5>
            <Policy_Set_Id_d_r5>
              <Policy_Id_d_r5>
              <Policy_Name_d_r5>
              <Policy_Type_d_r5>
              <Policy_Source_d_r5>
              <Policy_Source_Name_d_r5>
              <Policy_Source_Type_d_r5>
              
```

Connection-scope security

A connection-scope security policy is used to refer to a data source security entry in a catalog. It allows you to control user access to different subsets of data, and ensures that people only see what they are supposed to see: certain records and/or columns.

If you want to implement the same security policy in a group of reports, you can simply apply an existing security policy to the reports, without having to repeatedly build the security information for each report. The following diagram illustrates the security structure of the connection-scope security policy.



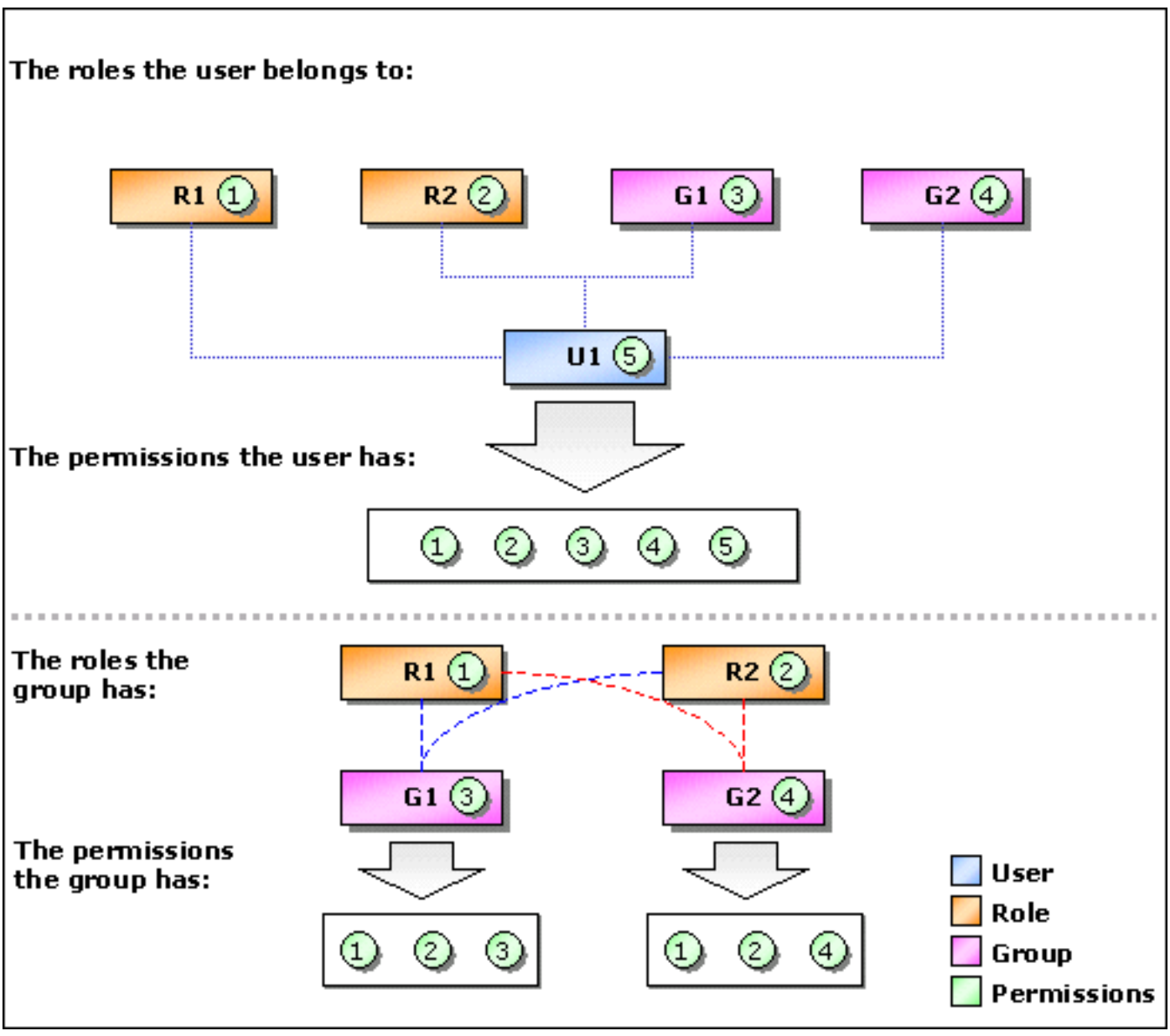
- **User**
  The users for connection-scope security fall into two categories: User Defined and Server. The user defined user is created in JReport Designer, while the server user is imported from JReport Server.

- **Role**
  The roles for connection-scope security fall into two categories: User Defined and Server. The user defined role is created in JReport Designer, while the server role is imported from JReport Server.

- **Group**
  The groups for connection-scope security fall into two categories: User Defined and Server. The user defined group is created in JReport Designer, while the server group is imported from JReport Server.

- **Security Policy**
  Permissions can be assigned to users, roles, and groups. A role/group can have more than one user, and a user can belong to more than one role/group. The permissions that a user may have depends on the permissions of the roles/groups it belongs to and its own permissions. The inheritance relationship is logic OR.

  For example, if user U1 belongs to two roles and two groups, such as R1, R2, G1, and G2. The permissions he/she will have are the permissions collections of the roles, groups and itself, which is P1, P2, P3, P4, and P5, as shown in the following diagram.



  The group and role relationships are the same. Here is another example, if group G1 has two roles, R1 and R2, together with its own permissions, the users in group G1 will have P1, P2, and P3, and also the permissions assigned directly to the users.

  When calculating a user's permissions, the permission collections of its roles and groups are first calculated using logical OR. If the calculation result conflicts with the user permissions, the user's permissions will have priority. That is, if one access right is permitted for the user's groups and roles but denied by the user's permissions, the user will not be permitted to have this access right.

The major steps for using RLS and CLS with a connection-scope security policy are as follows:

1. Set up the security policy.

   You use this step to set up the security policies for different report categories. The predefined security policies work within a connection range.

2. Apply the security policy to your reports.

   After you have defined the security policies, you can then apply them to your reports respectively.

3. Publish the reports with RLS (CLS) to JReport Server and run them.

Pick a task from the following:

- Setting up a connection-scope security policy
- Applying a connection-scope security policy to a report

Record-level and column-level security

The record-level security (RLS) and column-level security (CLS) of JReport Designer allow you to control user access to different subsets of data and ensure that people only see what they are supposed to see: record-level security allows you to define which records are to be revealed to any given user, while column-level security allows you to define which report column is revealed to any given user. This enables you to provide different users with accordingly different, but appropriate contents. No matter to whom you need to provide information, a plant manager or thousands of customers, JReport Designer allows you to control access to information according to your requirements.

JReport Designer have two types of the security policies, one is a security policy based on a data source connection (connection-scope security policy), and the other is a security policy based on a single report (report-scope security policy). Record-level security can be applied simultaneously to both connection and report scopes. However, if a report-scope security policy has already been applied to a report, it will override a connection-scope security policy applied to the report. That is, report-scope security policies have a higher priority than connection-scope security policies.

Pick a topic from the following to get details about the security policy:

- Connection-scope security
- Report-scope security

Cached report bursting

Cached report bursting is a security mechanism for controlling access to data at the group level in a report. By defining which groups of data are available to which users, groups, or roles, report results are created for each user, role and group. When a user accesses the report result, JReport checks the user, group and role of the user and merges the groups of data in the report the user is authorized to see and displays it to the user. Cached report bursting is implemented with these security properties on the group panel: Cascade, Grant, Groups, and Roles. The feature enables different users to view different data groups according to their access privileges. It also applies to nested groups.

Setting up a cached report bursting policy for a report

The Cascade, Grant, Groups, and Roles properties are provided for building a report with cached report bursting. For details about the properties, see Security.

Below is an example which illustrates how to set cached report bursting:

1. In JReport Designer, create a banded report for customer information which is grouped by the Country field.
2. Create a formula to control the Grant property value. This formula returns a String value indicating which user will have the privilege to access which group of data.

   Here, we write the formula Burst_User to set the security identifier as follows:

   ```
   if (@Country == "China" || @Country == "Canada")
   return "admin";
   if (@Country == "USA")
   return "jennifer";
   ```

   "admin" and "jennifer" are two users assigned by the JReport Server administrator. The above formula states that, the user "admin" is authorized to view only the China and Canada groups, while the user "jennifer" can only view the USA group. If the formula is written as follows:

   ```
   if ( @Country =="USA")
   return "user1|user2|user3";
   ```

   Then, user1, user2, and user3 can view the USA group.

   > **Note:** The returned value specified in the formula must be the same as the user ID defined in JReport Server. Also, the user ID is case sensitive.

3. Create a formula Burst_Group to control the Groups property value. This formula returns a String value indicating which group of users will have the privilege to access which group of data.

   ```
   if(@Country=='Italy')
   return 'group1';
   if(@Country=='USA')
   return 'group2';
   ```

4. Create a formula Burst_Role to control the Roles property value. This formula returns a String value indicating which role will have the privilege to access which group of data.

   ```
   if(@Country=='Japan')
   return 'role1';
   if(@Country=='USA')
   return 'role2';
   ```

5. In the Report Inspector, select the node that represents the group. Then, in the Security section of the Properties panel, set Cascade to **true**, the Grant property's value to the formula **Burst_User**, Groups to **Burst_Group**, and Roles to **Burst_Role**.

Since the control of report access is not possible without user ID, the significance of this function is only apparent when other users try to access a report after it has been published to JReport Server. When an end user views a report with cached report bursting on JReport Server, the corresponding group will be displayed according to the security identifier.

In the above example, if "admin" belongs to group1 and role1, he will be able to view the China, Canada, Italy, and Japan groups. jennifer will be able to view only the USA group if she belongs to group2 and role2.

Security

Security in a report is a kind of privilege control. This chapter demonstrates controlling access to reports and different subsets of data by means of defining report security. No matter to whom you need to provide information, JReport Designer allows you to control access to it according to your requirements.

Generally, report security can be classified into three groups: cached report bursting, record level and column level. After which, users will then only see what the access privileges allow.

Security in this chapter is introduced as follows:

- Cached report bursting
- Record-level and column-level security
- Member-level security
- Security Context Support

Inheriting the style from a data container

The four types of data containers - table, crosstab, chart, and banded object in a page report - can remember the style applied to them and therefore their style can be inherited by another component or a new segment inserted into them.

When inserting a component or a new segment into a table, crosstab, chart, or banded object, by default the former will inherit the style from the latter. However, when inserting a table, crosstab, chart, or banded object via the component creation dialog into a table or banded object, JReport provides you with the opportunity to cancel the way of inheriting style and instead specify a new style to the component. In this case, the inserted component reserves its own style independent from the style of the parent data container.

An option Inherit Style is provided for enabling a table, crosstab, chart, or banded object to inherit the style from a parent table or banded object. It is available:

- In the component creation wizard when inserting a table, crosstab, chart, or banded object into a table or banded object.
- In the Style screen of the wizard of a table, crosstab, chart, or banded object which is in a table or banded object.
- In the style list on the toolbar. The Inherit Style here only works on the objects which are in a table, crosstab, chart, or banded object.

When modifying the style of one of the four types of data containers, if the child component's style type is Inherit Style, it will change to apply the new style accordingly. However if the child component has a style, it will reserve its own style.

Applying styles

You can apply a style in the process of creating a report or at any time when designing a report. During report design, if you have applied more than one style to a component, the last style always covers the previous style.

The following introduces all the ways of specifying a style. Choose the appropriate methods available to you.

- **Using wizards**

  Report wizards contain a Style screen for you to specify a style.

- **Using the style drop-down list on the toolbar**

  The style drop-down list

on the Standard toolbar provides a list of styles for choosing. In this way you can apply a style to one or more selected objects.

- **Using the Report Inspector**

  There is a Style Group property in the Others category of the Report Inspector for a report tab or a web report. It allows you to specify an XSD style.

Notes:

- There are 15 sets of demo XSD styles provided by JReport, which can be applied to crosstabs, tables, banded objects and DBFields converted from previous versions. Other components can only apply the XSD styles which are created by yourself. XSD styles are primarily for compatibility with previous versions.

- By default, the demo XSD styles are not included in either the Style box of the report wizards or the style drop-down list on the toolbar. If you want to have them displayed, uncheck the option "Only CSS styles available in the style list" in the Component category of the Options dialog.

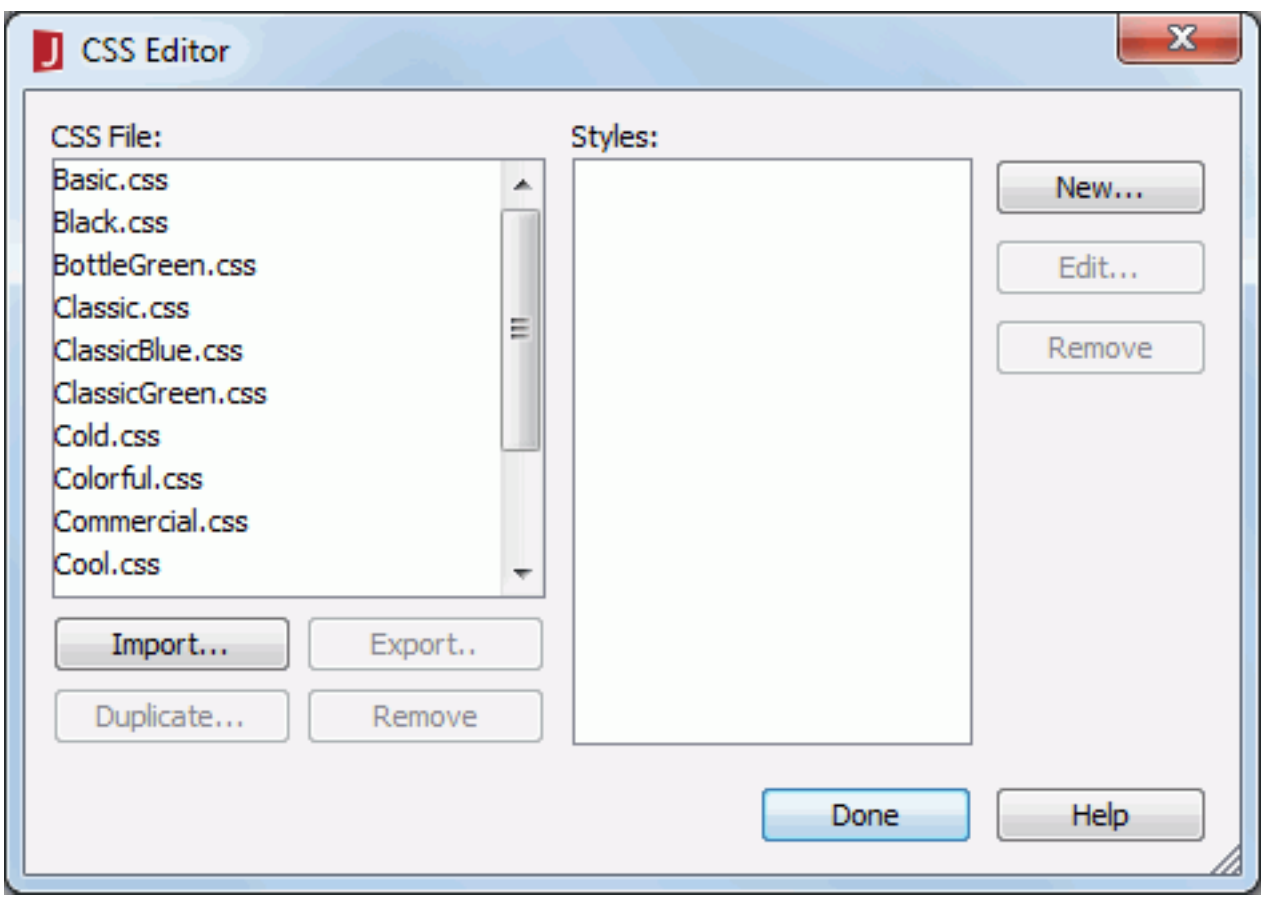
- If you choose to apply a style by setting the Style Group property, later when you preview, export or publish the report, you will be asked to select the style group you want to use or publish. The style specified by the Style Group property will be applied automatically only when you directly run the report on JReport Server.

- When applying CSS styles, you may get warning message showing the CSS properties that are not supported by JReport. If you want to cancel the warning message, uncheck the option "Show warning message when CSS properties are not supported by JReport" in the General category of the Options dialog.

Managing CSS styles

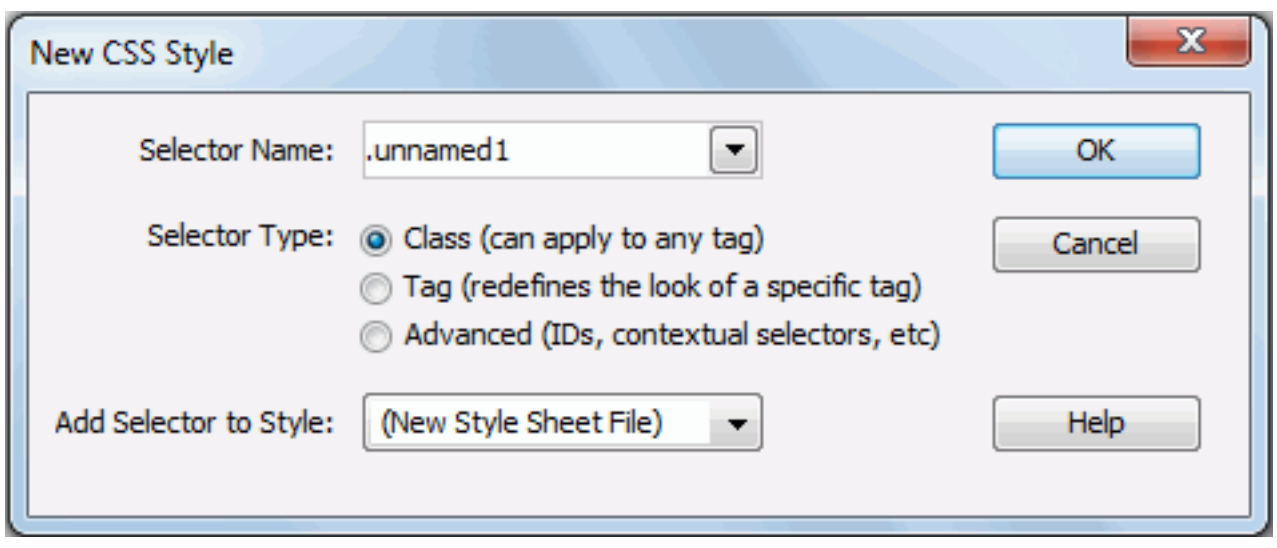
In JReport Designer, you can manage CSS styles at any time in the CSS Editor, including: importing, duplicating, editing, exporting and removing a CSS style.

To access the CSS Editor, click **Edit** > **CSS Editor**, and the CSS Editor dialog appears.



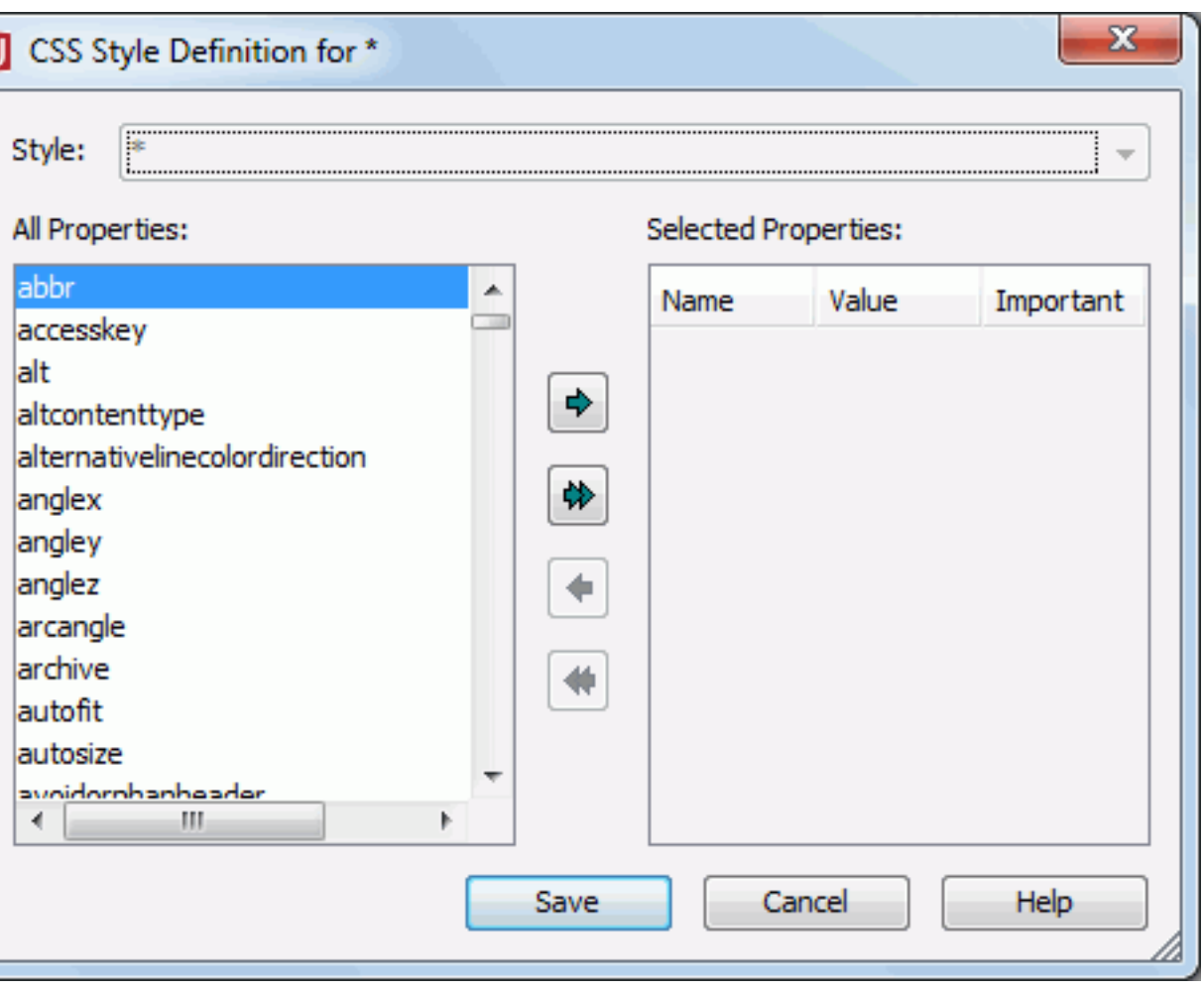
- **To import a CSS style:**
  1. Click the **Import** button in the CSS Editor dialog.
  2. In the Import CSS File dialog, select the CSS style you want to import and click **Open**.

     Note: In the imported CSS styles, only the valid property names, which match with those in JReport Designer, will take effect.
- **To duplicate a CSS style:**
  1. Select the CSS style you want to duplicate from the CSS File box.
  2. Click the **Duplicate** button.
- **To export a CSS style:**
  1. Select the CSS style you want to output from the CSS File box, then click the **Export** button.
  2. In the Save As dialog, specify the directory to locate the CSS style.
  3. Click **Save** to save the CSS style to the specified directory.
- **To remove a CSS style:**
  1. Select the CSS style file you want to remove from the CSS File box.
  2. Click the **Remove** button below the CSS File box to remove the style.
- **To create a style sheet in a CSS style:**
  1. Select the CSS style you want to edit from the CSS File box, then click the **New** button on the right.
  2. In the New CSS Style dialog, define the type of the CSS style as required:



     - To create a custom style that can be applied as a class attribute to a range or block of text, select the **Class** option and then enter a name for the style in the Selector Name combo box.

       Note: Class names must begin with a period and can contain any combination of letters and numbers (for example, .myhead1). If you don't enter a beginning period, JReport Designer automatically enters it for you.
  3. To redefine the default formatting of a specific tag, select the **Tag** option and then select a tag from the Selector Name combo box.
  4. To define the formatting for a particular combination of tags or for all tags that contain a specific ID attribute, select the **Advanced** option and then enter one or more tags in the Selector Name combo box or select one from the box.
- Click **OK** and the CSS Style Definition dialog appears.



- Add the properties you want to contain in the style sheet, specify the values for the properties and eck if the properties are important or not.
- Click **Save** to create the style sheet.
- **To edit the style sheets in a CSS style:**

1. Select the CSS style you want to edit from the CSS File box. All the style sheets in the style file will be displayed in the Styles box.
2. Select the one you want to edit and click the **Edit** button on the right.
3. In the CSS Style Definition dialog, edit properties for the selected style sheet as required.
4. Click **Save** to confirm the editing.

- **To remove a style sheet from a CSS style:**

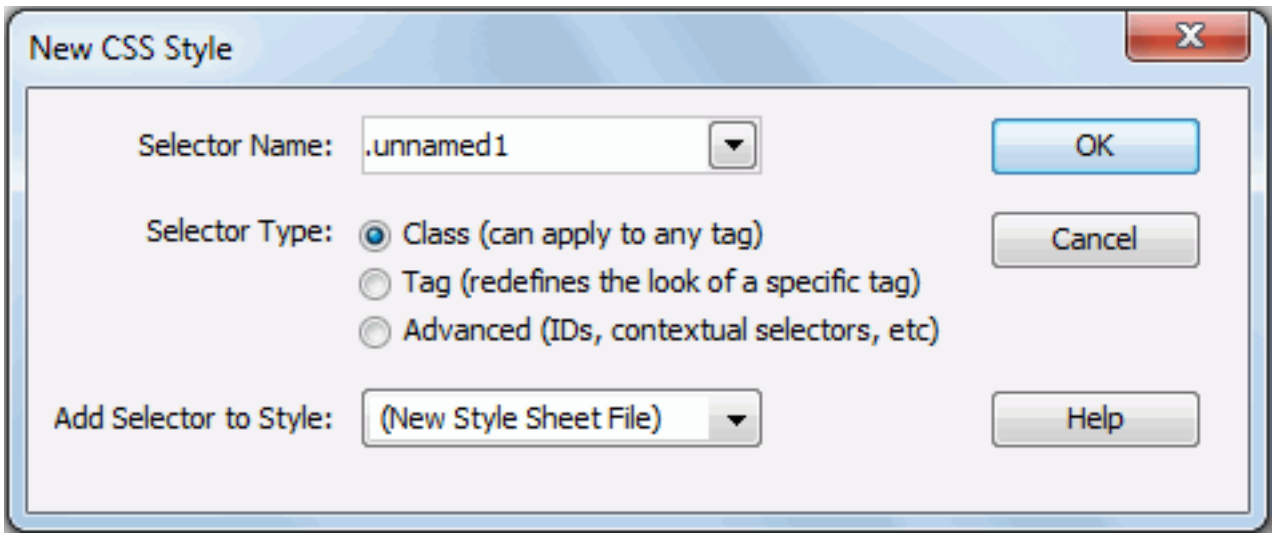
1. Select the CSS style you want to edit from the CSS File box. All the style sheets in the style file will be displayed in the Styles box.
2. Select the style sheet you want to remove and click the **Remove** button on the right.

Creating a CSS style

You can apply a CSS style that is created directly using the properties provided in JReport Designer.

To create a CSS style in JReport Designer:

1. Right-click an object in the design area and then select **Save Style** from the shortcut menu. The New CSS Style dialog appears.



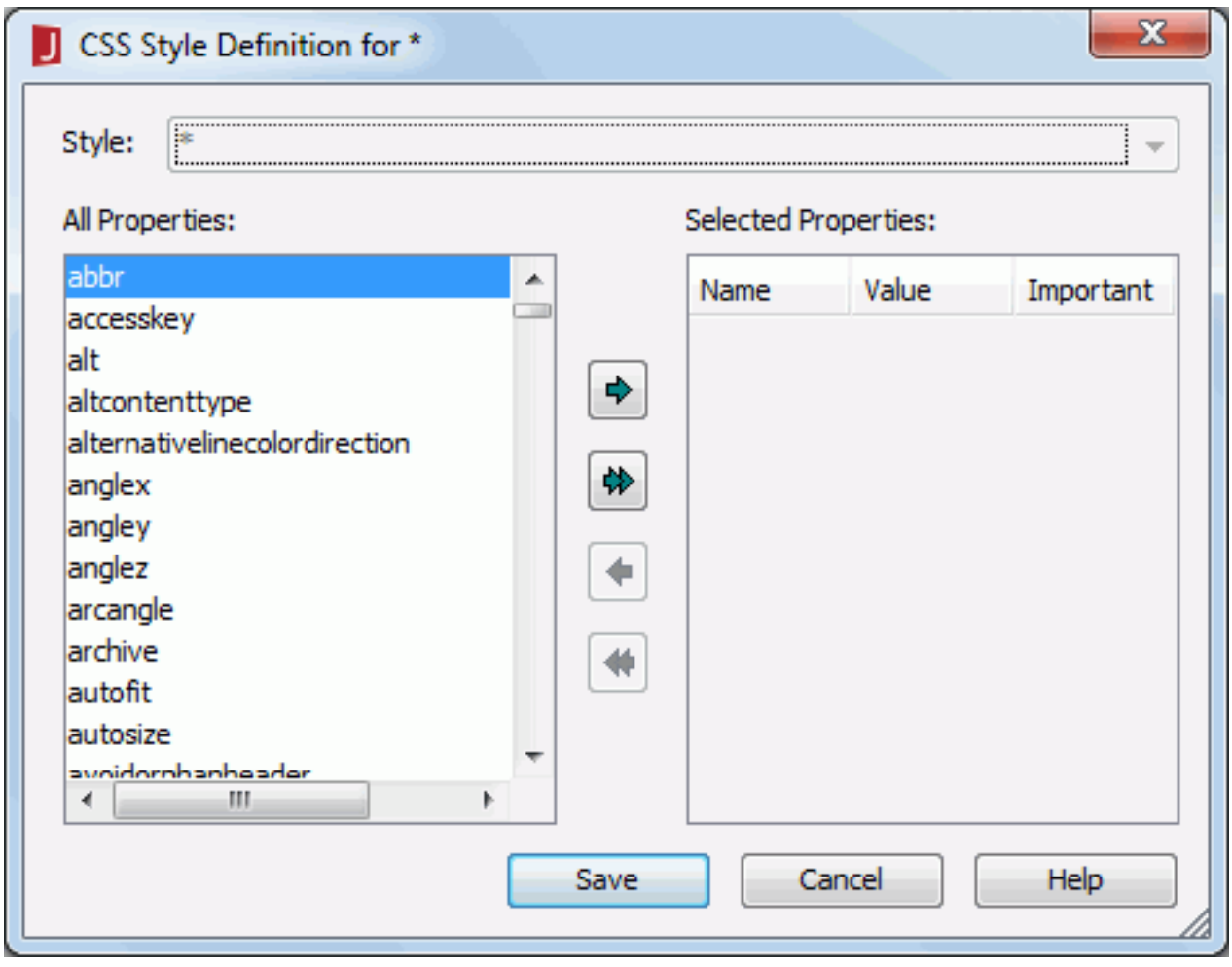
2. Define the type of the CSS style as required:
   - To create a custom style that can be applied as a class attribute to a range or block of text, select the **Class** option and then enter a name for the style in the Selector Name combo box.

     Note: Class names must begin with a period and can contain any combination of letters and numbers (for example, .myhead1). If you don't enter a beginning period, JReport Designer automatically enters it for you.

3. To redefine the default formatting of a specific tag, select the **Tag** option and then select a tag from the Selector Name combo box.

4. To define the formatting for a particular combination of tags or for all tags that contain a specific ID attribute, select the **Advanced** option and then enter one or more tags in the Selector Name combo box or select one from the box.

5. Select the location in which the style will be defined in the Add Selector to Type drop-down list.
   - If you want to create an external style file, select **New Style Sheet File** and click **OK**. Then,
     a. In the Save CSS As dialog, specify the name of the new CSS file.
     b. Click **Save** to save the file and the CSS Style Definition dialog appears.



     c. Define the properties of the CSS file. When done, click **Save** to save the CSS file.
   - If you want to embed the style in an existing CSS file, select any of the existing style and click **OK**. Then,
     a. In the CSS Style Definition dialog, define the properties of the style as required.
     b. When done, click **Save** to save the file.

CSS styles

A CSS style file is like a style group, which contains a set of styles with many predefined properties. In JReport, the CSS style file, which is defined according to CSS 2.1 standard of W3C, is supported. Properties of all of the components in a report can be controlled by a CSS file. You can customize, edit and save a CSS style with visible UI of the CSS Editor in JReport Designer, and can also compile and save a CSS file manually according to the W3C standard. After that, these CSS style files can be imported to apply to a component. By using JReport CSS styles, you can create various visual presentation sets from a single report, and change the visual presentation of your report.

JReport supports some of the W3C CSS 2.1 standard properties at present, which are background-color, border-style/border-top-style/border-bottom-style/border-left-style/border-right-style, color, font-size, height, padding/padding-top/padding-bottom/padding-left/padding-right, text-align, and width.

This section explains how to create CSS styles and manage them in detail.

- Creating a CSS style
- Managing CSS styles

Editing the XSD style files

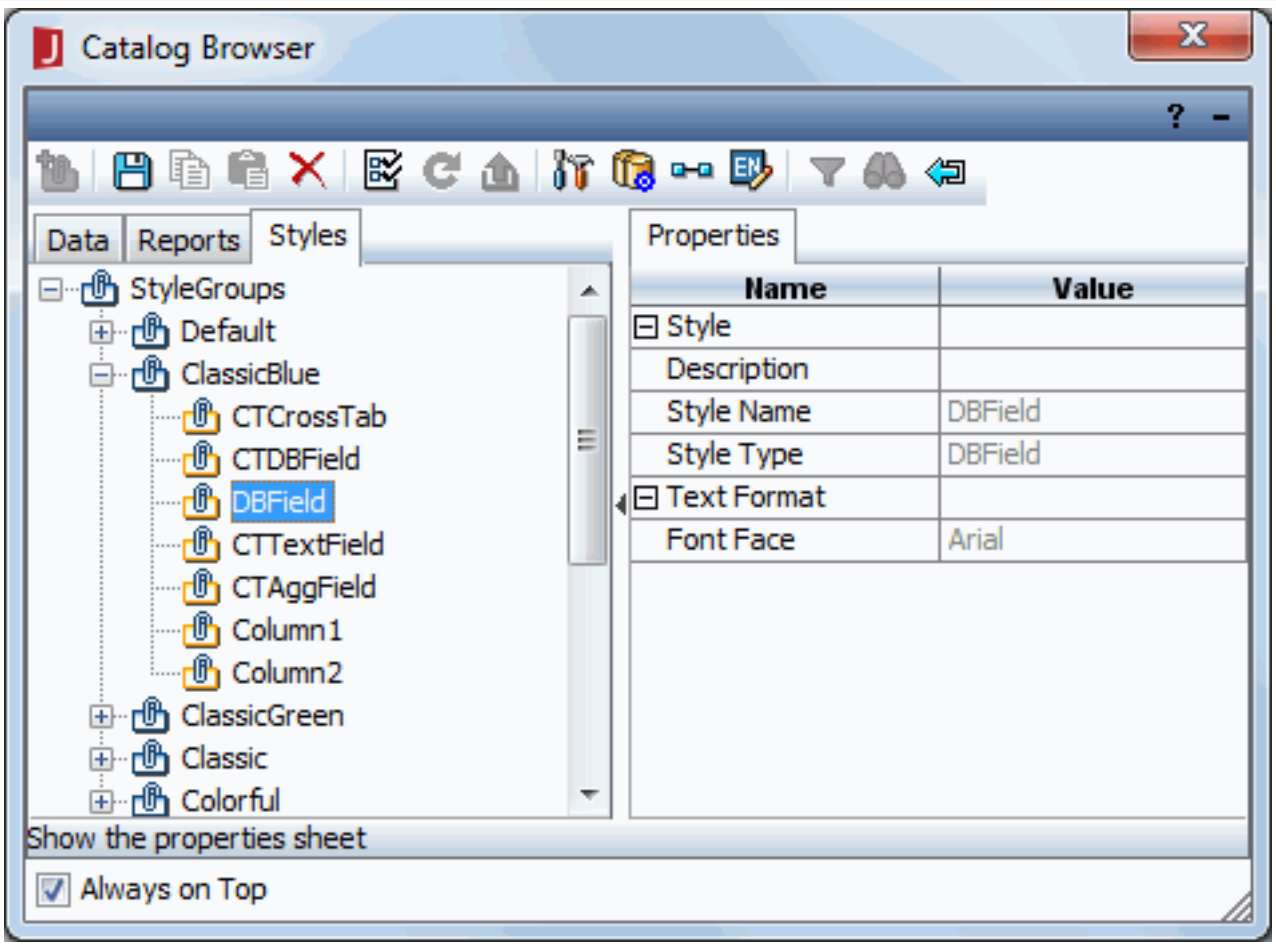
In addition to editing the styles (style groups) in JReport Designer, you can also edit the XSD format styles (style groups) outside JReport Designer.

When you save a catalog, the style information is saved to disk at the same time. The related file will be located as follows:

	File Name	File Description	Location
1.	CatalogName_stl.xml	Style Group Structure in the Catalog	Catalog Folder (The directory where the catalog resides)
2.	StyleGroupName_stl.xsd	Group/Style information	`<install_root>\style`

The CatalogName_stl.xml file keeps the style group structure information. The following shows an example:

The style group structure in Catalog Browser:	The corresponding information in SampleReports_stl.xml:

	`<?xml version="1.0" encoding="UTF-8"?>`
	`<StyleGroup>`
	`<StyleGroupname="Default">`
	`</StyleGroup>`
	`<StyleGroup name="ClassicBlue">`
	`</StyleGroup>`
	`<StyleGroup name="ClassicGreen">`
	`</StyleGroup>`
	`</StyleGroup>`

The StyleGroupName_stl.xsd file keeps the group/style information, which in the above sample will be as follows:

The file content of ClassicBlue_stl.xsd:

```
<?xml version="1.0" encoding="UTF-8"?>
<xs:schema attributeFormDefault="qualified" elementFormDefault="qualified"
version="1.1" xmlns:xs="http://www.w3.org/2001/XMLSchema">
  <xs:notation name="TYPE" public="StyleGroup"></xs:notation>
  <xs:attribute default="" name="Description" type="xs:string"></xs:attribute>
  <xs:attribute default="ClassicBlue" name="GroupName" type="xs:string"></xs:attribute>
  <xs:complexType mixed="true" name="DBField">
    <xs:all maxOccurs="unbounded" minOccurs="0">
      <xs:element default="RightCenter" name="Alignment" type="AlignmentType"></xs:element>
    </xs:all>
    <xs:attribute default="DBField" name="StyleName" type="xs:string"></xs:attribute>
    <xs:attribute default="DBField" name="StyleType" type="xs:string"></xs:attribute>
    <xs:attribute default="" name="Description" type="xs:string"></xs:attribute>
  </xs:complexType>
</xs:schema>
```

As you can see in the above code sample, an XML style group file contains style group information, style information, and the style property entry information. You can edit the attributes of an element.

- To add a style entry, copy and paste the style information section, and then modify the attributes.
- To add a style property entry, copy and paste any element line, and then modify the attributes.
- To delete a style entry or style property entry, select the corresponding section, and delete it.
- To add a new style group, follow the steps below:
  1. Copy and paste an existing XML style group file.
  2. Rename it using the required name (Format: GroupName _stl.xsd).
  3. Change the default attribute in the GroupName element line using the required group name. Make sure that the name is the same as the group name specified in the XML style group file name.
  4. Add a new line for this group in the CatalogName_stl.xml file. For example, if the new group name is GroupA, you would then insert the following line to the file:

     `<StyleGroup name="GroupA"></StyleGroup>`
  5. Modify the XML style group file (GroupName_stl.xsd).
  6. Open the catalog in JReport Designer, and you can see the changes.

Notes:

- The value you type for the default attribute must be consistent with other attributes in the element, and is recognized by JReport Designer.

  For example, if you want to change the Font Face property, you must type a proper font name that JReport supports for the default attribute, such as Arial, Times New Roman, Tahoma, and so on. You cannot type strings that are not recognized by JReport. Otherwise, the element will not take effect at runtime.
- Make sure that the GroupName attribute is consistent with the XML style group file name.

  For example, with GroupA in the XML file, the GroupName attribute is:

  `<xs:attribute name="GroupName" type="xs:string" default="GroupA"></xs:attribute>`

  The XML style group file name should then be GroupA_stl.xsd.

Modifying an XSD style

After an XSD style has been set up, you can further modify it as follows to suit your requirements.

- **Duplicating an XSD style in the group tree**
  1. In the Styles tab of the Catalog Browser, right-click the style group that your want to duplicate.
  2. Select **Copy Style Group** from the shortcut menu.
  3. Right-click the node where you want to duplicate the style group, and select **Paste Style Group**. A new group with all styles within it will be duplicated in the tree.
- **Editing an XSD style**
  1. Right-click the style group and select **Edit Group** from the shortcut menu. The Edit Style Group dialog will then appear.
  2. In the All Styles List, specify the styles that are to be included in the group, and click to add them to the Selected Styles List (or, you can remove some existing styles from the group by clicking).
  3. If you want to create a new style in the group, click the **New Style** button and specify the properties of the style as required.
  4. When done, click **OK** to finish editing.
- **Editing a style in an XSD style**
  1. Right-click the style that you want to edit and select **Edit Style** from the shortcut menu.
  2. In the Save Style dialog, edit the style as required.
  3. When done, click **OK** to apply the changes.
- **Removing an XSD style from a catalog**
  1. Right-click the style group that you want to remove and select **Delete** from the shortcut menu (or press **Delete** on your keyboard).
  2. Click **OK** in the warning message to confirm the removal.

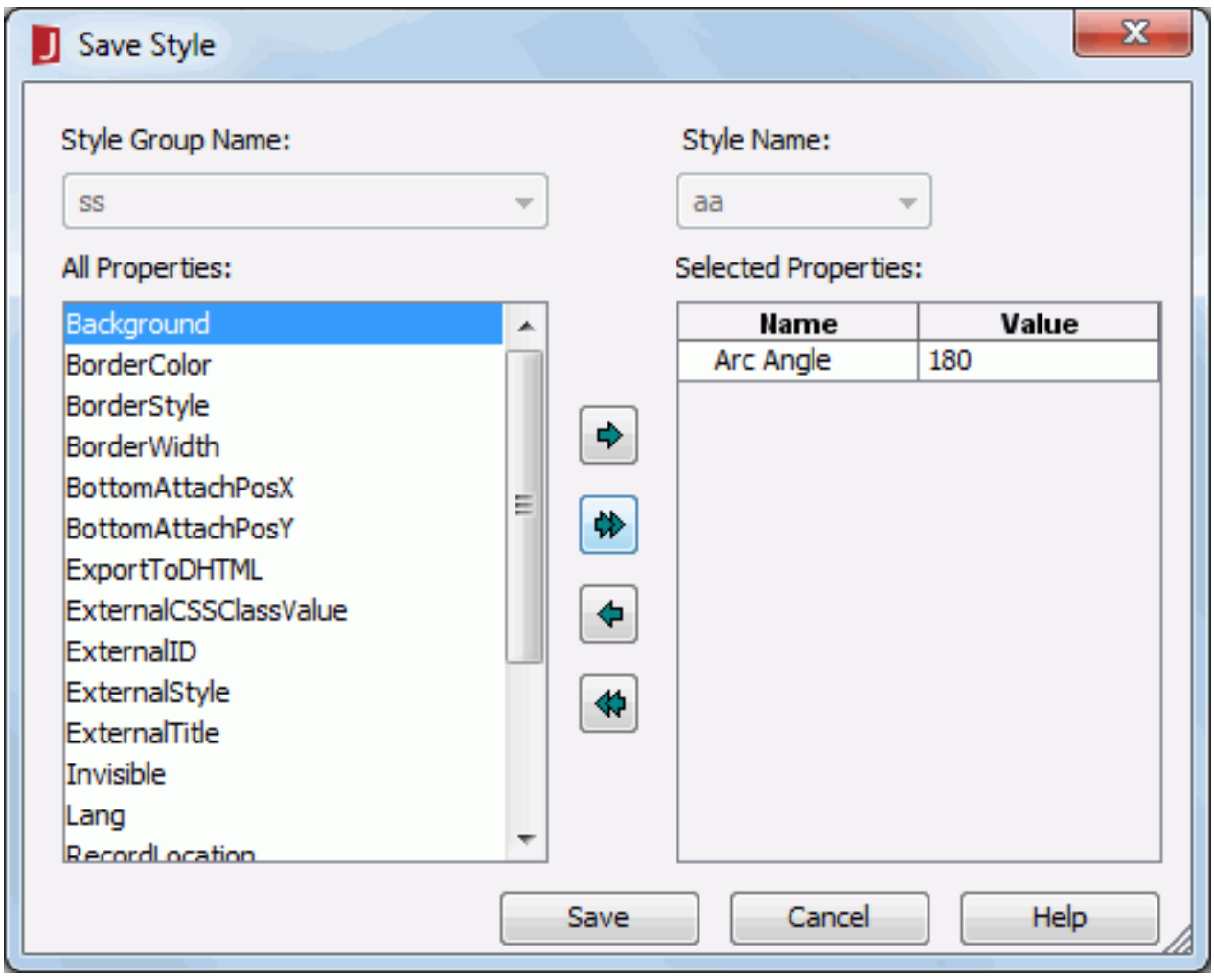
     Note: The default XSD style can neither be deleted nor nested.

Creating an XSD style

To create an XSD style, that is to add a style group in a catalog, follow the steps below:

1. In JReport Designer, open the catalog to which you want to add the style group.
2. In the Styles tab of the Catalog Browser, right-click any of the existing group and select **New Style Group** from the shortcut menu.
3. In the New Style Group dialog, enter the name and description of the group, then click **OK**. A new style group will be added to the tree.
4. Right-click the group and click **New Style**.
5. In the New Style dialog, specify the name, type and description of the new style, then click **OK**.

   The style type indicates the application scope of a style, for example, a style with Label type can only be applied to label objects, and a style with DBField type can only be applied to DBFields.
6. In the Save Style dialog, add the properties to be included in the style and change the property values as required.



   To add a property, select it from the All Properties box and click . To add all the available properties to the style at a time, click .

   To edit the value of a property, click its value cell in the Selected Properties box, then input the new value and press **Enter** on the keyboard to confirm the change.

   To remove a property that has been added, select it from the Selected Properties box and click . To remove all the added properties at a time, click .
7. Click **Save** to add the style to the group.
8. Repeat steps 4 to 7 to add more styles to the group.

Now, an XSD style will have been created. You can then apply the style to your reports.

Notes:

- The root of the style group tree can only hold style groups. The styles must reside in the style groups. You cannot create a style directly in the style group tree root.
- You cannot create style groups in the Default style group.
- The name of each style group must be unique in the style group tree.
- Each style has a style type to indicate its application scope, for example, a style with Label type can only be applied to label objects, and a style with DBField type can only be applied to a DBFields.
- You can create more than one style for a style type, and apply different styles to report objects of the same type. For example, suppose that you have a number of DBFields in your report, such as address fields, totals, and quantity fields. You probably will not want all the fields to look the same. You can create two or more styles, using the same style type - DBField, and then apply them separately to the fields that you want to distinguish.
- You can apply a default style to all the objects that hold the same type.

  In the Default style group, create a new style, making sure that the style name and style type are the same, for example, style name=Label, style type=Label. Then, modify the style properties. Then, when a new object of the specified type is inserted into the report, the default style will be applied to it automatically. Note that you can see the difference only at runtime.

XSD styles

An XSD style, that is a style group in the Catalog Browser, can hold a large amount of predefined object properties, giving you more controls over the appearance and presentation of your report. By using XSD styles, you can create various visual presentation sets from a single report, and change the visual presentation of your report at runtime.

XSD styles are a JReport specific format and not an industry standard so you may want to use CSS styles which are an industry standard.

By simply selecting a style group at runtime, you can change the look and feel of your report in a dramatic way. JReport style groups can be nested, where the parent group is used as a container of other groups. The style group feature makes it easy to save and apply styles without repeatedly designing them each time.

This section explains how to manage XSD styles using the Styles tab of the Catalog Browser as follows:

- Creating an XSD style
- Modifying an XSD style
- Editing the XSD style files

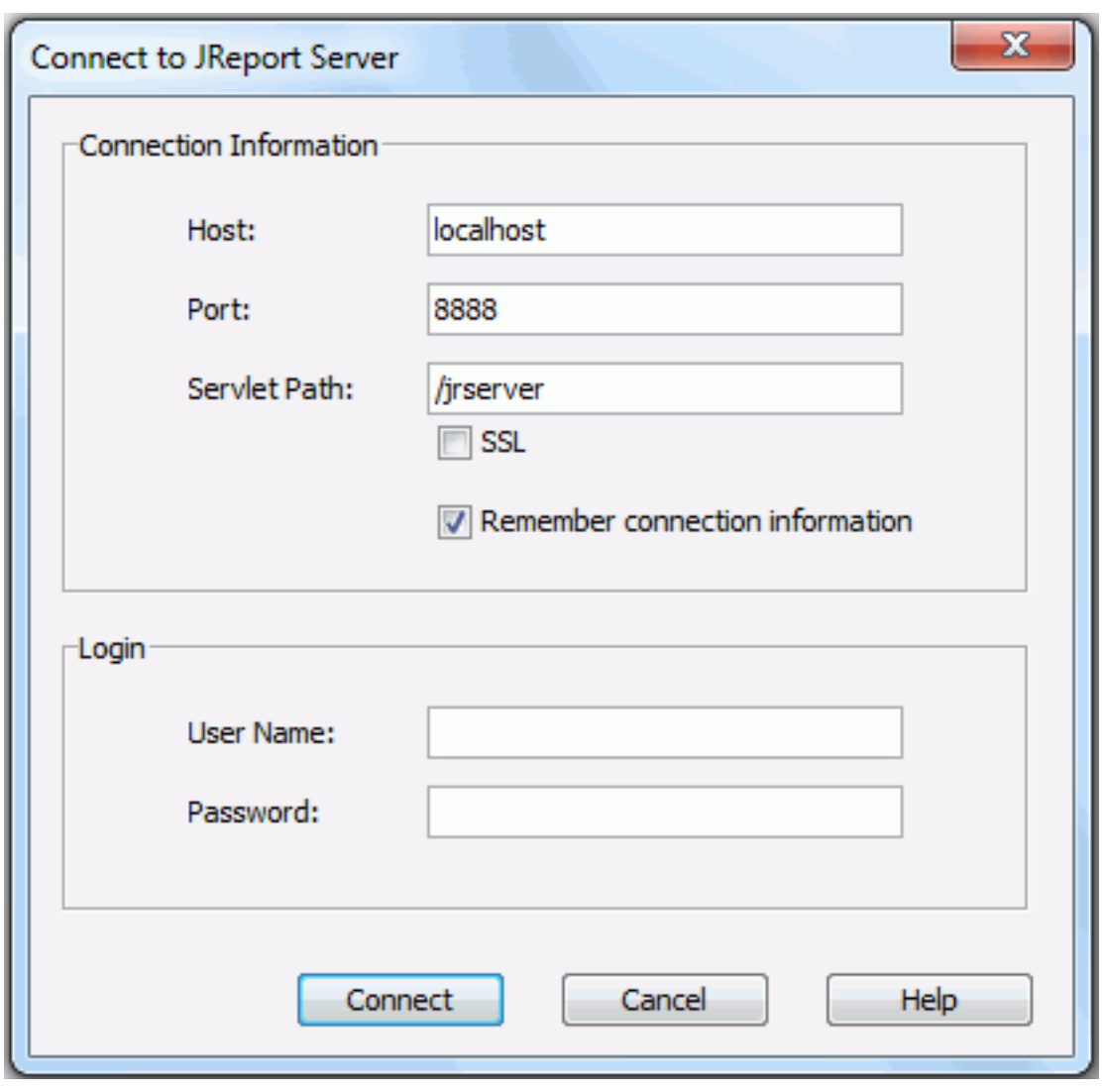
Downloading resources from JReport Server

Resources managed on JReport Server can be downloaded to your local computer and further modified using JReport Designer.

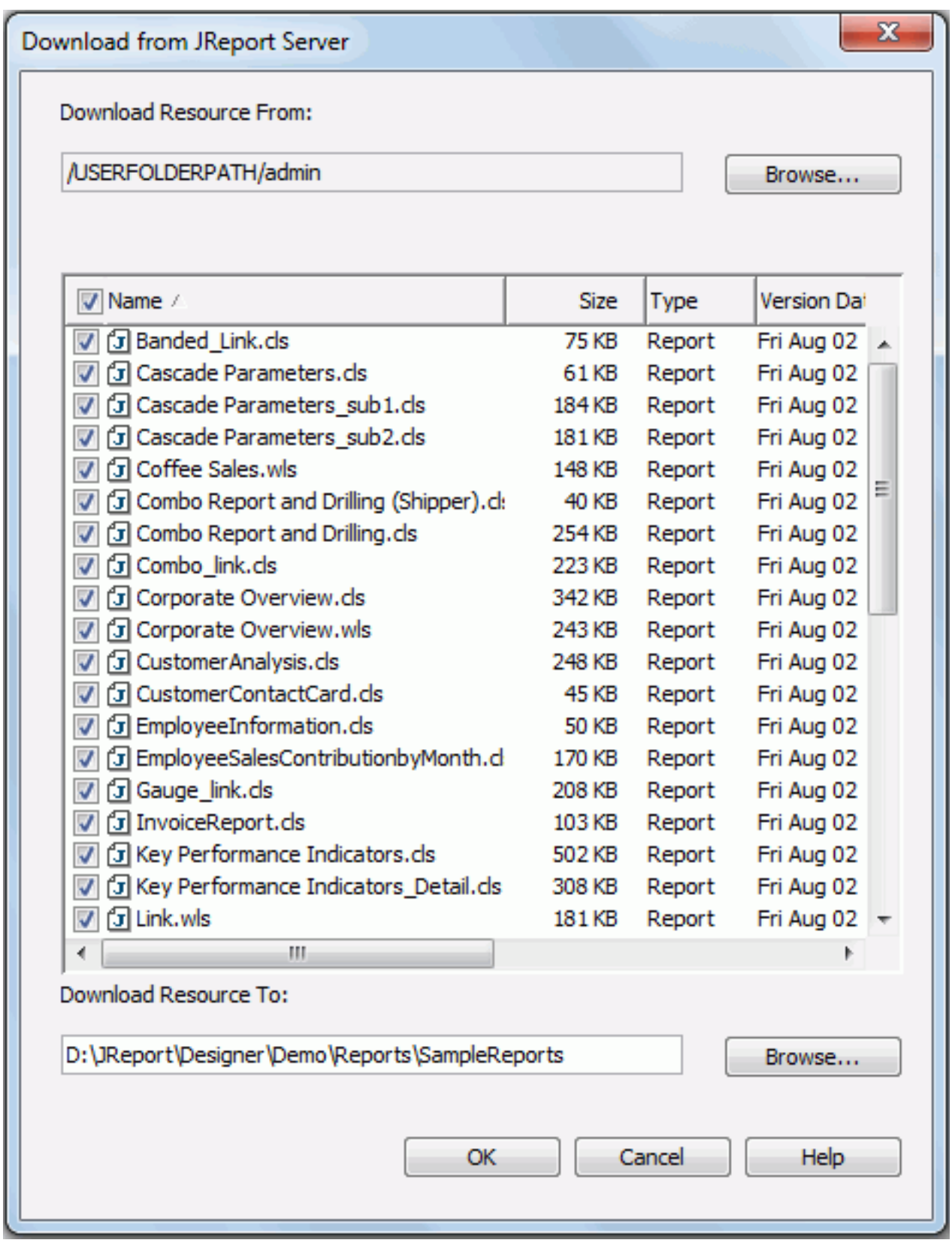
Downloading reports and library components

1. In JReport Designer:
   - Click **File** > **Publish and Download** > **Download from Server** > **Download Report from Server** if you want to download reports.
   - Click **File** > **Publish and Download** > **Download from Server** > **Download Component from Server** if you want to download library components.

   The Connect to JReport Server dialog appears.



2. Type in the host and port of the remote JReport Server in the Host and Port text fields.
3. When using a standalone server, enter **/jrserver** in the Servlet Path text field or the context root of your web application if deployed as a war or ear fie such as /jreport/jrserver.
4. If the server has been integrated with another web server that supports SSL, select the **SSL** (Security Socket Layer) checkbox to create an SSL connection.
5. Check **Remember connection information** to enable JReport Designer to remember the connection information.
6. Provide the user name and password of the server. When the organization feature is enabled and when the user is an organization user, User Name should include the organization name. Use \ to separate the organization name and the user name, for example, org1\user1.
7. Click **Connect**. The Download from JReport Server dialog appears.

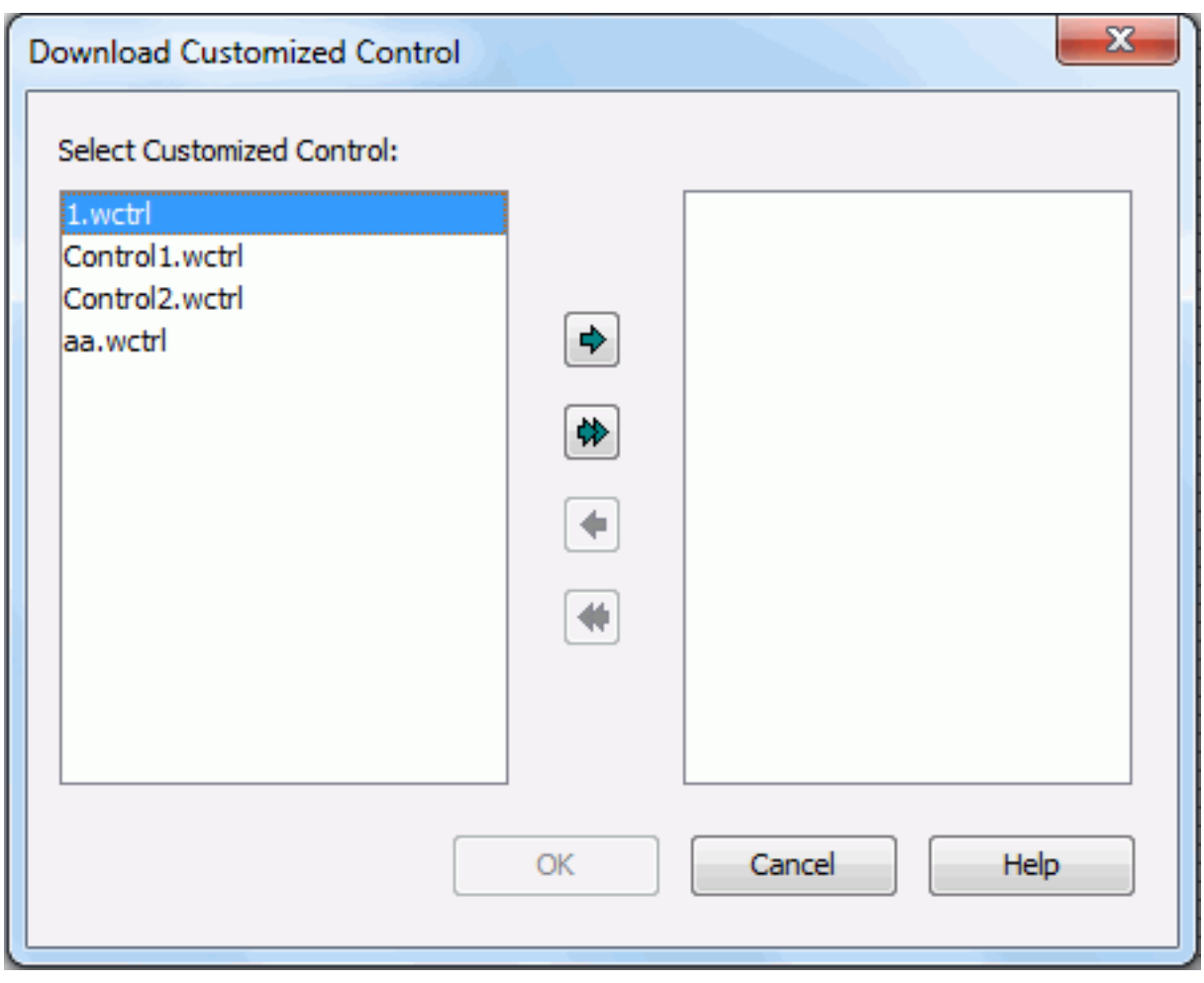


8. In the Download Resource From text field, click the **Browse** button to specify the folder on the server where to download resources. The resources in the specified folder will then be loaded automatically in the resource tree box.
9. In the resource tree box, check the checkboxes in front of the resources that you want to download.
10. Click the **Browse** button next to the Download Resource To text field to specify the path to which resources will be downloaded, or type in the directory in the text field directly.
11. Click the **OK** button to download the resources from JReport Server.

When the resources are successfully downloaded from JReport Server, you can then open them in JReport Designer, and further edit them if required. However, if any report is created in Page Report Studio based on business views, it can only be previewed in JReport Designer.

Downloading customized controls

1. Click **File** > **Publish and Download** > **Download from Server** > **Download Customized Control from JReport Server** on the menu bar.
2. In the Connect to JReport Server dialog, specify the information to connect to a started JReport Server. The Download Customized Control dialog is displayed.



3. Select the customized control files you need, click the right-headed single arrow button to add them to the right box, and then click **OK**.

Notes:

- Resources in the Public Reports folder of JReport Server are protected by permissions. If you specify to download resources from that folder, which resources you can see in the resource tree box and which resources you can download are determined by the user name with which you use to connect to the server. You can only view the resources on which you have the Visible permission in this box and download resources on which you are assigned the Read permission.
- Downloading reports which use business views as data source from JReport Server requires that JReport Designer and JReport Server each has its own Live license.
- When downloading customized control files from server, the downloaded files will overwrite the local ones with the same name directly without warning.

Working with APIs

APIs provide a way to build JReport components and services into other applications, thus integrating JReport with existing platform infrastructure. This chapter is aimed at Java developers wanting to understand how JReport Server works, and how to write Java code that uses the API effectively.

The comprehensive Java doc matrix covering all the classes and methods of the API is available in `<install_root>\help\api`. This chapter of the user guide is an introduction and overview of the APIs so that the vastness of the Java docs will be easier to master.

The technical architecture section reviews specific implementation details about JReport Server. The technical view is different from the product architecture covered in another section of the user guide. This shows the programming elements and processes that exist, rather than the services they provide. The focus is on pointing out Java web application elements and Java servlets and Java programming in the JReport Server product. It introduces the various elements that make up the full JReport API.

The tour of the API section is a high level survey of the JReport Server API. The intent is to show boundaries of interest so that a developer can narrow down the set of APIs to look at when needing to use JReport Server for a particular task.

Beyond that are explanations of use cases, with examples of code.

- Technical architecture
- Tour of the Java API
- Java API for a servlet
- Java API for an application
- JReport Server API
- Page Report Studio API
- JReport Catalog API
- JReport Design API
- JReport Result Viewer Bean
- JReport Catalog Bean
- JReport Exit functions

•

Technical architecture

JReport Server is built with Java and uses standard Java and web technologies. The report server functionality is deployed in classes packaged into jar files that are accessed by class loading based on class path settings. Working web applications built on top of the JReport Server are JSP pages and servlets running in a web server's Servlet Container, using HTTP protocol, generating HTML and Java Script that is passed to the browser.

This section gives a picture of how things fit together and what technologies are used.

This provides a context for later detailed explanations about writing Java code to extend an application to use JReport Server.

Software components

When talking about JReport Server, there are three levels of software components to understand.

- JReport Server - software components

Machine configurations

There are a variety of configurations for setting up a network to use JReport Server. This section will discuss how JReport Server may fit into a variety of network layouts, starting with the simplest plain JReport Server installed on a local PC or server. The discussion continues with how it fits with an existing web application. Then more complex configurations are explained, that scale up and support large web applications.

- JReport Server - out of the box - ready to run
- JReport Server - with existing web application
- JReport Server - remote dedicated server
- JReport Server - cluster

•

JReport Server - software components

The JReport Server product is a combination of several software components.

- At the highest level, there are JSP files that generate interactive web pages that form the JReport Server web application.
- Below that are compiled servlets that supply a group of useful actions controlled by query parameters in the requesting HTTP Request.
- At the bottom level is the Java API. This is a library of Java classes and methods to be used by developers to access JReport Server functionality.

The Java API is at the heart of the JReport Server functionality but it is the higher level components that provide immediate value and use.

JReport Server has production-ready web pages for accessing reports

JReport Server provides a large set of **JSP-based web pages** that are ready for production. These provide standard operations that most applications need.

When these JSP pages are deployed to the website, a visitor can access a web page that lists the set of reports available to them. From there they can schedule and run reports, including entering parameters that control the purpose and scope of the report. They can use this web application to view the results, or direct the reports to a PDF file, an e-mail, or a variety of destinations.

These pages are available immediately. Full report functionality is available through these interactive web interface without writing any Java code or knowing anything about the low level Java API.

These pages are written by Jinfonet engineers using the JReport Server Java API. Since they are JSP pages with visible source, they can be studied for ways of using the Java API or they can be refactored for a local variation of the web page. Using the CSS files and images it is also very easy to private label the application.

Providing these JReport Server web pages in your application is as easy as adding links that lead to the JSP pages. The JSP code can be customized as needed to fit into the design of the application so that they appear seamlessly as part of the whole.

Any web server technology can be used to build an application that includes these JReport Server web pages, including .NET and static HTML.

JReport Server has production-ready servlets for accessing reports

JReport Server also provides a set of **compiled servlets** that perform commonly needed actions related to running and scheduling reports, and viewing results. These services are requested via HTTP requests, with a set of parameters in the URL that define and control the requested service. They are called with parameters that completely define their service so it is possible to use them as helper operations behind an interactive web page that gathers information from a visitor and posts to the servlet URL, with parameters telling the servlet what to do with any resulting output.

Accessing these components of JReport is done by building and sending an HTTP request to the web server with a URL that points to the servlet. This can be done by a desktop program that interacts with a user to build up the needed parameters. A front end for the servlets could be a static HTML web page that provides a form for a user to fill in with the needed parameters before doing a submit to the URL. The web application can dynamically build appropriate links and put them into a menu or other page link. Any type web technology, including .NET can be used to build up an appropriate URL to call up the action of the servlets. Running the servlets does not depend on using Java.

There are examples of how the servlets are called in the sample file TestURL.html which is located in `<instal_root>\help\samples\URLSamples`.

JReport Server has a large set of classes and methods in its Java API

JReport Server provides a library of Java classes that can be used by a Java program. This is the **Java API**, which gives the most complete and versatile way to access JReport functionality. In later architectural illustrations, this may be labeled JReport Server API.

An application developer can extend their application to provide report functionality by writing custom code to call the JReport Server Java API to perform any operations needed. This is the key aspect of JReport Server that makes it appealing for application developers who need reporting functionality resident within the same web server.

Looking at the Javadocs for the Java API is intimidating. There are hundreds of classes and thousands of methods.

A later section gives a high-level tour of the Java API, organized by functional area. This will introduce the Java API so it is easier to understand. It will explain how the classes are organized to provide specific types of functionality.

A developer only needs to understand the working set of classes and methods that relate to a particular application need. A developer does not need to know about every class or method documented in the Java API Javadocs.

•

JReport Server - out of the box

JReport Server is a ready-to-run product that appears at first as a web application that lets users run reports and view results.

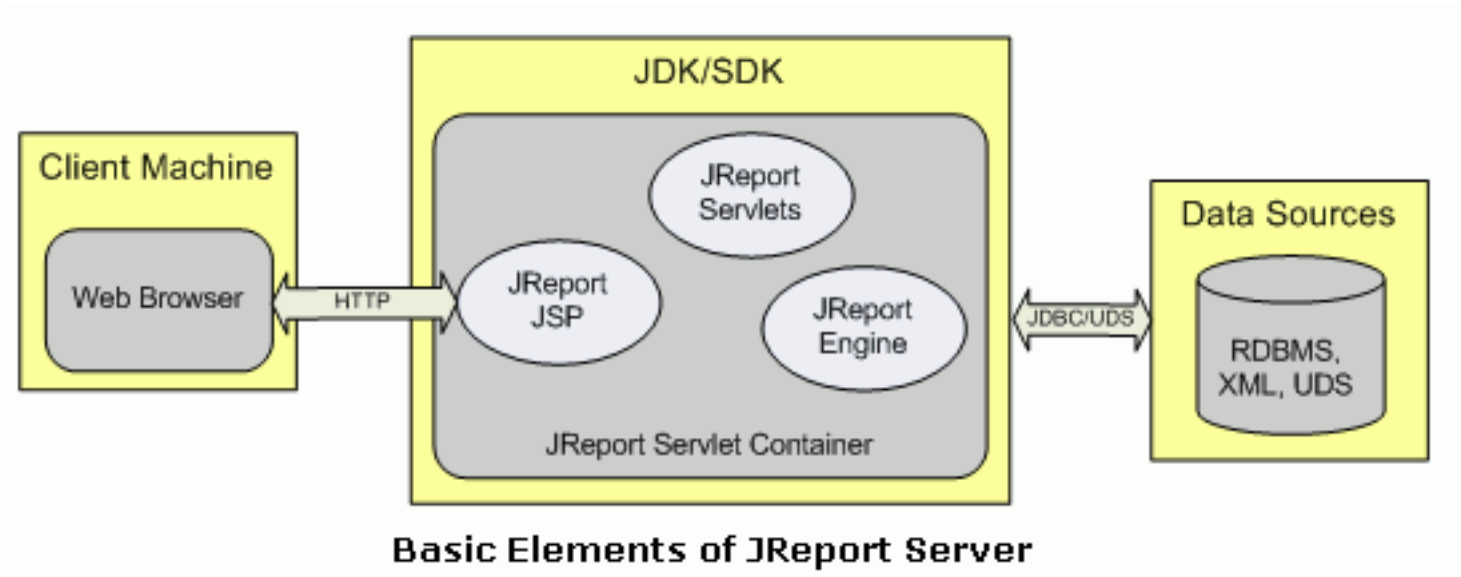
Install JReport Server and start it running by using the JRServer command or Windows Start menu. It is a web server ready for requests. It comes with a initial set of registered users and a set of sample reports that are in place to be run and viewed.

Start a browser and enter the URL to the entry page. Assuming the install is onto the local PC, this is the URL to use: `http://localhost:8888`.

Login with user/password of admin/admin. After login, the JReport Server web application is available.

What's running and how does it work?

JReport Server - web application



Basic Elements of JReport Server

The JReport Server's web application is based on JSP files in several folders under the server's public_html folder and a few servlets compiled into classes. These JSP files and servlets create the HTML passed back to the browser, which gives users the web pages that access to the JReport services. A visitor to the site can see a list of reports available, select one to schedule or run, and view or download the results. This is a standard web application giving access to a set of sample report resources that come with the product.

Viewing report results may involve use of Java Script running within an HTML page. This javascript code is supplied by a library of files in the public_html\javascript folder.

The web application also includes a folder of JSP pages (public_html\admin) that are dedicated to performing administrative duties for managing and configuring the server. These can only be run by an administrative user.

The servlet code that generates web pages makes calls to classes in the JReport Engine which does the real work. These classes are the Java API.

The files that make up JReport Server's web application code are JSP files, compiled servlets, and compiled classes deployed in library jar files.

- <install_root>\public_html\jinfonet*
- <install_root>\public_html\dhtmljsp*
- <install_root>\public_html\admin*
- <install_root>\public_html\javascript*
- <install_root>\lib\JREntServer.jar
- <install_root>\lib\JREngine.jar
- <install_root>\lib\JREESServlets.jar

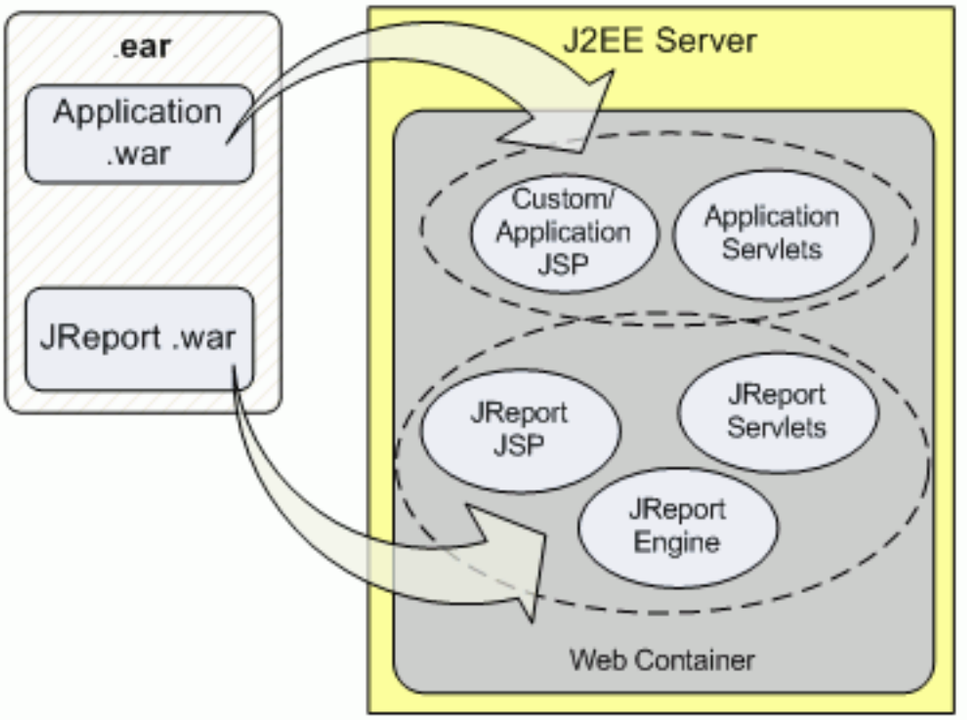
Login and servlet session

All web applications need to protect web page access from unauthorized use. JReport Server's web application includes a check at the start of every JSP page and servlet to verify that the requesting user is an authenticated JReport user before running.

User identity for a web session is managed across HTTP requests by data in the servlet session that is passed into the Servlet container with the HTTP Request.

•

JReport Server - with existing web application



Integrate JReport Server with Existing Application

JReport Server can be integrated into an existing web application. Put the application's JSP files and servlets together with the JReport JSP files and servlets, along with the JReport Server library files.

After JReport Server is installed along with the existing application, the application can be extended by writing new code that uses the JReport Server API library.

It is when JReport Server is used as part of a larger web application that it's other identity begins to show. JReport Server API comes into view. JReport Server contains is a large set of classes with methods that can be used by any Java program. This set of classes and methods is the JReport Java API.

Integrating JSP pages

An existing application can integrate with JReport Server web pages in a simple fashion by adding links from the existing application to the JReport Server JSP pages.

This provides the JReport Server web pages with the look and feel that is native to JReport Server.

Because the JReport Server web application is implemented as JSP pages, these can be edited to change the look and feel while not changing the functionality.

An existing application can be extended to access JReport Server functionality by adding Java code to access the Report Server classes and methods that are defined in the Java API. An existing application can have complete control over the look and feel of web pages while providing access to running and scheduling reports and viewing results.

Many of the JSP pages in JReport Server could be copied to the customer's application, modified, ignored or removed and have web interactions with users provided by an extended existing application, using Java calls to request report functionality.

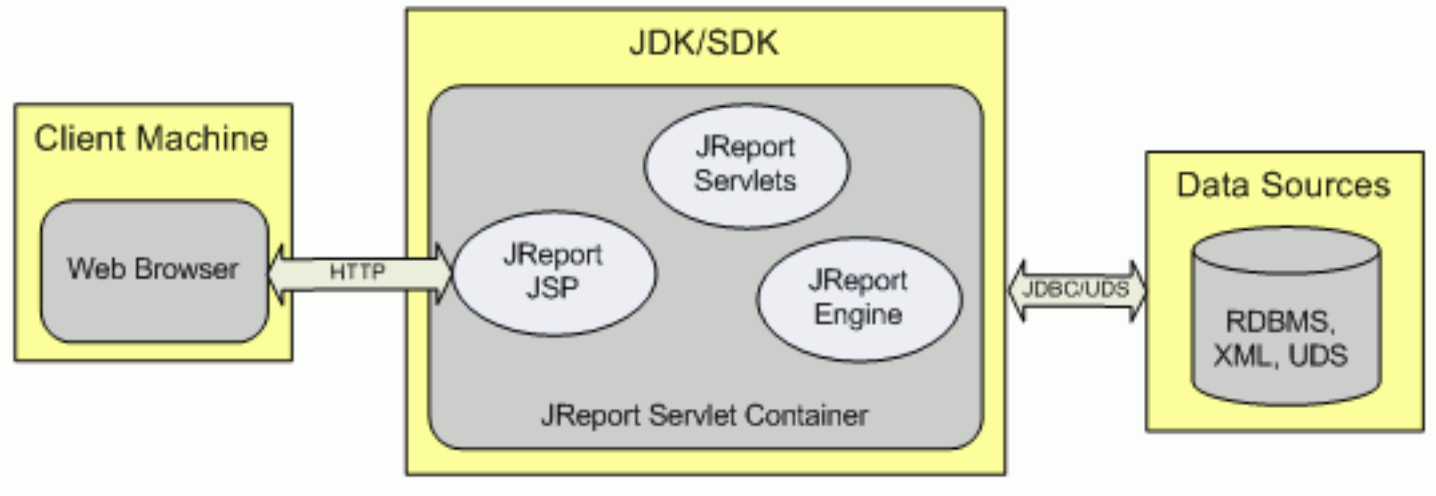
•

JReport Server - remote dedicated server

JReport Server can run in standalone mode directly as it is installed out of the box or can run as a Windows Service or as a WAR file running in any standard servlet container such as provided by Tomcat, JBoss, WebSphere, WebLogic, and so on. The customers application and users then access the JReport Server across the net either via RMI to servlets from Java application code or JSP page, or via HTTP requests with a URL that directly call JReport's servlets and JSP pages.

In a servlet container

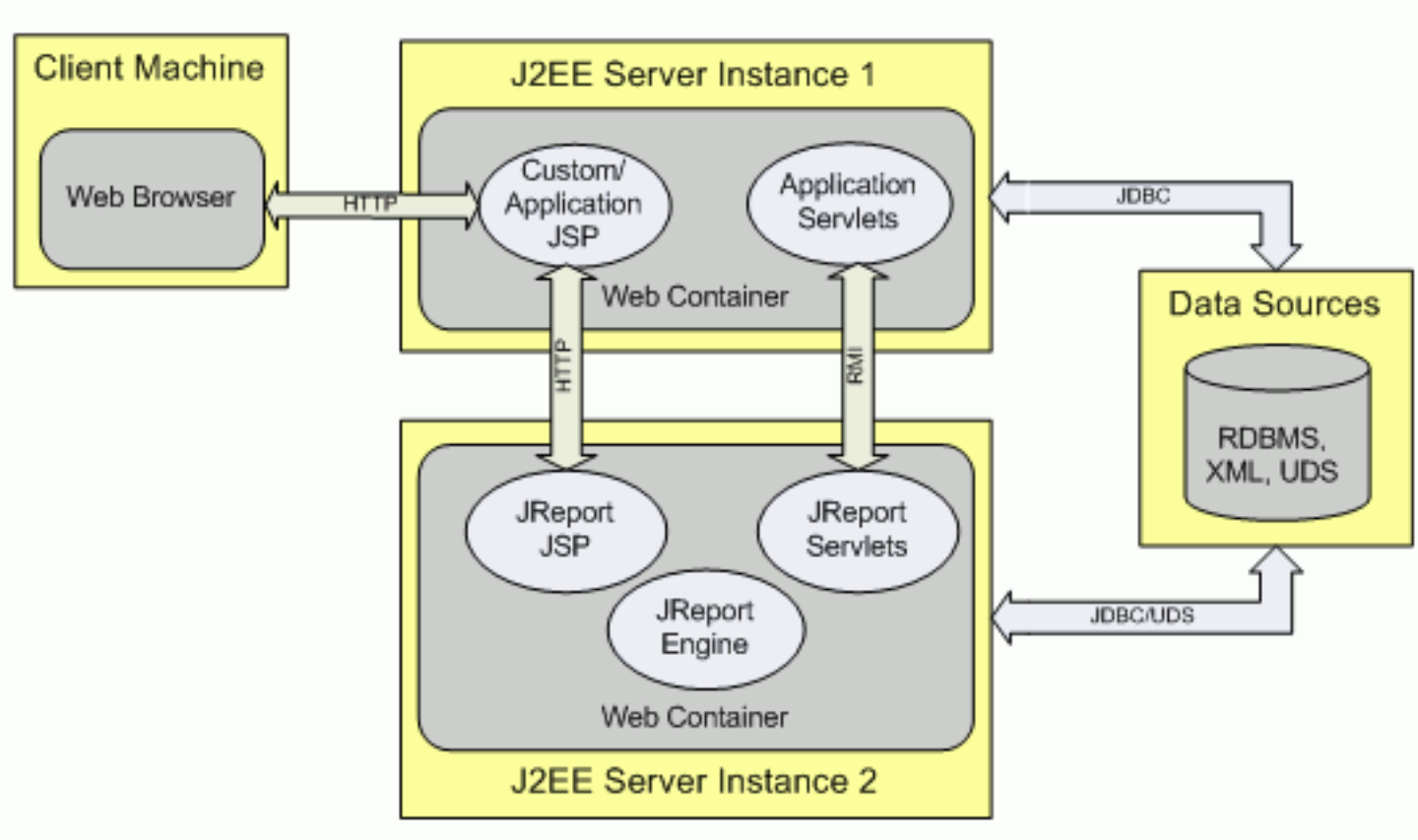
This is a common configuration for accessing JReport functionality for small applications. JReport Server is installed in a servlet container, ready to handle requests. The client machine can be simply a browser or be a full web application. It requests JReport functionality by using URLs to JReport servlets and JReport JSP pages running in the JReport Servlet Container. This is the enjoinment when you run JReport Server from the JRServer command line or build a jreport.war and and deploy it to an application server to run in it's own container.



JReport Server in a Servlet Container

As a remote dedicated server

When JReport is embedded into an existing application it is usually run in a separate JVM or a separate server machine. JReport runs as a back end server handling requests coming from the web application server's front end. JReport Servers' JSP pages are moved to the front end machine into the same servlet container or JVM as the application code. The front end web application's Java code runs with version of the Java API that are stubs providing seamless access to the back end methods using RMI (Remote Method Invocation) to call the real Java methods in the JReport Server in the other JVM.



JReport Server Accessed by RMI

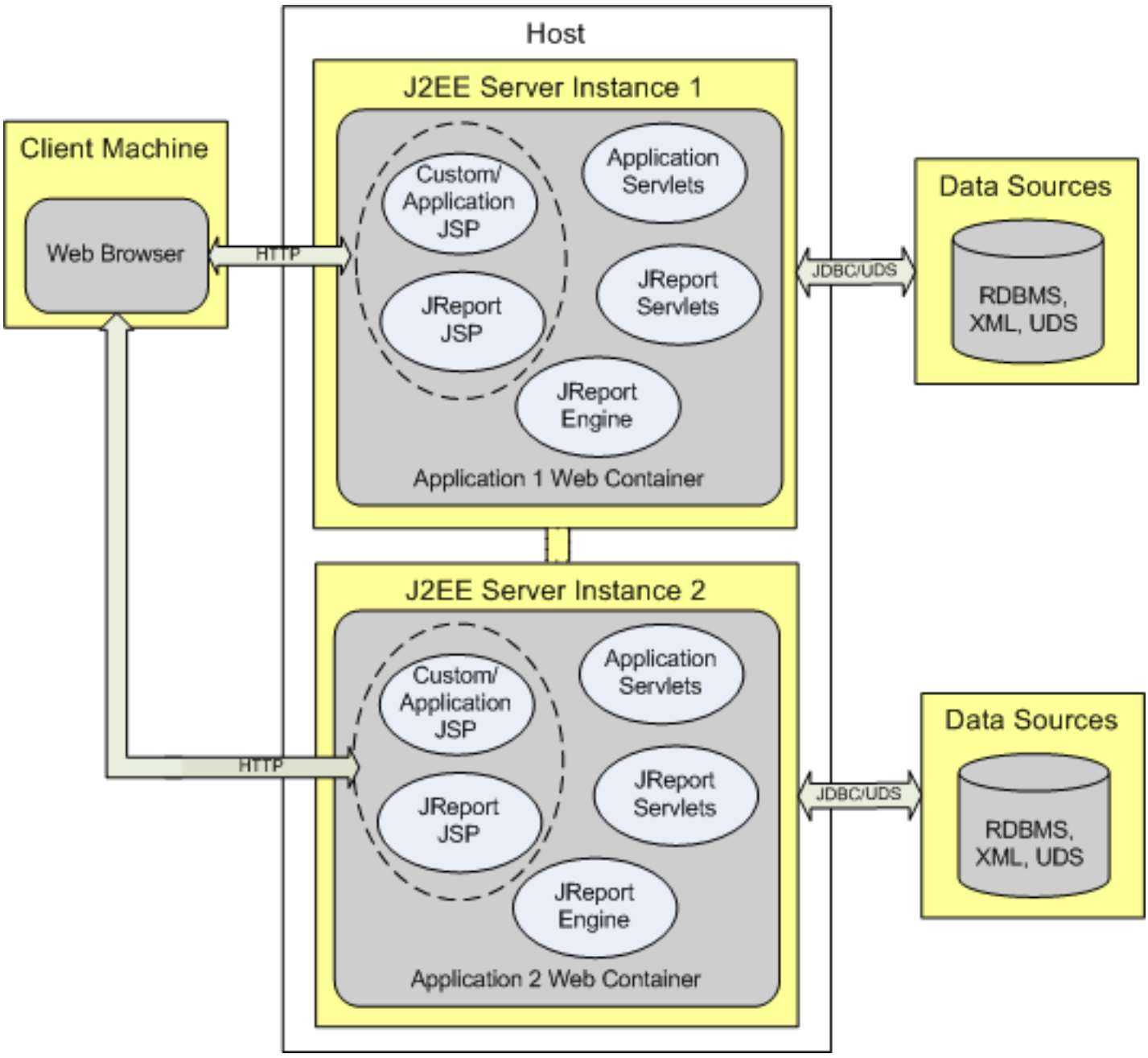
•

JReport Server - cluster

JReport can also run as a cluster using either a cluster of remote JReport standalone instances or as a cluster of JReport WAR files running inside a group of independent application server instances or as an application server cluster such as JBoss's cluster technology. When JReport is in a clustered environment, there is a built in dispatcher that has several available algorithms to distribute the runtime load. A Java interface is defined that allows application developers to write their own load distribution algorithm and register that with the JReport Cluster service.

JReport Server Cluster

The following diagram shows one potential configuration where JReport and the application are running in web containers. The JReport instances can be clustered so no matter where a resource is published it is visible on both systems and even if one system goes down for maintenance, the other system can handle the requests and new instances can be quickly started to handle larger loads. The client machines can go to either server so the user does not need to be aware of which JReport Server is handling the request.



Load Balancer

JReport provides several built in algorithms for load balancing and assigning tasks to servers. In most cases, one of the built in methods described in load balancing mechanism is adequate; however, JReport provides an interface called LoadBalancer that allows you to customize assigning tasks to specific server instances. For example, you may want to assign tasks to specific servers based on the user id yet still go to other servers if the specified server is overloaded or down.

You can view the Javadoc at jet.server.api.cluster.LoadBalancer and jet.server.api.rmi. RemoteDispatcher. See the examples in `<install_root>\help\samples\APICluster`.

Tour of the Java API

Using JReport Server with an existing application

Integrating JReport Server with an application:

- Use the web application component of JReport Server JSP pages. Allow users to browse in the application, select data, and generate reports using the JReport Server web application. Use JReport programs to offer special features based on calls the Java API. The web application components offer a full-featured, ready-to-use, web-based reporting system with scheduling, notification, and security and permissions. These can be used with an application built with any technology including .NET and HTML web pages.
- Use the JReport Server services to make direct requests to URL. Although not technically an API, they serve the same purpose. Action requests are the method calls, with query parameters similar to method parameters. The set of server actions is limited to running, scheduling and viewing results. The URL interface is simpler than the Java API and can be used from any web server or URL reading and permissions. These can be used with an application built with any technology including .NET and HTML web pages.
- Use the Java API classes and methods. They are called directly from a Java program to extend functionality beyond the web application components. The Java API provides the most flexibility and control. Use the Java API classes and methods and JReport Server JSP pages and JReport Server web application that describes how to use it.

Every functionality in JReport Server is available through classes in the Java API including creating and modifying catalogs and reports, providing security such as single sign on (SSO), authentication and permissions, scheduling, notifications, and running and viewing reports.

The Java API classes are documented in the JReport Server Reference Guide. The Reference Guide describes the method and parameters. It is this library of classes and methods that is explained below. Some of the classes are used by the web application components.

This tour section categorizes the Java API classes by functionality making it easier to find specific methods to use for particular functions.

Java Application Programming Interface (API)

The Java API classes are in the com.jrepublic.jrepublic.server package.

The major functional areas in the library are categorized below into classes and methods.

Some classes are used in the web application components and have a general description. Below that will be a list of common methods, and other information as needed.

Initializing and connecting to JReport Server

Class: com.jrepublic.jrepublic.server.JRepublic

The JRepublic class is the main class for initializing a server and for calls to execute an instance of the server, getting the server object and a handle to the server for use in later API calls.

Class: com.jrepublic.jrepublic.server.JRepublicServer - This class provides the programs running within JReport Server and access to the server's resources.

Additionally, the two most commonly used methods of the Java API are in this class. These two methods are used to get a handle to the server and to execute a request. Most other methods are used in conjunction with the JRepublicServer class.

- **Method:** JRepublic.getServer() - initialize and start the singleton JReport Server running on the machine and return a handle to the server.
- **Method:** JRepublicServer.getRequest() - connect to the running JReport Server and return a handle to the server to use in future API calls.

Security

Every web application must manage access control, identity of a visitor, permissions for a visitor, and authentication of a visitor. JReport Server provides a full set of API to do this.

JReport Server provides two methods for doing this authentication and access control. Both JRepublicServer and query parameters to pass in log in credentials. The security mechanism includes a framework for single sign on (SSO) authentication and a set of permissions for controlling access to the server. The security mechanism is described in the JReport Server Administration Guide.

- **Control access to web pages**
  - **Class:** com.jrepublic.jrepublic.server.JRepublicServer - This class provides a variety of utilities for programs running within a Java Servlet container.
  - **Method:** JRepublicServer.isAuthorized() - This method only returns when a validated JReport User is authorized to access the current web page. Otherwise it redirects to the login page. The URL for the validation and login is saved based on the credentials in the current HTTP request. If no logged in user is found, the browser is redirected to the login page. If a valid user is found, the browser is redirected to the saved URL. Cookies are used to keep the browser logged in. The credentials are saved back in the same JSP page in the next HTTP Request.
  - **Method:** JRepublicServer.getCookieUsers() - This class provides methods that help to manage the cookies used to keep the browser logged in. Cookies are used.
  - **Method:** JRepublicServerWebManager.checkLogin() - redirects if a validated JReport User is logged into the server. If not a user is logged in, it redirects to the login page. The credentials are saved in the cookie and in the current HTTP request. It does not automatically send a response to the browser. Use JRepublicServerWebManager.redirectToLogin().
  - **Method:** JRepublicServerWebManager.redirectToLogin() - redirects to the URL in the current HTTP request (HTTP 401) to the browser, which initiates a HTTP Authentication process to obtain a valid credentials for the JRepublicServer.
  - **Sample code:** RPWebscriptsAppLogoutSignedOn.jsp - entry point page for a group of JSP pages that provide a web interface to JRepublic. This page logs the current user out of the JRepublic server.
- **Network security:** Using the checkLogin() system for connecting access to the web pages to the JRepublic server, the web pages can be accessed through a secure HTTP connection. The JRepublic server can be configured to require a secure connection. Then the web pages can be accessed only over SSL(s for web pages using https://URL) instead of http. Secure Socket Layer (SSL) can easily be added to the JRepublic server configuration.
- **Permissions:** JRepublicServerAuthorizer.getAuthorizer() - This class provides access to the JReport User ID (user name) for the one who is currently logged into the web session.
  - **Method:** JRepublicServerAuthorizer.getAuthorizer().getUserID() - This method is used to the browser redirecting the user to the application login page, and return a true to let JReport Server know that the application has taken over the authentication process.
  - **Sample code:** RPWebscriptsAppLogoutSignedOn.jsp - entry point page for a group of JSP pages implementation of the interface. Read comments in the source for detailed explanations on the implementation of the interface.
- **Authentication:** Use the login page provided or write your own. Authentication is based on a login page (JReport Server, or when using a server to register it with JReport Server by stating an API call) and a cookie to keep the browser logged in. The login page can be a JSP page or a web page. Cookies are used to keep the browser logged in. This base authentication system can be extended to use application developers can write and deploy within JReport Server that co-operates with the JRepublicServer methods and authentication system.
  - This base interface defines a protocol for the application to tell JReport Server that a user is already logged in. This is used when the application has authenticated the user. The JRepublic server will then redirect to the application login page. The application should then redirect back to the JRepublic server. The application should then call the JRepublic server to get a valid session. The application should then redirect back to the application to provide the single place where a user logs into the web application. This also provides a single place for writing the HTTP Authentication Protocol.
  - Single Sign On is turned on when the implementation of this base interface is compiled and deployed. Read the source code for detailed explanations.
  - **Method:** JRepublicAuthenticationSingleSignOnInterface.getUserID() - Return the JReport User ID (user name) for the one who is currently logged into the web session.
  - **Method:** JRepublicAuthenticationSingleSignOnInterface.isAuthenticated() - Tell a response to send to the browser redirecting the user to the application login page, and return a true to let JReport Server know that the application has taken over the authentication process.
  - **Sample code:** RPWebscriptsAppLogoutSignedOn.jsp - entry point page for a group of JSP pages implementation of the interface. Read comments in the source for detailed explanations on the implementation of the interface.
  - **Implementation:** Use the sample code provided or write your own. Authentication is based on a login page (JReport Server, or when using a server to register it with JReport Server by stating an API call) and a cookie to keep the browser logged in. The login page can be a JSP page or a web page. Cookies are used to keep the browser logged in. This base authentication system can be extended to use application developers can write and deploy within JReport Server that co-operates with the JRepublicAuthenticationSingleSignOnInterface.
  - **Class:** com.jrepublic.jrepublic.server.JRepublicAuthenticationSingleSignOnImplementation - This class provides a framework to allow Single Sign On, to read this experience. When Single Sign On is turned on, this class will be called to provide the authentication. Read the source code for detailed explanations on the implementation of the interface.
  - **Sample code:** RPWebscriptsSingleSignOnConnectionsessionNotificationAuthorizer.jsp - example of a server that uses Single Sign On. Read the source code for detailed explanations.
  - **Single Sign On:** JReport Server will use Single Sign On when the implementation of the interface is compiled and deployed is registered with it.
  - **Sample code:** RPWebscriptsAppLogoutSignedOn.jsp
  - **Registration:** Use the sample code that starts up your application or JReport Server. The property jrs.loginPrimaryAuthenticator should be set to the name of the implementation class. For the web application components, the property jrs.loginPrimaryAuthenticator should be set to the name of the implementation class. Read the source code for detailed explanations.
  - **Register with Java API call:**
  - **Class:** com.jrepublic.jrepublic.server.JRepublicServer - This class provides methods that manage the user session associated with a request.
  - **Method:** JRepublicServer.getSessionManager().setAuthenticator() - turn on Single Sign On by setting the name of the implementation class. Read the source code. When a null value is used as the class instead to set, this turns off Single Sign On.
  - **Register with JReport Server configuration:**
  - **Class:** com.jrepublic.jrepublic.server.JRepublicServer.java - example of a server that sets Single Sign On.
  - **Where to turn on Single Sign On**
  - Single Sign On authentication is turned on for the JVM where the JSP pages are running. If the application and JReport JSP pages are running on a web server with JReport Server installed on a machine, Single Sign On will be turned on for the JVM where the web server is running. If the application and JReport JSP pages are running on a web server and JReport Server is running on a different machine for the web pages, write access to the back end JReport Server will be over RMI, which does not support Single Sign On. Use HTTP access instead.
  - **Single Sign On Connections:** Single Sign On connections are made from the JRepublic server to the web pages that demonstrate how Single Sign On works. Read comments in the JSP page to learn more about the connections.
  - **Authentication Interface:** The interface defines a class that application developers can write that replaces the built in user validation system of JReport. Even when using Single Sign On, the system will call this class to validate a user. The application can use this class to connect to a third party authentication system and check if the user is valid. A custom implementation can use the DBMS data in the application's data to validate the user. Another option rather than to authenticate the user is to use the system to do authentication based on the JReport User or Active Directory user.
  - **Interface:** per server.api.custom.security.AuthenticationProvider
  - **Implementation:**
  - **Class:** com.jrepublic.jrepublic.server.DefaultAuthenticationProvider
  - **Authorization** - This built in package provides methods for determining authorization rights to application and JReport Server resources. Read the source code for detailed explanations. Use the API to add/ edit/ delete/ manage the mapping users, groups, and roles. Permissions to resources can be placed on the mapping of users, groups, and roles. The JRepublic Server Authorizer can be extended to use application developers write and more flexibility as individual users change jobs while roles are stable for long periods of time.
  - **Class:** per server.api.Authorizer
  - **Method:** isAuthorized()
  - **Method:** addUser()
  - **Method:** removeUser()
  - **Method:** getUsers()
  - **Class:** com.jrepublic.jrepublic.server.JRepublicAuthorizer
  - **Custom Authorization** - This base interface defines a class that application developers can implement to add custom authorization to the JRepublic Server. Read the source code for detailed explanations. Use the API to add/ edit/ delete/ manage the mapping of users to roles. Permissions to resources can be placed on the mapping of users, groups, and roles. Permissions to resources can be placed against the target resource. Methods in this base interface will be called to determine if the user has access to the target resource. The JRepublic Server Authorizer can be extended. You can implement your own method to extend the base interface.
  - **Interface:** per server.api.custom.security.AuthorizerProvider
  - **Implementation:**
  - **Class:** per server.api.admin.SecurityAdminServer
  - **Method:** addRole()
  - **Method:** removeRole()
  - **Method:** addUser()
  - **Method:** removeUser()
  - **Sample code:** JRepublicSecurityAdmin.jsp, JRepublicSecurityAdminSecurityUserEdit.jsp

Event Framework - Task Listener

The JRepublic Server provides a framework model that allows the application to provide methods that are called at various points in the processing of requests in JReport Server. These include calls to methods before and after a report is run, before and after a report is saved, before and after a report is deleted, before and after a report is scheduled, before and after a report is run by a scheduled task. Using this framework you can keep detailed logs of information about the report in addition to the logging information that JRepublic Server provides.

Using these methods, the application developers can implement and deploy to provide extra functionality and access to extra resources.

- **Interface:** per server.api.TaskListener
- **Method:** beforeRun
- **Method:** afterRun
- **Method:** validateParameter
- **Sample code:** RPTestLoadserverTestTaskListener.java

Engine API

JReport Engine allows you to run reports directly in your application without connecting to a JReport Server. Reports can be run from a Java Servlet, Java Application, or Java Applet. The JRepublic Server API provides a library that is also available to the application to run reports directly in the application.

- **Class:** per server.api.engine.ReportEngine
- **Method:** Report Engine constructor - Creates a report server been
- **Method:** setRepositories - There are a number of required set methods to set repository, catalog, report name, and report path.
- **Method:** exportPDF - There are a number of export methods and parameters to export a report to a PDF file, HTML, Excel, Text, and Comma Separated Value.

NLS

This Java interface defines an implementation you can build to provide your own methods to do date formatting, number formatting, and translation. The methods provided by JReport Server work out of the box but can be replaced.

- **Interface:** per.nls.NlsBundleInterface
  - NlsBundleImpl
  - **Sample code:** RPNLSWebNlsBundleInfo.jsp

Design API

JRepublic Report Design class and the subclass WorkbookDesigner allow a developer to write programs that create and modify reports with API calls that are fully compatible with reports you create using the JRepublic Report Designer.

- **Class:** per.server.api.engine.ReportDesigner
- **Class:** per.server.api.engine.WorkbookDesigner
- **Method:** Report Designer constructor - Creates a report server been
- **Method:** setRepositories - There are a number of required set methods to set repository, catalog, report name, and report path.
- **Method:** exportPDF - There are a number of export methods and parameters to export a report to a PDF file, HTML, Excel, Text, and Comma Separated Value.
- **Method:** setCellValue - Set a value in the report set. It is the only container object that can be inserted into banded parts.
- **Method:** insertBandedObject - Creates a Banded Object structure that includes BandedObject, BandedObjectGroup, BandedObjectGroupHeader, BandedObjectGroupFooter, BandedObjectDetailPart.
- **Method:** insertBandedGroup - Inserts a group into a BandedObject or into a BandedObjectGroup.
- **Method:** insertChart - Inserts a chart into the parent part
- **Method:** insertCrossTab - Inserts a cross tab into the parent part
- **Method:** insertTable - Inserts a table into the parent part
- **Method:** getNamedItem - Retrieve a named item in the hierarchies listed above.
- **Method:** insert - Inserts labels, DBfields, parameters, formulas, summaries, and special fields into the parent part.
- **Method:** set - Set the attributes of the inserted objects such as text, font, foreground, format, and alignment.
- **Sample code:** RPWebscriptsDesignerTestSimple.jsps, TestDesignerSimple.jsps, TestDesignerCrossTabWorkbook.jsps, TestDesignerTestTableWorkbook.jsps, TestDesignerDetailPart.jsps, TestDesignerBandedGroup.jsps, TestWorkbookDesign.jsps

Catalog API

The Catalog class and the subclass WebUserCatalog have methods that allow application programs to create and modify catalogs and reports. The Catalog class is used by the JRepublic Server Administrator to create and modify catalogs and reports. When the web application is used, it provides a set of methods so multiple users updating the catalog will not collide with each other. You should want to use this if you are building a web application that allows users to edit reports. The JRepublic Server Administrator is a web application that allows users to create and modify catalogs and reports. You can also use the Catalog class and the subclass WebUserCatalog to write your own web application to edit reports. Report Cubes and Business Cubes are referred to as Business Logic in the Catalog API.

- **Class:** per.server.catalog.Catalog
- **Class:** Designer getCatalogs() - Gets a catalog API instance for a given catalog from a Repository.
- **Method:** mkdir() - objects in catalog
  - getObjectNames() - objects in catalog
- **Sample code:** RPCatalogTestCatalogAPI.jsps, TestCatalogBC.jsps

User Defined Formula - UDF

User defined formula allow developers to create their own formulas to use in JReport just like the built in formulas. The JRepublic Server provides the ability to create some useful functions to read and write files on the server, such as for logging usage.

- **Class:** per.formula.UserDefinedFormula
- **Sample code:** RPUDFPrivateUserJarUserDefinedFormulas.jsps

User Defined Data Source - UDS

User defined data sources allow developers to build a java jar RecordSet with custom code that JReport Server will use exactly the same as how JReport Server uses a JDBC RecordSet.

- **Class:** per.datasource.JRHierarchicalDatasource
- **Class:** per.datasource.JRHierarchicalDatasource
- **Class:** per.server.UDSUserJarUDSCommonExample.java, UriorSampleRecordSet.java

Resource Management

The ResourceManager class provides methods to control all aspects of deploying resources to the JRepublic Server. Resources are files that are already deployed. Resources include folders, catalogs, reports, and other resources.

- **Class:** per.server.api.ResourceManager
  - **Immediate:** false
  - **Method:** createFolder()
  - **Method:** deleteFolder()
  - **Method:** addResourceToFolder()
  - **Immediate:** true
  - **Method:** getAuthorizedUsers().checkPrivilege()
  - **Method:** setPrivilege()
  - **Immediate:** Delete
  - **Method:** deleteResource()
  - **Method:** removeResource()
  - **Immediate:** true
  - **Method:** getReportsForPath()
  - **Method:** getAuthorizedUsers()
- **Sample code:** RPServerAdminPanelDeployReport.jsps

Trigger Manager API

The Trigger Manager allows the application code to fire a trigger that will start running of all the scheduled tasks waiting on the trigger.

- **Class:** per.server.api.TriggerManager
- **Method:** subOnDemandTask()
  - **Method:** fire()
- **Sample code:** RPServerOpenJRTrigger.jsps

Run Report - RpServerAPI

This class is used by programs that run JReport Server. Although it is documented as a Java interface, it is not intended to be replaced by a custom implementation built by developers. This section will talk about the API. The API provides methods to connect to the JRepublic Server, get a list of available reports, run a report, get a list of scheduled and run and schedule reports. Report results can also be saved as resources using these methods.

- **Class:** per.server.api.RemoteObjectServer
- **Method:** getRemoteObjectServer() - connect to the JRepublic Server to see that an instance of RemoteObjectServer is needed by an application program before any other API methods may be called.
- **Class:** per.server.api.RpServer
- **Method:** runReport() - run on demand report
  - **Method:** runReportSchedule() - run a new report using a schedule to run the report.
- **Sample code:** RPServerAPIRemoteRunAndSaveReport.jsps

E-mail

This class is used by programs that want to interface to an SMTP server to export results to e-mail as attachments or as embedded reports in the e-mail body.

- **Sample code:** RPServerAPIRemoteSendMail.jsps

Remote File Service

This class is used by client programs that run a report on a remote JReport Server but then want the results saved on the client machine. The API allows for the to be available to open locally on the client machine.

- **Class:** per.server.api.RemoteRpClientFileServer
  - **Method:** getRemoteRpClientFileServer()
  - **Method:** copyFromRemote()
  - **Sample code:** RPServerAPIRemoteRpClientFileServer.jsps

Remote Method Invocation - RMI

This class is used by programs that need to request reports to run on a remote server and to copy the results back to the client machine. Although it is documented as a Java interface, it is not intended to be replaced by programs written by developers. Although it is documented as a Java interface, it is easy to write an application that tries to connect remotely before starting a local RpServer.

- **Class:** per.server.api.rmi.RemoteRpServerTaskRes
  - **Method:** getRemoteRpServer()
  - **Method:** runTask()
  - **Method:** getResourceManager()
  - **Method:** getRemoteRpServer()
- **Sample code:** RPWebscriptsServerRemoteRPOpenPublishJRE.jsps, RPWebscriptsServerRemoteRPOpenPublishJRE.jsps, RemoteRPOpenPublishJRETimeout.jsps

Schedule Report using RpServer

This class allows programs to add, modify and delete schedules based on time or triggers.

- **Class:** per.server.api.RpServer
  - subnOnDemandTask() - run scheduled report task
- **Sample code:** RPServerAPIRemoteRpServerRunReport.jsps and RPWebscriptsServerRemoteRPOpenPublishJRE.jsps

Remote Object Server

The RemoteObjectServer class allows for a remote system application to determine the load balance options, and connection options to start up the run on specified servers.

- **Class:** per.server.api.rmi.RemoteObjectServer
  - RemoteObjectServerLoadBalancer
- **Sample code:** RPCluster/DemoLoadBalancer.jsps

Page Report Studio API

This class allows applications to directly control the options and features provided to the user when using Page Report Studio to view reports.

- per.web.direct.PageReportStudio

Server Configuration

This class allows applications to configure the server similar to the JReport Administration Console. For example, the application can add the active JDP pages under jrepublic/jrepublic.

- per.server.api.admin.cfg

Server Monitor

JReport Server Monitor is a client application that can be downloaded for free from the JRepublic Website. This class provides the ability to monitor activity on a single server or on an entire group of servers in a cluster. The API can be used to start and stop servers as well as configure properties on them remotely.

- per.server.api.monitor

Server Profiling

This is the class used by JReport Server to implement Server Monitor. It allows your application to monitor the activity of resources use as Server Monitor using JMX technology.

- per.server.api.profiling

Server Cached Resustable Data (CRD)

This class provides the ability to create a cache that the application can schedule just the cached result data to be updated. The application can then use the cache to use it. Often it is for datasets that many reports might need such as month end sales reports.

- per.server.api.crd

JSP and URL Samples

JSP_DATAHUNDREDREPORT

This JSP page is a sample that builds URLs to run a report in Page Report Studio. The user can choose the report, the output format, the number of records, the output type (inline or attachment), and the output compression. The user can also add a search function to search the current report for specific content. To use this page, the URL should be set to the JRepublic Server URL and port. The URL should be http://localhost:8080/demo/tour.jsp. The default port is 8080, and you can use the demo using the URL http://localhost:8080/demo/tour.jsp.

Java API for a servlet

The Java servlet API provides a set of Object-Oriented classes and interfaces which can be used to write a CGI-like. This section contains a brief overview of the Java servlet API. Java servlets are Java programs that run on a Web server and interact with client requests and Web server responses. Java servlets are similar to CGI programs in that they can be used to generate dynamic content in response to client requests. Unlike CGI programs, however, Java servlets are more efficient than CGI programs because they are designed to handle Web processing tasks. Examples below will show the standard patterns for reading method calls to handle processing.

Java servlet API

Java servlets are Java programs that run on a Web server and interact with client requests and server responses.

Java servlets technology provides two basic extensions needed for servlets. Processing the requesting URL and forming the response. Java servlets are similar to CGI programs in that they can be used to generate dynamic content in response to client requests. Unlike CGI programs, however, Java servlets are more efficient than CGI programs because they are designed to handle Web processing tasks.

```
import javax.servlet.*;
import javax.servlet.http.*;
import java.io.*;

// the service method defined in get-post-put-delete service
// may be called from the run-time
// the service method calls the doGet method
String cvsRes = "<meta http-equiv=Content-Type content=text/html;
                 charset=iso-8859-1><HTML><HEAD><TITLE>Test Servlet
                 Output</TITLE></HEAD><BODY>";
// the service method calls the doGet method to handle the request
// the service method calls the doPost method to handle the request
// the end is a total call to cvsRes()
// the doGet method is called by the service method
// the doGet method calls the doPost method to handle deprecated values
// the doPost method
// the doPost method is called by the service
// the doPost method is identify current user
// the doPost method calls the getCurrentUser() current user and checkUserExist()
// the doPost method calls the doPost method to handle the form's action and checks to
doPost(HttpServletRequest request,
HttpServletResponse response)
{
// the doPost method calls the doPost method to handle the form's action
// the doPost method calls the doPost method to handle the form's action
// the doPost method calls the doPost method to handle the form's action
```

•

Java API for an application

JReport provides two ways for the application to call JReport directly without using JSP pages and URLs.

- JReport Server API: JReport provides a library of methods to directly start JReport Server as part of the customers application running in the same JVM as the application.
- JReport Remote Server API: JReport provides methods to connect to an already running instance of JReport Server either running on the local machine in a different JVM or running on an entirely different server. Other than the first call to connect either using RMI or directly in the current JVM, the rest of the API is identical so it is easy to write an application that can work either remotely or locally the same as the JSP pages that JReport provides.

How to write client code using the Server API

If you want the entire JReport Server to run in the context of your application in the same JVM you can use the Server API. The Server when started this way starts a number of threads and runs exactly the same as when the server is started as a standalone server. The server just starts with the application and stops with the application. All access to the server such as RMI access from other applications is still allowed.

```
      //set report home
  System.getProperties().put("reporthome", "C:\\JReport\\Server");

      // Creates instance of RptServer ( this does nothing if JReport Server is already running in this
JVM)
  HttpUtil.initEnv(System.getProperties());

      // Get a handle to the server instance
  RptServer server = HttpUtil.getHttpRptServer();
```

How to write client code using the Remote Server API

It is exactly the same code as above except you get a remote handle to the RPTServer.

```
      //set report home for logging and location of rmi.auth file for security
  System.getProperties().put("reporthome", "C:\\JReport\\Server");
  System.getProperties().put("jrs.rmi.auth_file", "C:\\JReport\\Server\\bin\\rmi.auth");

      //search the running server using default host and port
  String host = "localhost";
  String port = "1129";

  RptServer server = RemoteReportServerToolkit.getRemoteWrappedRptServer(host, port);
```

JReport Server API

JReport Server API is a set of Java programming interfaces that run reports, explore report resources, and provide access control for report servers. It is mainly used for writing servlets, JSPs and Java applications with report server features.

The following topics are covered in this chapter:

- Installing the Server API
- Creating and getting instances of the ReportEngine
- Creating and getting instances of RptServer or HttpRptServer
- Using JSP page with a dedicated machine
- Using RMI in JReport Server
- Loading user data source classes at runtime
- Applying a user defined CSS to the HTML result file
- Specifying paths for the result files when using On-Demand API
- Setting dynamic parameter values when running/scheduling report via API
- Setting values of multi-value parameters via API
- Specifying report running properties in the session
- NLS API
- Dashboard Listener API
- Dynamic connection API
- Dynamic security API
- Information Bus API
- API demos
- RMI demos

Installing the Server API

When you install JReport Server, JReport Server API is also installed at the same time. After installation, you will have the following files:

- In `<install_root>\lib`:
  - JRESServlets.jar (local calls)
  - JREngine.jar (local calls)
  - JRSRMI.jar (remote RMI calls)
  - servlet.jar
  - hsqldb.jar
  - log4j-1.2.8.jar
  - sac.jar
  - commons-codec-1.2.jar
  - commons-logging-1.1.jar
  - quartz-all-1.6.5.jar
- In `<install_root>\derby\lib`:
  - derby.jar
  - derbyclient.jar
  - derbynet.jar
  - derbytools.jar

The Server API classes are stored in the archive file JRESServlets.jar.

When you install JReport Designer, the libraries for JReport Server can also be found in `<designer_install_root>\server\lib` since JReport Designer includes a full JReport Server for previewing reports.

If you are calling the Server API directly in the same JVM as JReport Server, you need to set the classpath environment variable to the local versions of the classes. Append the following jar files to your class path that compile and run applications which call the Server API:

```
<install_root>\lib\JRESServlets.jar; <install_root>\lib\JREngine.jar; <install_root>\lib
\servlet.jar; <install_root>\lib\hsqldb.jar; <install_root>\lib\log4j-1.2.8.jar;
<install_root>\lib\sac.jar; <install_root>\lib\commons-codec-1.2.jar; <install_root>\lib
\commons-logging-1.1.jar; <install_root>\lib\quartz-all-1.6.5.jar; <install_root>\derby
\lib\derby.jar; <install_root>\derby\lib\derbyclient.jar; <install_root>\derby\lib
\derbynet.ja; <install_root>\derby\lib\derbytools.jar;
```

If you are calling the Server API using RMI from a different JVM or different server than where JReport Server is running, you need to set the classpath environment variable to the remote libraries. Append the following jar files to your class path that compile and run applications which call the Server API:

```
<install_root>\lib\JRSRMI.jar; <install_root>\lib\servlet.jar; <install_root>\lib
\hsqldb.jar; <install_root>\lib\log4j-1.2.8.jar; <install_root>\lib\sac.jar;
<install_root>\lib\commons-codec-1.2.jar; <install_root>\lib\commons-logging-1.1.jar;
<install_root>\lib\quartz-all-1.6.5.jar;
```

Notes:

- If you are not using Derby as your Server DBMS, you need to replace the Derby libraries with the JDBC driver for your DBMS system.
- chart.jar is used in HTML applet and should be placed after jrengine.jar in the class path so as to avoid exceptions.
- If you want to export reports to the following formats, you should add the corresponding class package or jar with a valid path to the class path:
  - To e-mail or use the E-mail Notification function: activation-1.1.jar and mail-1.4.jar.
  - To FTP: commons-net-ftp-2.0.0.jar.
  - To PDF & Page Report Result: itext_1.5.4.jar.
  - To Excel: poiHSSF_151.jar.
  - To XML: xml-apis.jar and xercesImpl.jar.
  - To Page Report Result: JRWebDesign.jar.

Creating and getting instances of ReportEngine

To get an instance of the ReportEngine, you can use the method getInstance() or getInstance(boolean setDebugLevel) in the jet.server.api.engine.ReportEngineFactory.

For example,

```
bean = ReportEngineFactory.getInstance();
```

or,

```
bean = ReportEngineFactory.getInstance(true);
```

Reference: See JReport Javadoc in `<install_root>\help\api` for usage of the ReportEngine methods.

Creating and getting instances of RptServer or HttpRptServer

You can use the following methods to get access to the instance of JReport Server. JReport Server is a singleton - only one instance can exist in the system. Access to the server is through a returned RptServer object or a returned HttpRptServer object, depending on the environment of the request to get access. RptServer is designed to be instantiated in a Java Application while HttpRptServer is designed to be instantiated in a JSP page. HttpRptServer has a few extra methods available to use in the HTTP session.

Method 1

Use the method jet.server.api.http.HttpUtil.initEnv(Properties props).

The method jet.server.api.http.HttpUtil.initEnv(...) creates and initializes the HttpRptServer object. If the HttpRptServer has already been started in this application, it will use the existing HttpRptServer instance. Use this method to avoid creating more than one HttpRptServer instance.

```
        //  prepare report server initial properties.
        //  set report home minimally.
        //  System.getProperties() will bring in all configured settings.
    System.getProperties().put("reporthome", "C:\\JReport\\Server");

        //creates instance of ReportServer or does nothing if one is running already in this JVM
    HttpUtil.initEnv(System.getProperties());

    HttpRptServer server = HttpUtil.getHttpRptServer();

    // do something using the returned server object.
```

Method 2

Call the method HttpUtil.checkLogin(HttpServletRequest req, HttpServletResponse res).

The method jet.server.api.http.HttpUtil.checkLogin(...) implicitly calls HttpUtil.initEnv(...) if necessary. So you can get the HttpRptServer after calling HttpUtil.checkLogin(...).

```
        // use checkLogin to control access to rest of code in JSP file.
    if (HttpUtil.checkLogin(request, response))
        {
            // here, when a JReport user is logged into the current session.

            // get the instance of HttpRptServer .
        jet.server.api.http.HttpRptServer httpRptServer = HttpUtil.getHttpRptServer();

        // do something using the returned httpRptServer object.

        }
```

Using JSP with a dedicated machine

JReport Server can be configured so that JSP pages run on a web server machine while the JReport Server Engine is deployed to another machine. In this configuration, the web server machine is viewed as the client machine accessing the JReport Server system.

When this configuration is used, the Java API library used on the client side is different than the one used when JSP pages run in the same JVM as JReport Server. The method calls are the same, but the library for the client provides stubs for the methods that do a Remote Method Invocation (RMI) to the remote JReport Server to have the processing performed over there. All of the classes that are needed are in `<install_root>\lib\JRSRMI.jar`.

The use of RMI as the way to obtain processing is transparent to the JSP code in the client. Developers write exactly the same code in the JSP file. At deploy time, that JSP code will be bound to the proper library to use for the specific configuration.

By using JSPs in your web applications, the Remote Server API enables you to perform report running tasks remotely, to view the report result directly from the client side, and to administer JReport Server, without having to install JReport Server on the client machine.

The Report Server API is provided by the class jet.server.api.rmi. This class can be used explicitly to call methods that are specific to the configuration where RMI is used.

You can find more information about the jet.server.api.rmi package in the JReport Javadoc located in `<install_root>\help\api`.

Before using the Remote Server API, you must make sure that the machines have been configured properly:

1. Open the RMI service by configuration on the JReport Server side.

   Set the two properties server.rmiserver.enable and server.rmiadminservice.enable to **true** in the server.properties file located in `<install_ root>\bin`.

   When JReport Server resides behind a firewall, you need to specify a fixed port in order to pass through the firewall and obtain the remote objects from the client side by setting the server.rmiserver.fixed_port property in the server.properties file.

2. JReport Server remote host and port information is passed to the client application.

   Set the following parameters as the JVM environment variables in the client side:

   - Djrs.remote.host=<*HOST_NAME/IP_*address>
   - Djrs.remote.rmiport=<*HOST_PORT*>
   - Djrs.rmi.auth_file=<authFileName> (optional)

   You have two alternatives for setting these parameters:

   - You can add -Djrs.remote.host, -Djrs.remote.rmiport and -Djrs.rmi.auth_file (optional) to the batch file that is used to start your application.
   - If you are in an integration environment, you can also add the following to the web.xml file:

```
<context-param>
    <param-name>jrs.remote.host</param-name>
    <param-value>127.0.0.1</param-value>
</context-param>
<context-param>
    <param-name>jrs.remote.rmiport</param-name>
    <param-value>1129</param-value>
</context-param>
<!-- The third param(jrs.rmi.auth_file) is optional. "authFileName" includes the auth
file's realpath and auth file name. For example: C:\JReport\Server\bin\rmi.auth-->
<context-param>
    <param-name>jrs.rmi.auth_file</param-name>
    <param-value>C:\JReport\Server\bin\rmi.auth</param-value>
</context-param>
<listener>
    <listener-class>jet.server.servlets.JRServerContextListener</listener-class>
</listener>
```

Administering JReport Server remotely

JReport Server can be administered from a remote client by Remote Admin Service API. The administration of a JReport Server includes cluster administration, resource administration, security administration, configuration, connection pool management, resource alias management, connection information provider service, and catalog information management.

You can find the Remote Admin Service API in the jet.server.api.rmi.admin package available in the JReport Javadoc which is located in `<install_root>\help\api`.

Before a JReport Server can be performed the remote administration operation on, you need to set the following properties in the server.properties file located in `<install_ root>\bin` in the JReport Server side:

- **server.rmiserver.enable=true**
  Used to open the RMI service.
- **server.rmiadminservice.enable=true**
  Used to open the administration service in the RMI server. By default this property is set to false and the JReport Server does not support remote administration.
- **server.rmiserver.fixed_port=<AnOpenedPort>**
  Required when JReport Server runs behind a firewall.

Using single sign-on in RMI client

When using a Single Sign On (SSO) system, the web application machine is changed. This is where the work is done to set up the Single Sign On system and where to deploy the needed class implementations of the Java interface ExternalAuthorized.

For implementation of SSO in a standalone server, follow the steps below:

1. Write your HttpExternalAuthorized implementation.
2. Compile Java classes. Compiling requires the library JRESServlets.jar, which can be found in `<install_root>\lib` directory.
3. Modify the classpath used by your RMI client to include your external authentication classes, so that your application of RMI can access them.

   If it is in an integration environment, you can add the authentication classes in a folder named classes in the WEB-INF folder.

4. Modify the start file of your RMI client to define the system property jrs.httpExternalAuthorized with your implementation.

   For example, assuming that the implementation of this interface is com.mycorp.HttpExternalAuthorizedImpl.class, use the command line to start the RMI client:

   `java -Djrs.httpExternalAuthorized=com.mycorp.HttpExternalAuthorizedImpl ....`

   Then, restart the RMI client.

Using RMI in JReport Server

Remote Method Invocation (RMI) is used in JReport Server for distributed functionality, such as for clustering, event system, Remote API, and monitoring system. To avoid security risks and the mess of using the RMI registry (all the remote objects bound with the RMI registry directly can cause the remote objects to be unmanageable), **universal remote object management** has been introduced into JReport Server.

In the universal remote object management system, when getting a remote object to perform certain operations, either on a server (local or remote) or in a 3rd-party application (via Remote API), a request must be sent to the remote object management to get that special remote object. Also, authentication is required before any request for a remote object can be processed.

The authentication file

The authentication process requires a remote authentication file that contains the authentication information to accomplish the security check. The remote authentication file (`<install_root>\bin\rmi.auth`) is generated by JReport Server automatically during the server installation process. You can protect your authentication file by moving it to a safe place after installation.

Using the authentication file

The authentication file is commonly used in the following circumstances:

- In JReport Server, the remote object management system will use an authentication file, which resides in the `<server_install_root>\bin` directory.
- In JReport Server Cluster, make sure that the authentication files on all clustered servers are the same. If you want to add a new JReport Server that has a different authentication file to the server cluster, you should first backup the authentication file of that server, and then copy the authentication file from a clustered server in the JReport Server Cluster to `<server_install_root>\bin`.
- This file can then be used in JReport Server Monitor. Copy the authentication file generated by the JReport Server installation process to the directory `<monitor_install_root>\bin`. JReport Server Monitor will then fetch it from `<monitor_install_root>\bin` and build an authInfo object.
- If the Remote API is used, you must specify the authentication file when launching an application that contains the Remote API invocations. For example, `java -cp ... -Djrs.rmi.auth_file=%authFileName% mainClass`. Also, you can specify a string by specifying the content of the authentication file. For example, `java -cp ... -Djrs.rmi.auth_string=my_auth_info mainClass`. Alternatively, you can specify the authentication string yourself arbitrarily. As long as the authentication strings that the local and remote server both hold are the same, you can pass the authentication check: `java -cp ... -Djrs.rmi.auth_string=my_auth_info mainClass`.

  > **Note:** In JReport Server and JReport Server Monitor, you can also use -Djrs.rmi.auth_file or -Djrs.rmi.auth_string in the startup file. JReport Server/JReport Server Monitor looks up these three locations in turn (-Djrs.rmi.auth_string => -Djrs.rmi.auth_file => `<monitor_install_root>\bin`) when checking authentication information. The information first found will be used for the authentication check in the remote object management system.

Ways to generate authentication information

The authentication information can be of any bytes, as long as the authInfo objects match between the local and the remote server. The following are the methods for specifying authentication information:

- You can get the authentication file (rmi.auth) generated when installing the server.
- You can use the RMIAuthFileCreator.bat file stored in `<server_install_root>\bin` to generate the authentication file.
- You can compose any string into the auth file using any text editor.
- You can compose any string as the value of the system property -Djrs.rmi.auth_file or -Djrs.rmi.auth to specify authentication information.

Loading user data source classes at runtime

JReport allows you to create your reports using user data source (UDS), user defined objects (UDO), and user defined formulas (UDF). When you publish the reports using UDS, UDO or UDF to your production environment, JReport requires the UDS, UDO or UDF jars to be in the class path or part of the EAR bundle. This might be difficult for you. You need to shut down your application and wrap the new UDS, UDO or UDF classes into the WAR or EAR or put them in the class path of the application server.

The feature Loading User Classes at Runtime can solve this issue. That is, for standalone JReport Server, or in your embedded application server calling the Server API, you can control loading the classes at runtime without shutting down your application.

Methods

There are two important methods in the Server API that you need to call to dynamically load your UDS classes:

- **String jet.server.api.ServerEnv.getDynamicClassDir()**
  Retrieves the path of the dynamic classes of the report server.
- **static void jet.server.api.engine.ReportEngineFactory.loadClasses(String path)**
  Loads the dynamic classes from the location specified by the path parameter.

Usage

Follow the steps below:

1. Specify a folder location on the machine where JReport Server runs to hold all the UDS/UDO/UDF jar files, so that you can specify to load the classes from that folder. When JReport Server initializes, you can specify the location in one of the following ways:
   - **Using the Server API static void jet.server.api.http.HttpUtil.initEnv(java.util.Properties props)**

     Use the key server.dynamic.class.dir to set the location, for example:

     ```
     System.getProperties().put("server.dynamic.class.dir", "c:\\JReport\\Server\
     \dynamic"); HttpUtil.initEnv(System.getProperties());
     ```

   - **Using the option -Dserver.dynamic.class.dir in the server / application startup file**

     It will be loaded to system properties. Here is an example:

     `-Dserver.dynamic.class.dir=`*`YOUR_FOLDER`*

   - **Using the property entry server.dynamic.class.dir in the server.properties file located in <install_root>\bin**

     The default value of this property is `<install_root>\dynamicclasses`. Here is an example:

     `server.dynamic.class.dir=`*`YOUR_FOLDER`*.

     Note: In general, among the above three approaches, the first has the highest priority, and the third has the lowest.
2. Check the dynamic UDS/UDO/UDF folder any time by calling the method static String jet.server.api.ServerEnv.getDynamicClassDir().

   Usage of the method:

   ```
   HttpUtil.getHttpRptServer().getServerEnv().getDynamicClassDir()
   ```

   Example:

   ```
   System.out.println("Dynamic path"+httpRptServer.getServerEnv().getDynamicClassDir
   ());
   ```

   You can get the path's value in your own properties file. You can even copy your UDS/UDO/UDF jar files to the above folders programmatically.
3. Specify to load classes at runtime by calling the method static void jet.server.api.engine.ReportEngineFactory.loadClasses(java.lang.String path). The parameter path is the dynamic class folder.

   For example:

   ```
   ReportEngineFactory.loadClasses(httpRptServer.getServerEnv().getDynamicClassDir());
   ```

Notes (technical issues):

- Only *.jar or *.zip packages in the specified folder including resource files (recommended *.jar) will be loaded, which excludes the sub directories and unpackaged classes.
- If there are duplicated classes (having the same package and class name) in different *.jar files, the default loading sequence is in time order. The latest modified file will be loaded first, and the older ones will be neglected.
- If you split one dynamic UDS classes into different *.jar or *.zip packages and also want to take advantage of dynamic loader, then you should follow the below rules:
  - The jar with the main class implementing JRUserDataSource class of the dynamic UDS must be in the dynamic class folder.
  - The classes in the *.jar or *.zip packages in the dynamic class folder can call or refer the classes under the JVM's class path, it cannot work in reverse.
- If your UDS classes also go to load the recourse files,
  - You should bundle the resource files into a jar no matter whether it is in the same class jar file or not.
  - In addition, the classes (in the *.jar or *.zip) in the dynamic class folder can get the recourse files both in the dynamic class folder and under the JVM class path. However, the classes under the JVM class path cannot get the resource files in the dynamic class folder.
- When you call the API for loading the classes, JReport Server will not kill the running catalogs and reports but run them using the old UDS classes, and freeze new reports until the loading action has finished.
- JReport Server will try loading dynamic classes in the above way. If failed, it will try loading them in the traditional way (using the class path in the JReport Server startup file, or EAR/EAR in embedded integration environment). Once it fails to find the classes, ClassNotFoundException will be thrown.
- For the UDO classes, only the creator class and render class can be loaded.

Applying a user defined CSS to HTML result file

When exporting a report to HTML format, a .css file will be generated automatically by JReport to control the appearance and layout of the result file. When exporting a report to HTML format with Server API, JReport enables you to apply your own .css file to the HTML result file.

To apply user defined CSS to HTML result file:

1. In JReport Designer, specify the CSS selector in your own CSS for report objects.

   Select the object which you want to apply in your own CSS, and in Report Inspector, specify a value for the CSS property External CSS Class Selector. The value of the property should be the selector in your .css file.

2. Publish the report to JReport Server.

3. Edit your JSP file that invokes the interface used to export the report to HTML format.

4. Export the report to HTML format using JReport Server.

JReport provides a demo JSP file ApplyUserCSS.jsp which is stored in `<install_root>\help\samples\JSPSamples\ApplyUserCSS` for your reference. You can copy the whole ApplyUserCSS folder to `<install_root>\public_html\jinfonet`, then

access the JSP via `http://host:port/jinfonet/ApplyUserCSS/ApplyUserCSS.jsp`.

Specifying paths for the result files when using On-Demand API

When you run a report in the Advanced mode using the on-demand report execution API, you can specify the path for the report result files. See the following demo code:

Demo code of running the report to the PDF format directly

```
//Set report properties
Properties props = new Properties();

//Set result type as .pdf format.
props.put(APIConst.TAG_RESULT_TYPE, String.valueOf(APIConst.PDF));

//Set result file location. The value of TAG_LOCATION_TO_SERVER_RESOURCE_PATH is 0,
//and the value of TAG_LOCATION_TO_SERVER_DISK_REAL_PATH is 1.
//The default value of TAG_RESULT_LOCATION_TYPE is 1.

//props.put(APIConst.TAG_RESULT_LOCATION_TYPE, String.valueOf(APIConst.
TAG_LOCATION_TO_SERVER_RESOURCE_PATH));
//props.put(APIConst.TAG_RESULT_LOCATION, "/");

props.put(APIConst.TAG_RESULT_LOCATION_TYPE, String.valueOf(APIConst.
TAG_LOCATION_TO_SERVER_DISK_REAL_PATH));
props.put(APIConst.TAG_RESULT_LOCATION, "C:\\");
System.out.println("========to pdf=" + props);
tempResult = rptServer.runReport(user, catalog, rptName, props);
```

Demo code of exporting the RST file to HTML format

```
//Set task properties
Properties props = new Properties();
props.put(APIConst.TAG_RESULT_TYPE, String.valueOf(APIConst.RST));
//Set result file location.
props.put(APIConst.TAG_RESULT_LOCATION_TYPE, "0");
props.put(APIConst.TAG_RESULT_LOCATION, "/");
System.out.println("========to pdf=" + props);

String tempResult = server.runReport("admin", catalog, rptName, props);
System.out.println("========tempResult="+tempResult);

//Export to HTML
props.put(APIConst.TAG_HTML, "Crosstab");
props.put(APIConst.TAG_HTML_DIR, "/");
props.put(APIConst.TAG_HTML_DIR_TYPE, String.valueOf(APIConst.TAG_LOCATION_TO_SERVER_RESOURCE_PATH));
System.out.println("\n========exportResult="+props);
ExportedFileList fileList = server.exportResult("admin", tempResult, props);
System.out.println("\n========fileList.getHTMLFilenameList()="+fileList.getHTMLFilenameList());
```

Setting dynamic parameter values when running/scheduling report via API

When running or scheduling a report with parameters via API, you can specify dynamic parameter values by implementing the ParameterGenerator interface. See JReport Javadoc in `<install_root>\help\api` for detailed usage of the API.

To set dynamic parameter values to a report:

1. Create a class MyParameterGeneratorImpl to implement the ParameterGenerator interface. See the API demo DemoParameterGenerator.java in `<install_root>\help\samples\APIParameter` for an example of implementing the ParameterGenerator interface.

2. Add the class path into the class path of your implementation of running or scheduling the report such as DemoParameterRunReport.

3. In DemoParameterRunReport, replace the way of setting static parameter values and instead set dynamic values of the required parameters in the report. Take the parameter PToday for example:

   First disable or delete the line:

   ```
   props.put(APIConst.TAG_PARAM_PREFIX + "PToday", "2007-5-21");
   ```

   and then add a line as follows:

   ```
   props.put(APIConst.TAG_PARAM_PREFIX + "PToday", APIConst.TAG_PARAM_GEN_PREFIX +
   "MyParameterGeneratorImpl";
   ```

Setting values of multi-value parameters via API

To run a report with multi-value parameters via API, take the following example to set values to a multi-value parameter Customers_Country:

```
String[] a = {"USA", "Canada"};
props.put(APIConst.TAG_PARAM_PREFIX + "Customers_Country", a);
```

If you want the parameter to use all its values, take the following:

```
props.put(APIConst.TAG_PARAM_PREFIX + "Customers_Country", new String[]{APIConst.
MULTIPLE_ALL_VALUE});
```

For the API code of running a report, see APIDemoRunReport.java in `<install_root>\help\samples\APIServer`.

Specifying report running properties in the session

When designing integration and security with your application, compared to putting all information in the URL, setting sensitive information in the session and letting the server fetch them when running reports is a more secure way.

JReport provides a key word JReport _running_parameter for holding property variables in the session. Users can set a string with this key word in the session before calling JReport services. The string should take the query string format used in URLs of running reports/dashboards and obey application/x-www-form-urlencoded format and be encoded by UTF-8.

When running reports or dashboards, JReport gets information first from URL and then picks up provided information from the session using the key words. For information that exists in both the URL and the session, the session setting overrides the URL settings.

For running reports via server JSP

For properties both in the URL and the session, the properties in the session will override those in the URL.

For properties only in the session, the properties will be appended to the URL to run the report.

Examples:

- **Set parameter value in the session:**

  HttpSession.setAttribute("JReport_running_parameter", "jrs.param$parameter1=xxx&jrs.param$parameter2=xxx...");

- **Set dynamic connection in the session:**

  HttpSession.setAttribute("JReport_running_parameter", "jrs.jdbc_driver=com.mysql.jdbc.Driver&jrs.jdbc_url=jdbc:mysql://IP:3306/test&jrs.db_user=xxx&jrs.db_pswd=xxx");

For running web reports via JSON

The following rules apply when merging each object from the session and from the URL:

- **jrd_report**
  Session overrides URL. Once this object is specified in the session, only the specified report can be run whenever the URL submits to run any report.

- **jrd_catalog**
  Session overrides URL.

- **jrd_param$**
  Session overrides the same-name parameters and appends the other parameters to the URL.

- **jrd_userinfo**
  Session overrides the properties in the object.

- **jrd_datasources**
  It should contain an object array and use "ds" as the key. The session overrides the objects with the same key and appends the other objects to the array.

Examples:

- **Set parameter value in the session (JSON format):**

  HttpSession.setAttribute("JReport_running_parameter", "jrd_param$={\"parameter1\":\"xxx\", \"parameter2\":\"xxx\"}");

- **Set dynamic connection in the session (JSON format):**

  HttpSession.setAttribute("JReport_running_parameter", "jrd_datasources=[{\"ds\":\"Data Source 1\", \"uid\":\"xxx\", \"pwd\":\"xxx\", \"type\":\"0\", \"url\":\"jdbc:mysql://IP:3306/test\", \"driver\":\"com.mysql.jdbc.Driver\"}]");

For running dashboards via JSON

The following rules apply when merging each object from the session and from the URL:

- **jrd_resext**
  For single properties, session overrides the properties in the URL and adds those not in the URL.

- **reslst**
  It should contain an object array and use "name" as the key. The objects with the same key plus dsh_params and dsh_datasources settings in the session will be appended to the object in the URL. Other objects from the session will be added to the array in the URL.

  For example:

  jrd_resext={"active":0,"reslst":[{"name":"/USERFOLDERPATH/admin/Dashboard1.dsh","dsh_params":[{"jrd_params":{"P_StartDate":"01/01/2006"}}]}]}

  Dashboard1.dsh will run with the parameter P_StartDate="01/01/2006".

  A user can set "JReport_running_parameter" in the session as:

  jrd_resext={"active":0,"reslst":[{"name":"/USERFOLDERPATH/admin/Dashboard1.dsh","dsh_params":[{"jrd_params": {"P_StartDate":"01/01/2007","P_EndDate":"12/31/2007"}}]}]}

  Dashboard1.dsh will then run with P_StartDate="01/01/2007" (applies the value setting from the session) and P_EndDate="12/31/2007" (append the other parameters from the session to the URL) for all library components.

Examples:

- **Set parameter value in the session for a specific library component:**

  HttpSession.setAttribute("JReport_running_parameter", "jrd_resext={\"active\":0,\"reslst\":[{\"name\":\"/62309_Session/62309.dsh\", \"dsh_params\":[{\"jrd_params\":{\"parameter1\":\"xxx\", \"parameter2\":\"xxx\"}}, {\"lc_names\":[\"s13dfc53ea67\"],\"jrd_params\":{\"parameter1\":\"xxx\",\"parameter2\":\"xxx\"}}, {\"lc_names\":[\"s13dfda08c0a\"],\"jrd_params\":{\"parameter1\":\"xxx\", parameter2:\"xxx\"}}]}]}");

- **Set dynamic connection in the session for specific library components:**

  HttpSession.setAttribute("JReport_running_parameter", "jrd_resext={\"active\":0,\"reslst\":[{\"name\":\"/62309_Session/62309_datasource.dsh\", \"dsh_datasources\":[{\"jrd_datasources\":[{\"ds\":\"Data Source 2\", \"uid\":\"root\", \"pwd\":\"1234\", \"type\":\"0\", \"url\":\"jdbc:mysql://IP1:3306/test\", \"driver\":\"com.mysql.jdbc.Driver\"}]}, {\"lc_names\":[\"/COMPONENT_LIB/62309_Session/62309_datasource.lc\"], \"jrd_datasources\":[{\"ds\":\"Data Source 2\", \"uid\":\"root\", \"pwd\":\"1234\", \"type\":\"0\", \"url\":\"jdbc:mysql://IP2:3306/test\", \"driver\":\"com.mysql.jdbc.Driver\"}]}, {\"lc_names\":[\"s13e0cdc04f2\"], \"jrd_datasources\":[{\"ds\":\"Data Source 2\", \"uid\":\"root\", \"pwd\":\"1234\", \"type\":\"0\", \"url\":\"jdbc:mysql://IP3:3306/test\", \"driver\":\"com.mysql.jdbc.Driver\"}]}]}]}");

NLS API

NLS API enables you to run reports in different languages, which includes global NLS API, catalog level NLS API, report level NLS API and resource tree NLS API. For detailed usages about the NLS API, see the JReport Javadoc in `<install_root>\help\api`.

Global NLS API

With the global NLS API, you can add a language, then specify to translate text in another language into a version in this language, or specify to replace a font face and size with another face and size. Likewise, you can get, update, or remove a language, a text translation, or a font replacement.

All text translations and font replacements specified for the global NLS will be applied each time a report runs.

Text is classified into four groups. You will find the four types in `NLSType` in the API Javadoc.

- COLUMN
  Text for columns. This type is for reports running in Page Report Studio.
- LABEL
  Text for labels and web controls.
- PROMPT
  Text for prompt information.
- TOC
  Text for resources displayed in the TOC tree.

For local environment

- Use this interface: `public interface RptServer`.
- Invoke `GlobalNLSLibrary getGlobalNLSLibrary()`.
- Call the methods that can help with your purposes.

For RMI usage

- Use this interface: `public interface RemoteRptServer`.
- Invoke `RemoteGlobalNLSLibrary getGlobalNLSLibrary()`.
- Call the methods that can help with your purposes.

Catalog level NLS API

With the catalog level NLS API, you can translate object names in a catalog to different languages.

For local environment

- Use this interface: `public interface NLSLibraryManager`.
- Invoke `CatalogNLSLibrary getCatalogNLSLibrary()`.
- Call the methods that can help with your purposes.

For RMI usage

- Use this interface: `public interface RemoteNLSLibraryManager`.
- Invoke `RemoteCatalogNLSLibrary getCatalogNLSLibrary()`.
- Call the methods that can help with your purposes.

Report level NLS API

With the report level NLS API, you can translate object names in a report to different languages.

For local environment

- Use this interface: `public interface NLSLibraryManager`.
- Invoke `ReportNLSLibrary getReportNLSLibrary()`.
- Call the methods that can help with your purposes.

For RMI usage

- Use this interface: `public interface RemoteNLSLibraryManager`.
- Invoke `RemoteReportNLSLibrary getReportNLSLibrary()`.
- Call the methods that can help with your purposes.

Resource tree NLS API

With the resource tree NLS API, you can translate object names in the resource tree of JReport Server to different languages.

For local environment

- Use this interface: `public interface RptServer`.
- Invoke `ResourceNLSManager getResourceNLSManager()`.
- Call the methods that can help with your purposes.

For RMI usage

- Use this interface: `public interface RemoteRptServer`.
- Invoke `RemoteResourceNLSManager getResourceNLSManager()`.
- Call the methods that can help with your purposes.

Dashboard Listener API

Dashboard Listener API defines the start and end events of adding and deleting library components for users to get dashboard related information at the event and to use the information in their own application.

Event types

- The start action of adding a library component
- The end action of adding a library component
- The start action of deleting a library component
- The end action of deleting a library component

Basic Information

- Catalog information
- Dashboard information
- Library component information

For detailed usages about the Dashboard Listener API, see JReport Javadoc com.jinfonet.web.modules.dashboard package in `<install_root>\help\api`. Also, there is a sample implementation file DashboardListenerDemo.java in `<install_root>\help\samples\APIDashboardListener`.

Applying the implementations of the Dashboard Listener API

Different uses should have different implementations of the Dashboard Listener API. The implementations can be managed on the server by administrators. After a new implementation is imported to the server and enabled, it will be able to take effect immediately without restarting the server.

Besides, after an implementation is uploaded to the server, you can apply it in the URL which is used for running dashboards. The URL way has higher priority than the enabled implementation on the JReport Administration page.

To import and enable an implementation on the JReport Administration page:

1. Log onto the JReport Administration page.
2. Click **Configuration** on the system toolbar and then select **Dashboard Listener** from the drop-down menu.
3. In the Dashboard Listener panel, click **New Dashboard Listener**.
4. In the New Dashboard Listener dialog, click the **Browse** button to locate the implementation file which could be a zip or a jar file. Then click **Next**.
5. From the Class Name drop-down list, choose a class name which is defined in the implementation file.
6. Click beside the Target box, then in the Select Target dialog, specify the target dashboards you would like the implementation to apply to. Click **OK** to confirm the selection.
7. In the Description text field, enter a description for the implementation if necessary.
8. Select the **Enabled** checkbox to make the implementation enabled.
9. Click **Finish** and the registered class will take effect right away.

To apply an implementation via URL:

To apply the dashboard listener via URL, append the key-value pair "jrd_dsh_listener=PackageName.ClassName" in the URL which is used for running dashboards.

For example, if you want your implemented class TestListener to take effect on a new created dashboard (not the target defined on the JReport Administration page), you can define it in the URL. Such URL will not affect the Dashboard Listener Table on the JReport Administration page.

```
http://localhost:8888/dashboard/app/entry/run.jsp?jrs.authorization=YWRtaW46YWRtaW4%
3D&jrd_dsh_listener=com.jinfonet.TestListener&jrd_resext={"active":0,"reslst":
[{"name":"/USERFOLDERPATH/admin/Dashboard1.dsh","ver":"-1"}]}
```

Dynamic connection API

JReport Server manages dynamic connections by the DynamicConnectionManager. Users can call jet.server.api.admin.AdminService.getDynamicConnectionManager() to get the DynamicConnectionManager object, and call its methods addDynamicConnection(...), updateDynamicConnection(...) or removeDynamicConnection(...) to maintain dynamic connections for multiple tenancy clients by programs.

When running a report, the server calls the DynamicConnectionProvider (the built-in provider calls DynamicConnectionManager internally) to get dynamic connections for all used data sources at first and then passes the dynamic connection properties into the engine. The engine will merge the changed properties with the original catalog connection properties to setup the database connection.

Users can implement jet.server.api.dynamiccon.DynamicConnectionProvider to get dynamic connections from their own systems and then register a customized DynamicConnectionProvider into JReport Server by adding the option server.custom.DynamicConnectionProvider in the server.properties file in `<install_root>\bin`.

For detailed usages about the dynamic connection API, see JReport Javadoc jet.server.api.dynamiccon package in `<install_root>\help\api`.

Dynamic security API

JReport Server provides API for dynamic security management.

- **getDynamicSecurities**
  Gets all dynamic security objects.
- **addDynamicSecurity**
  Adds a new dynamic security object.
- **deleteDynamicSecurity**
  Deletes an existing dynamic security object.
- **getCatalogPath**
  Gets the full path of the catalog resource.
- **getSecurityFileName**
  Gets the name of the security file.
- **getSecurityFile**
  Gets the security file for downloading.
- **setSecurityFile**
  Sets the security file after uploading.
- **setDefault**
  Sets the value of the property IsDefault to true/false.
- **APIConst.TAG_SECURITY_FILE_NAME**
  Defines the security at runtime.

Information Bus API

Information Bus API enables you to access information container and get or put information in the container.

- **InformationBus**
  It is used to transmit information in JReport Server. It contains three kinds of information containers, global level, organization level and user level, and you can get or put information in the containers.

- **InformationBusManager**
  It is used to get the Information Bus instance from the Information Bus Manager.

- **InformationContainer**
  It is used to store user information of the following types:
  - **LONG_TIME**
    The information exists in the container until it is removed or the container it belongs to is removed.
  - **SPECIFIED_TIME**
    The information exists in the container until the time you specify arrives or you remove it, or the container it belongs to is removed.
  - **ONCE_TIME**
    The information will be removed from the container once you get or remove it, or the container it belongs to is removed.

- **InfoLifeCycleType**
  It is used to specify the life cycle type of the information, which can be LONG_TIME, SPECIFIED_TIME OR ONCE_TIME.

Reference: See the JReport Javadoc in `<install_root>\help\api` for detailed usages of Information Bus API.

Related topics:

- Information Bus

API demos

Here is a list of the JReport Server demo programs.

- Programs in `<install_root>\help\samples\APICluster`:
  - **DemoLoadBalancer.java**
    Demo to implement the jet.server.api.cluster.LoadBalancer for randomly selecting a clustered server.
  - **DemoRemoteDispatcher.java**
    An example of dispatching request to remote clustered server according to Round-Robin algorithm.
- Programs in `<install_root>\help\samples\APIParameter`:
  - **DemoParameterGenerator.java**
    An example of implementing the jet.server.api.custom.schedule.ParameterGenerator for setting dynamic parameter values. For details, see Setting dynamic parameter values when running/scheduling report via API.
- Programs in `<install_root>\help\samples\APISecurity`:
  - **CustomizedSendFileAuthorizor.java**
    Demonstrates implementation of SendFileAuthorizor, which is used to control users' accessibility to certain files in server machine when they access the server via http.
  - **DemoAuthenticationProvider.java**
    Demo for implementation of the jet.server.api.custom.security.AuthenticationProvider interface.
  - **DemoAuthorizationProvider.java**
    Demo for implementation of the jet.server.api.custom.security.AuthorizationProvider interface.
- Programs in `<install_root>\help\samples\APISecurity\AddPrincipal`:
  - **AddGroup.java**
    An example of creating groups and getting the current group information.
  - **AddRole.java**
    An example of creating roles and getting the current role information.
  - **AddUser.java**
    An example of creating users and getting the current user information.
  - **SecurityGroupCover.java**
    An example of creating groups and configuring other security settings.
  - **SecurityRoleCover.java**
    An example of creating roles and configuring other security settings.
  - **SecurityUserCover.java**
    An example of creating users and configuring other security settings.
- Programs in `<install_root>\help\samples\APISecurity\LoginLogout`:
  This folder holds JSP source files that experiment with how the two versions of checkLogin() work. The files provide a way to access pages protected by checkLogin() by doing various forms of login to see how the process works. After doing a login, the user can do a logout, in order to establish the same initial state to try another scenario.

  The experiments relate to accessing files that are protected by a call to one of the checkLogin() mehtods which validates that a user is logged into the session or logs a user into the session based on the current HTTP Request.

  Comments in the entry point page, loginIndex.jsp, describe how the files work and how to deploy them for use.
  - **loginIndex.jsp**
    Main entry point for the group of JSP files. This is the dispatcher and home page for the group.
  - **loginDoCheckLogin.jsp**
    This does the work to run the right JSP page that matches the scenario chosen by the user in loginIndex.jsp.
  - **loginUtilCheckLogin.jsp**
    This is the page that is protected by a call to HttpUtil.checkLogin(). After checkLogin() allows it to run, it reports on how it was run and what happened to allow it.
  - **loginSessionCheckLogin.jsp**
    This is the page that is protected by a call to HttpUserSessionManager.checkLogin(). After checkLogin() allows it to run, it reports on how it was run and what happened to allow it.
  - **loginTunOnSSO.jsp**
    This is a helper page to register that the ExternalAuthorized package com.example.MyExternalAuthorized is to be used by JReport Server. This turns on the Single Sign On functionality. This page is only useful (and is only given as a choice by loginIndex.jsp) when the com.exmple.MyExternalAuthorized package is deployed and findable on the classpath. Read the comments in the source to the package found in the SingleSignOn folder next to this one.
  - **loginTunOffSSO.jsp**
    This is a helper page to unset the ExternalAuthorized that is currently set in the server. This turns off the Single Sign On functionality.
  - **loginLogout.jsp**
    This is a helper page that logs out a user from the session in order to have a standard starting state.
- Programs in `<install_root>\help\samples\APISecurity\SingleSignOn`:
  This folder holds sample source files that show examples of implementing a Single Sign On system to allow an existing application and JReport Server to co-operate with logging in users while running on the same web server.

  There are two example implementations of the HttpExternalAuthorized java interface. These implementations provide the Single Sign On functionality. Each file has comments inside that give directions on how to install and use them.
  - **CustomHttpExternalAuthorized.java**
    Implementation of HttpExternalAuthorized to load when JReport Server starts up.
  - **com\example\MyExternalAuthorized.java**
    Implementation of HttpExternalAuthorized written as a package to load on demand while JReport Server is running.

  One file is the source for a servlet that can be used to dynamically set which instance of an HttpExternalAuthorized is used by JReport Server. This would be compiled and deployed using local technique for deploying a servlet.
  - **CustomServlet.java**
    Demo to set implementation of HttpExternalAuthorized into the report server.

  The rest of the files in the folder are JSP files that can be placed into the web server to run through several scenarios that show how an application and JReport Server JSP pages can co-operate to manage user logins within a Single Sign On system. Each file has comments inside that explain what it does and how it fits into the demonstration. These JSP files can be copied to the folder `public_html\jinfonet` where they can be run with a standard install of JReport Server.
  - **customIndex.jsp**
    Entry point JSP page for the demo, to run through the various scenarios that show how Single Sign On behaves. This page includes operations for logging out, to recreate the state of a user not being logged in. It also has controls for turning on and off the Single Sign On functionality.
  - **customViewReport.jsp**
    This is the requested report used in the demo to show how Single Sign On supports authentication for it. It runs a simple report which is displayed in the browser.
  - **customNoAuth.jsp**
    Runs the scenario where nothing special is done in the run-report request to authenticate the user.
  - **customParamAuth.jsp**
    Runs the scenario where the User ID is passed in via a query parameter in the run-report request as a way to authenticate the user.
  - **customSessionAuth.jsp**
    Runs the scenario where the User ID is put into the servlet session variable emulating how the application login would do this, as a way to authenticate the user.
  - **customTurnOnSSO.jsp**
    Helper page that sets the instance of the HttpExternalAuthroized class registered with JReport Server, to turn on the Single Sign On functionality.
  - **customTurnOffSSO.jsp**
    Helper page that unsets the instance of the HttpExternalAuthroized class registered with JReport Server, to turn off the Single Sign On functionality.
- Programs in `<install_root>\help\samples\APIServer`:
  - **APIDemoDeployRpt.java**
    An example of publishing catalogs, reports, or folders to the server.
  - **APIDemoDynamicExportTask.java**
    An example of scheduling a customized task.
  - **APIDemoPublishRpt.java**
    An example of publishing a scheduled task.
  - **APIDemoReportEngine.java**
    An example of using the ReportEngine to run a report, export/print a report, and set the where portion.
  - **APIDemoRunAndExportReport.java**
    An example of exporting a report to some formats, and then saving them to local disk.
  - **APIDemoRunReport.java**
    An example of running a report.
  - **APIDemoRunReportWithTimeout.java**
    An example of running reports with timeout.
  - **APIDemoSendEMail.java**
    An example of publishing a report to e-mail.
  - **DemoTrigger.java**
    This demo creates a trigger, submit a schedule bound with the trigger, then fire the trigger to run the schedule.
  - **ViewHtmlPipelineServlet.java**
    This is a servlet demo of viewing report as HTML pipeline format.
- Programs in `<install_root>\help\samples\APITaskListener`:
  - **TestTaskListener.java**
    An example to demonstrate how to implement users' own scripts before and after task running and how to get the information about a task.
- Programs in `<install_root>\help\samples\JSPSamples`:
  - **ApplyUserCSS**
    An example of applying user defined CSS to the HTML report result. For details, see Applying a user defined CSS to HTML result file.
  - **DHTMLRunAndExport**
    An example of using JSP to build URL to run a report in Page Report Studio. After copying the contents of the folder (demo) to where JSP can be run, for example, `<install_root>\public_html`, you can run the demo using the URL `http://localhost:8888/demo/main.jsp`.
  - **OpenMultipleReports**
    Demo JSPs for opening multiple reports in one session.
  - **PrintReport**
    These JSP files show how to print reports by using ViewerApplet.
  - **RunReport**
    Demo JSPs for runing report in Advanced mode to various formats. To run the demo, copy the two JSP files to `<install_root>\public_html\jinfonet`, and then access getRptDescPage2.jsp (runReport2.jsp is called by getRptDescPage2.jsp) using the URL `http://localhost:8888/jinfonet/getRptDescPage2.jsp`.
  - **JinfonetGourmetJavaDemo**
    Demo JSPs used by the tutorial manual.
- Programs in `<install_root>\help\samples\URLSamples`:
  - **ExampleCodeForTags.html**
    The sample code for different tags.
  - **TestURL.html**
    Some example URLs.

RMI demos

Here are some demo RMI programs that JReport provides. They are available in `<install_root>\help\samples\APIRemoteServer`.

- **RemoteAPIDemoFileService.java**
  A demo about using remote report server.
- **RemoteAPIDemoPublishRpt.java**
  A demo about publishing reports. You can submit a schedule to export a report to version, disk, or e-mail.
- **RemoteAPIDemoRunAndExportReport.java**
  A demo about running reports and exporting report result to other formats.
- **RemoteAPIDemoRunReportWithTimeout.java**
  A demo about the Large Report Manager. This demo runs a report and marks it as a large report by setting the timeout value, and then returns the large report information.

Before running these demos, remember to do the following:

1. Modify the server.properties file in the `<install_root>\bin` directory to set the property server.rmiserver.enable to **true**.
2. Add **-Djrs.rmi.auth_file="%authFileName%"** to the command line that starts the required program.
3. You need to set the classpath environment variable. Append the following jar files to the class path that compile and run applications which call the Server API:

   ```
   <install_root>\lib\JRESServlets.jar;
   <install_root>\lib\JREngine.jar;
   <install_root>\lib\servlet.jar;
   <install_root>\lib\log4j-1.2.8.jar;
   <install_root>\lib\sac.jar;
   <install_root>\lib\commons-codec-1.2.jar
   ```

Page Report Studio API

Page Report Studio API is a set of Java programming interfaces which allow you to customize the Page Report Studio features (completely or partially) so that you can use it more conveniently.

You can refer to the built-in JSP dhtml.jsp that comes with JReport Server as a demo. It is located in `<intall_root>\public_html\dhtmljsp`.

For Page Report Studio API specifications, see the jet.web.dhtml package in JReport Javadoc that is located in `<install_root>\help\api`.

This section talks about the following three topics:

- Installing the Page Report Studio API
- Getting started with the Page Report Studio API
- Advanced Page Report Studio API application

Installing the Page Report Studio API

When you install JReport Server, JReport Server API including the Page Report Studio API is also installed at the same time. After installation, you have the following files:

- In `<install_root>\lib`:
  - JRESServlets.jar
  - JREngine.jar
  - servlet.jar
  - hsqldb.jar
  - log4j-1.2.8.jar
  - sac.jar
  - commons-codec-1.2.jar
  - commons-logging-1.1.jar
  - quartz-all-1.6.5.jar
- In `<install_root>\derby\lib`:
  - derby.jar
  - derbyclient.jar
  - derbynet.jar
  - derbytools.jar

JReport Server API classes are stored in the archive file JRESServlets.jar.

You will need to set the classpath environment variable. Append the following jar files to your class path that compile and run applications which call the Server API:

For Server:

```
<install_root>\lib\JRESServlets.jar; <install_root>\lib\JREngine.jar; <install_root>\lib
\servlet.jar; <install_root>\lib\hsqldb.jar; <install_root>\lib\log4j-1.2.8.jar;
<install_root>\lib\sac.jar; <install_root>\lib\commons-codec-1.2.jar; <install_root>\lib
\commons-logging-1.1.jar; <install_root>\lib\quartz-all-1.6.5.jar; <install_root>\derby
\lib\derby.jar; <install_root>\derby\lib\derbyclient.jar; <install_root>\derby\lib
\derbynet.jar; <install_root>\derby\lib\derbytools.jar;
```

For Page Report Studio, also include:

```
<install_root>\lib\JRWebDesign.jar;
```

Note: If you are not using Derby as your server DBMS, you need to replace the Derby libraries with the JDBC driver for your DBMS system.

Getting started with the Page Report Studio API

How to get an instance of Page Report Studio API

Note that before you can use the following methods to get a Page Report Studio API instance, firstly you need to import some relevant packages. For instance, jet.web.dhtml.* must be included here.

- **For local use**

  Both of the following methods can be used to get an instance of the Page Report Studio API locally. The results of these two methods are exactly the same.

  - Method 1:

    ```
    DHTMLClientService service = DHTMLServiceFactory.getDHTMLClientService(request);
    ```

  - Method 2:

    ```
    HttpRptServer server;
    if (System.getProperty(APIConst.REMOTE_DISPATCHER) == null) {
        server = HttpUtil.getHttpRptServer();
    } else {
        server = RemoteDispatcherFactory.getInstance().getDispatchedServer(request);
    }
    DHTMLClientService service = (DHTMLClientService)server.getDHTMLClientService();
    ```

- **For RMI use**

  This method can be used to get an instance of Page Report Studio API from a remote server (RMI).

  ```
  HttpRptServer server;
  if (System.getProperty(APIConst.REMOTE_DISPATCHER) == null) {
      server = HttpUtil.getHttpRptServer();
  } else {
      server = RemoteDispatcherFactory.getInstance().getDispatchedServer(request);
  }
  DHTMLClientService service = server.getDHTMLClientService();
  ```

Advanced Page Report Studio API application

Besides fundamental functions that Page Report Studio API provides, you can further obtain other instances from DHTMLClientService.

How to get the other instances

```
//Get a Page Report Studio API instance using the method 1 or
//2 provided in the preceding section, and then
//use the following method to get the other instances.

//The instance "service" is supposed to be derived
//from the above method 1 or 2.

 DHTMLConfig config = service.getDHTMLConfig();
 DHTMLDataInfo dataInfo = service.getDHTMLDataInfo();
 DHTMLWriter writer = service.getDHTMLWriter(sessionId, rptSetId)
```

How to use the other instances

Customizing Page Report Studio toolbar using the Page Report Studio API

Page Report Studio API provides you with the Page Report Studio interface. With the Page Report Studio API, you can easily embed the Page Report Studio page or components into your own JSP. Here jet.web.dhtml.DHTMLConfig is used to customize your own Page Report Studio toolbar.

- **Method:** jet.web.dhtml.DHTMLConfig.customizeToolbar (java.lang.String sessionId, java.lang.String rptSetId, java.lang.String toolbarname, int[] buttonId, boolean isVisible)

  This method is used to specify the buttons displayed on the Page Report Studio toolbar.

  Parameters:

  - **sessionId**
    The session ID of the current user or current request.
  - **rptSetId**
    The report ID of the current user or current request.
  - **toolbarname**
    Name of the toolbar, such as "Standard", "View", "Analysis", and user-defined ones.
  - **buttonId**
    The buttonId constant for the toolbar. The buttonId argument is an integer bitmask that can be DHTMLConstant.TOOLBAR_UNDO, DHTMLConstant.TOOLBAR_REDO, DHTMLConstant.TOOLBAR_SEARCH, DHTMLConstant.TOOLBAR_PREVIOUS_SEARCH, DHTMLConstant.TOOLBAR_NEXT_SEARCH, DHTMLConstant.TOOLBAR_EXPORTTOHTML, DHTMLConstant.TOOLBAR_EXPORTTOPDF, DHTMLConstant.TOOLBAR_EXPORTTOXML, DHTMLConstant.TOOLBAR_EXPORTTORTF, DHTMLConstant.TOOLBAR_EXPORTTOXLS, DHTMLConstant.TOOLBAR_EXPORTTOTEXT, DHTMLConstant.TOOLBAR_EXPORTTOPS, and so on, or a bitwise union of them (for example, DHTMLConstant.TOOLBAR_UNDO | DHTMLConstant.TOOLBAR_REDO).
  - **isVisible**
    If it is true, the buttons are visible.

- **Method:** jet.web.dhtml.DHTMLConfig.setUsedToolbar (java.lang.String sessionId, java.lang.String rptSetId, java.lang.String rptName, java.lang.String toolbarName, boolean isShow)

  This method is used to set the toolbar's default status: shown or hidden.

  Parameters:

  - **sessionId**
    The session ID of the current user or current request.
  - **rptSetId**
    The report ID of the current user or current request.
  - **rptName**
    The report name of the current user or current request.
  - **toolbarname**
    Name of the toolbar, such as "Standard", "View", "Analysis", and user-defined ones.
  - **isShow**
    If it is true, the toolbar is visible.

To customize the Page Report Studio toolbar, you must add the following code to run_report.jsp in `<intall_root>\public_html\dhtmljsp`. And this code must be after the RunReport() method.

```
DHTMLClientService service = DHTMLServiceFactory.getDHTMLClientService(request);
DHTMLConfig     dhtmlConfig = service.getDHTMLConfig();
String SessionID = DHTMLUtil.getSessionID(request);
String RptSetId = DHTMLUtil.getRptSetId(request);
int[] bId = new int[]{DHTMLConstant.TOOLBAR_EXPORTTOPDF,DHTMLConstant.TOOLBAR_EXPORTTOXML,DHTMLConstant.
TOOLBAR_SEARCH};
dhtmlConfig.customizeToolbar(SessionID, RptSetId, "View", bId, true);
dhtmlConfig.setUsedToolbar(SessionID, RptSetId, null, "View", true);
```

Reference: For more information on this topic, see the JReport Javadoc located in `<install_root>\help\api`.

JReport Catalog API

JReport Catalog API is intended for those who want to work with a catalog programmatically. You can create a catalog and manage the objects in it with the methods provided by the Catalog API instead of using the GUI design mode in JReport Designer. Compared with the Catalog Bean, the Catalog API offers greater functionality. As with a catalog in JReport Designer, the Catalog API provides methods which enable you to:

- Create a connection.
- Add and delete objects such as tables, queries, formulas, parameters, and WHERE portions.
- Modify a query and update UDS.
- Get and modify object information.

The combination of the Catalog API and the Design API provides you with a Java development environment that enables you to create any report to meet your requirements.

Pick a task from the following for details about how to use the Catalog API:

- Installing the Catalog API
- Programming with the Catalog API

Installing the Catalog API

When you install JReport Designer, which by default is located in `C:\JReport\Designer`, the Catalog API is also installed at the same time. After installation, you will have the following components in `<install_root>\lib`:

- report.jar
- JREngine.jar
- log4j-1.2.8.jar (Jar file used for JReport Engine logging system)
- sac.jar (Jar file used for installing Catalog API with Design API product)
- ...

The Catalog API class is stored in the archive file report.jar. You will need to do the two steps below before you can compile and run the program:

1. Set the class path environment variable. Append the following string to your class path that starts the Catalog API:

   ```
   <install_root>\lib\report.jar;<install_root>\lib\sac.jar;<install_root>\lib
   \JREngine.jar;<install_root>\lib\log4j-1.2.8.jar
   ```

2. In your Java program you must set the licensed user and Design API license key.

   ```
   DesignerUserInfo userInfo=new DesignerUserInfo("UID","Design API Key");
   ```

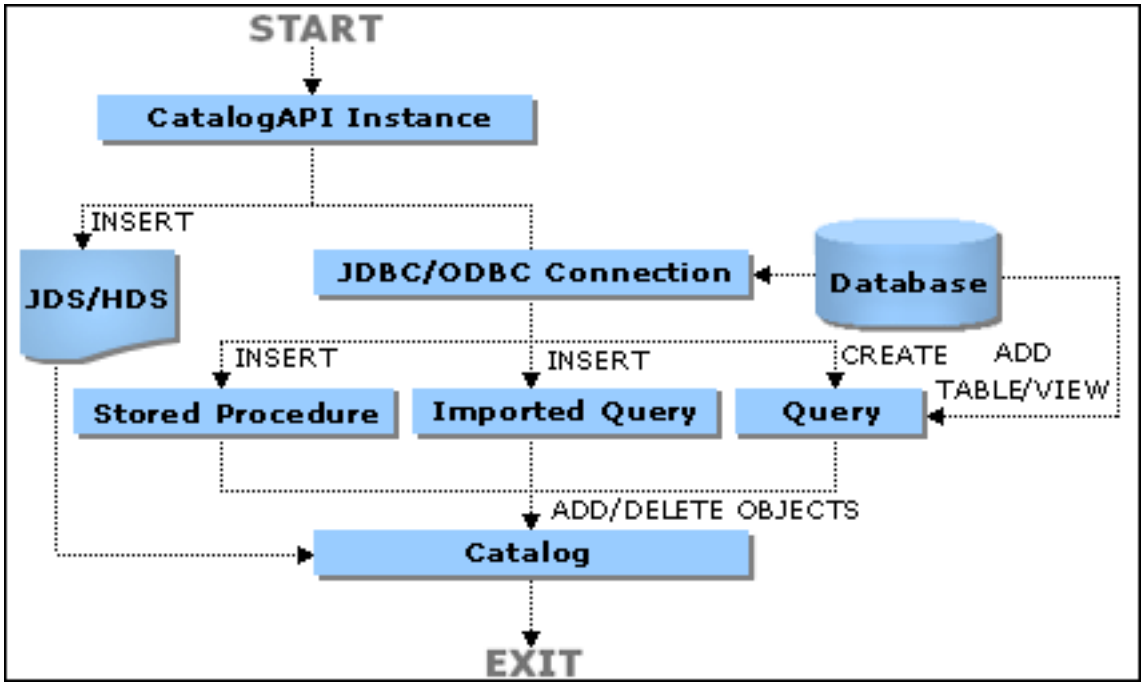
3. Set the reporthome parameter equal to the root path of JReport Designer. You can set this in your environment or with the -Dreporthome argument to your JVM.

JReport Server also includes the Catalog API. If you prefer to use the libraries from JReport Server include the jar file <server_install_root>\lib\JRESServlets.jar in your class path instead of report.jar. You would then need to use the Server Design API license key rather than the Design API license key.

Programming with the Catalog API

JReport Catalog API is the principal class that manipulates and accesses catalogs. The application creates a Catalog API instance, and then uses the Catalog API methods to operate the catalog. For more details about summaries of FIELD, CONSTRUCTOR and METHOD, see the JReport Javadoc jet.api package in `<install_root>\help\api`, which includes all documents generated by Javadoc. Also, you are provided with some sample files in `<install_root>\help\samples\APICatalog`.

With the methods in JReport Catalog API, you can do the following:



Select a task from the following for details:

- Preparations
- Using the Catalog API
- An illustrative example

Preparations

Before you can use the Catalog API to perform certain tasks, you will have to create a Designer object and then get a Catalog API instance.

Creating a Designer object

To create a Designer object, use the constructor Designer(String path, String name, DesignerUserInfo user) in the Design API. The constructor has three parameters: the catalog path, catalog name, and the user ID and license key provided by Jinfonet.

The path needs to be a valid path to an existiing directory. The catalog name can be the name of an existing catalog when you want to open a catalog, or the name of a new catalog when you want to create a catalog. If you want to create a new catalog, the path directory should not already contain a catalog file.

To create the DesignerUserInfo instance, use the following constructor providing the user ID and the Server Designer License Key or Designer License Key you received when you purchased JReport.

```
DesignerUserInfo userInfo=new DesignerUserInfo(Uid, key);
```

Getting a Catalog API instance

To get a Catalog API instance, use the method getCatalogAPI() in the Design API. You need to first get an instance from Designer as follows.

```
Designer desg = new Designer(catalogPath, catalogName, userInfo);
CatalogAPI catalog = desg.getCatalogAPI();
```

Using the Catalog API

After creating a Designer object and getting a Catalog API instance, you can then use the Catalog API to perform tasks as required. For example, you can use it to set up a connection in a catalog, insert an object into the catalog, and modify the query of the catalog.

The following sections describe how to program with the Catalog API:

- Creating a connection
- Inserting an object
- Inserting a table/view
- Inserting a stored procedure
- Importing an SQL file
- Inserting a UDS
- Inserting a query
- Inserting a business cube, report cube or business view
- Inserting a formula, summary, parameter or a WHERE portion
- Deleting an object
- Modifying a query
- Getting objects in a catalog
- Refreshing the reference table of a catalog
- Saving a catalog

Creating a connection

The following methods are used to insert a database connection and return the handle of the newly inserted connection. If not successful, a null value is returned:

- insert(boolean bUseJDBCODBCBridge, String name, String desc, String url, String driver)
- insert(boolean bUseJDBCODBCBridge, String name, String desc, String url, String driver, ConnectionInfo info)
- insert(String dataSourceName, boolean bUseJDBCODBCBridge, String name, String desc, String url, String driver)
- insert(String dataSourceName, boolean bUseJDBCODBCBridge, String name, String desc, String url, String driver, ConnectionInfo info)

Parameters

- UseJDBCODBCBridge - Indicates to connect to the database with a JDBC-ODBC Bridge.
- name - The name of the connection.
- desc - The description for the connection.
- url - The URL used to connect to the database.
- driver - The database driver.
- info - The connection information.

Note: The class ConnectionInfo is used to create the connection information. For details, refer to the JReport Javadoc jet.api package in `<install_root>\help\api`, which includes all documents generated by Javadoc.

Inserting an object

The following methods are used to insert an object into a catalog and to return the handle of the newly inserted object:

- insert(String name, int type)
- insert(String dataSourceName, String name, int type)

Parameters

- name - The object name.
- type - The object type. For more information, refer to the JReport Javadoc jet.api.CatalogAPI class in `<install_root>\help\api`, which includes all documents generated by Javadoc.

Inserting a table/view

The following methods are used to insert a table, view or synonyms into a catalog and to return the handle of the newly inserted object:

- insert(String catalogName, String schemaPattern, String tableName, int type)
- insert(String catalogName, String schemaPattern, String tablePattern, int type, boolean setMappingName)

Parameters

- catalogName - The catalog name.
- schemaPattern - The schema pattern.
- tablePattern - The table name.
- type - The table type.

Inserting a stored procedure

The following methods are used to insert a stored procedure into a connection and to return the handle of the newly inserted procedure.

- insert(String procName,String catalog,String schema,String name,String remarks,int iType)
- insert(String procName, String catalog, String schema, String name, String remarks, int iType)

Parameters

- procName - The procedure mapping name.
- catalog - The catalog name of the procedure.
- schema - The schema name of the procedure.
- name - The procedure name.
- remarks - The remarks on the procedure.
- iType - The result type of the procedure.

Importing an SQL file

The following methods are used to import an SQL file and to return the handle of the newly inserted SQL file:

- insert(String SQLName, String filename)
- insertSql(String dataSourceName, String SQLName, String filename)

Parameters

- SQLName - The SQL name.
- filename - The imported file name.

Inserting a UDS

The following methods are used to insert a user defined data source and to return the handle of the newly inserted UDS:

- insert(String strDSName, String strClassName, String strParameter, UDSColumnInfo udsColInfo)
- insert(String dataSourceName, String strDSName, String strClassName, String strParameter, UDSColumnInfo udsColInfo)

Parameters

- strDSName - The UDS name.
- strClassName - The name of the UDS class.
- strParameter - The parameter value.
- udsColInfo - The specified column information.

Inserting a query

The following methods are used to insert a query into a catalog and to return the handle of the newly inserted query:

- insert(String qryName, QueryFieldInfo queryFieldInfo,QueryJoinInfo queryJoinInfo, QueryQBEInfo queryQBEInfo, QueryAndInfo queryAndInfo)
- insert(String dataSourceName, String qryName, QueryFieldInfo queryFieldInfo, QueryJoinInfo queryJoinInfo, QueryQBEInfo queryQBEInfo, QueryAndInfo queryAndInfo)

Parameters

- qryName - The query name.
- queryFieldInfo - The table and fields information.
- queryJoinInfo - The join information.
- queryQBEInfo - The QBE information.
- queryAndInfo - The where condition information.

Reference: For details about the classes QueryFieldInfo, QueryJoinInfo, QueryQBEInfo and QueryAndInfo, refer to the JReport Javadoc jet.api package in `<install_root>\help\api`, which includes all the documents generated by Javadoc.

Inserting a business cube, report cube or business view

The following methods are used to insert a business cube or report cube into a catalog and to return the handle of the newly inserted object if it is inserted successfully, otherwise null if there is a failure:

- insertBusinessView(String dataSourceName, String queriableName, String businessViewName)
- insertBusinessView(String dataSourceName, String queriableName, String businessViewName, boolean isLogicView=true)

To insert a business view, you can only use the following method:

- insertBusinessView(String dataSourceName, String queriableName, String businessViewName, boolean isLogicView=false)

Parameters

- dataSourceName - The mapping name of the data source in the catalog to which you want to insert the object.
- queriableName - The queriable object's mapping name. The queriable object in JReport catalog could be a query, UDS, stored procedure or imported SQL. If queriableName is a valid name of an existing queriable object, a report cube or business view will be created; if it is null or does not exist, a business cube will be created.
- businessViewName - The name of the business cube, report cube or business view.
- isLogicView - Indicates whether to insert a business cube/report cube or a business view. If it is true, a business cube/report cube will be created, otherwise, a business view will be created.

Adding elements to business cube/report cube •Deprecated•

When you have a business cube or report cube created successfully, you can add elements into it. A business cube/report cube can contain the following elements: categories, dimension objects, measure objects, and detail information objects.

The following methods are used to insert a cube element into a business cube or report cube and to return the handle of the newly inserted object if it is inserted successfully, otherwise null if there is a failure:

- insertBLCategory(String parentHandle, BLCategoryInfo info)
- insertBLDimension(String parentHandle, BLDimensionInfo info)
- insertBLMeasure(String parentHandle, BLMeasureInfo info)
- insertBLDetail(String parentHandle, BLDetailInfo info)

Parameters

- parentHandle - The handle of the parent object. It can be a business cube/report cube or a category.
- info - The definition class of the cube element. It can contain its children's definition.

Adding elements to business view

Similarly, you can add elements into a business view using the following methods, which include categories, group objects, aggregation objects and detail objects:

- insertBusinessViewCategory(String parentHandle, BusinessViewCategoryInfo info)
- insertBusinessViewGroup(String parentHandle, BVGroupInfo info)
- insertBusinessViewAggregation(String parentHandle, BVAggregationInfo info)
- insertBusinessViewDetail(String parentHandle, BVDetailInfo info)

Parameters

- parentHandle - The handle of the parent object. It can be a business view or a category.
- info - The definition class of the view element. It can contain its children's definition.

See an example: The sample TestCatalogBC.java in `<install_root>\help\samples\APICatalog` shows how to create a report cube using the Catalog API.

Inserting a formula, summary, parameter or a WHERE portion

The following methods are used to insert a formula, summary, parameter or a WHERE portion into a catalog and to return the handle of the newly inserted object:

- insert(String formulaName, String desc, String expression)
- insert(String summaryName, String desc, int functionType, String fieldName, String groupByFld)
- insert(String parameterName, String desc, String prompt, String type, String default Value)
- insert(String wherePortionName, String desc, WherePortionInfo whereportionInfo)

Parameters

- formulaName - The name of the formula.
- desc - The description of the object.
- expression - The expression.
- summaryName - The name of the summary.
- functionType - The type of the function.
- fieldName - The field name calculated on.
- groupByFld - The field name grouped by.
- parameterName - The parameter Name.
- prompt - The input text prompt.
- type - The parameter data type.
- defaultValue - The defaultValue of the parameter.
- wherePortionName - The WHERE portion name.
- whereportionInfo - The WHERE portion information.

Deleting an object

The method delete(Stringhandle) is used to delete an object from its parent node and to return true if the specified object is removed.

Parameter

- handle - The handle of the object that is to be deleted.

Modifying a query

You can perform the following manipulations when modifying a query.

Adding or deleting a table

- **set(String dataSourceName, String qryName, String tablename, String columnname, boolean isFormula)**
  Adds a new table to a query and returns a Boolean value: true if successful, otherwise false.
- **deleteQueryTable(String dataSourceName, String qryName, String tablename)**
  Deletes the selected table in a query and returns a Boolean value: true if successful, otherwise false.

Note: The tables that are to be added should exist in the specified catalog.

Parameters

- qryName - The query name.
- tablename - The name of the table that is to be inserted or deleted.
- selectAllFields - Specifies whether or not to insert a table with all its fields.

Adding or deleting a field

- **set(String dataSourceName, String qryName, String tablename, String columnname, boolean isFormula)**
  Adds a new field to a query and returns a Boolean value: true if successful, otherwise false.
- **deleteQueryField(String dataSourceName, String qryName, String tablename, String columnname)**
  Deletes the selected fields in a query and returns a Boolean value: true if successful, otherwise false.

Parameters

- qryName - The query name.
- tablename - The table name of the inserted or deleted field.
- isFormula - Indicates whether the inserted field is a formula.
- columnname - The column name of the deleted field.

Adding or deleting a join

- **set(String dataSourceName, String qryName, String tableFrom, String columnFrom, String operator, String tableTo, String columnTo, boolean isSQL92, int outerJoin)**
  Adds a new join to a query and returns a Boolean value: true if successful, otherwise false.
- **deleteQueryJoin(String dataSourceName, String qryName, String tableFrom, String columnFrom, String operator, String tableTo, String columnTo)**
  Deletes joins in a query and returns a Boolean value: true if successful, otherwise false.

Parameters

- qryName - The query name.
- tableFrom - The name of the table from which the join links.
- columFrom - The name of the column from which the join links.
- tableTo - The name of the table to which the join links.
- columnTo - The name of the column to which the join links.
- operator - Operator of the join to be inserted or deleted.

Adding or deleting a QBE condition

- **setQBE(String dataSourceName, String qryName, String tablename, String columnname, String expression)**
  Adds a new QBE condition to a query and returns a Boolean value: true if successful, otherwise false.
- **deleteQBE(String dataSourceName, String qryName, String tablename, String columnname, String expression)**
  Deletes the QBE condition in a query and returns a Boolean value: true if successful, otherwise false.

Parameters

- qryName - The query name.
- tablename - The name of the table where the QBE condition will be added or deleted.
- columnname - The name of the column where the QBE condition will be added or deleted.
- expression - The expression of the QBE condition.

Adding or Deleting a where condition

- **setCondition(String dataSourceName, String qryName, String sExpression1, String sOperator, String sExpression2, String sLogic)**
  Adds a new where condition to a query and returns a Boolean value: true if successful, otherwise false.
- **deleteCondition(String dataSourceName, String qryName, String sExpression1, String sOperator, String sExpression2, String sLogic)**
  Deletes the where conditions in a query and returns a Boolean value: true if successful, otherwise false.

Parameters

- qryName - The query handle.
- sLogic - The logic string of the where condition that is to be inserted or deleted.
- sExpression1 - The first expression of the where condition that is to be inserted or deleted.
- sOperator - The operator of the where condition that is to be inserted or deleted.
- sExpression2 - The second expression of the where condition that is to be inserted or deleted.

Getting objects in a catalog

Getting tables

- **getTables(String dataSourceName)**
  Gets tables, views and synonyms in a connection and returns the mapping name array.
- **getTables(boolean bIncludeSystemTable)**
  Gets tables, views and synonyms in a data source and returns the name array.

Getting stored procedures

- **getProcedureNames(String dataSourceName)**
  Gets all the stored procedures in a catalog and to return the mapping name array.

Getting SQLs

- **getSQLs(String dataSourceName)**
  Gets all the SQL files in a catalog and to return the mapping name array.

Getting UDSs

- **getUDSs(String dataSourceName)**
  Gets all the UDSs in a catalog and to return the mapping name array.

Getting queries

- **getQueries(String dataSourceName)**
  Gets queries in a catalog and returns the mapping name array.
- **getNewQueryName()**
  Gets a default name for a new query, which must be unique. The name for the first query in a catalog is query1. The following query is query2, then query3 and so on.

Getting columns

- **getColumns(String dataSourceName, String queryName, boolean allColumns)**
  Gets columns in a query and returns the table name and column real names.
- **getColumnMappingnames(String dataSourceName, String queryName)**
  Gets column mapping names in a query and returns the table name and column mapping names.
- **getColumnsCanBeGroupedBy(String dataSourceName, String queryName)**
  Gets the columns which can be grouped by in a query.

Parameters

- queryName - The query name.
- allColumns - Selects all columns in the specified tables.

Getting fields

- **getDBFields(String dataSourceName, String qryName)**
  Gets DBFields that can be used in a query and returns the mapping name array.
- **getMappingnames(String dataSourceName, String DSName)**
  Gets the field mapping names in a data source and returns the table name and column mapping names.

Parameters

- qryName - The query name.
- DSName - The data source object name, such as Query, SQL, UDS, and Procedure.

Getting formulas, summaries or parameters

- **getFormulae(String dataSourceName)**
  Gets formulae that can be used in a catalog and returns the mapping name array.
- **getFormulaeCanBeGroupedBy(String dataSourceName, String queryName, String groupBy)**
  Gets formulae that are valid to a query and returns the formula names.
- **getSummaries(String dataSourceName)**
  Gets summaries that can be used in a catalog and returns the mapping name array.
- **getSummaryCanBeSortedBy(String dataSourceName, String queryName, String groupBy)**
  Gets summaries that can be sorted by.
- **getParameter(String dataSourceName)**
  Gets parameters in a catalog and returns the mapping name array.
- **getParameterCanBeGroupedBy(String dataSourceName)**
  Gets parameters that can be grouped by.

Parameters

- queryName - The query name.
- dataSourceName - The datasoure name.
- groupBy - The groupBy field.

Getting joins

- **getJoins(String dataSourceName, String queryName)**
  Gets joins in a query and returns the join info array.

  Note: String[i][0] = Table "From" Name, String[i][1] = Column "From" Name, String[i][2] = Table "To" Name, String[i][3] = Column "To" Name

Parameter

- queryName - The query name.

Getting AND or QBE condition

- **getAndConditions(String dataSourceName, String queryName)**
  Gets AND conditions in a query and returns a condition info array.

  Note: String[i][0] = "Logic", String[i][1] = "Expression1" String[i][2] = "Operator", String[i][3] = "Expression2"
- **getQBEInfo(String dataSourceName, String queryName)**
  Gets the QBE condition in a query and returns a QBE info array.

  Note: String[i][0] = column Name, String[i][j](j>=1) = condition expression of this column.

Parameter

- queryName - The query name.

Getting report names

- **getNewReportName()**
  Gets a default name for a new report, which must be unique. The name for the first report in a catalog is report1, after which the following reports will be named report2, report3 and so on.

•

Refreshing the reference table of a catalog

You can use the method public boolean refreshReference(CrossReferenceConfigure referenceConfigure) to refresh the reference table of a catalog.

Parameter

- CrossReferenceConfigure referenceConfigure - The configuration information that defines which data sources in a catalog and which types of object in the data sources will be added to the reference table.

Saving a catalog

The following methods are used to save catalogs:

- **save()**
  Saves an open catalog and returns a Boolean value: false if it fails, otherwise true.
- **saveAs(String path, String name)**
  Saves an open catalog as a new catalog file. You can save the catalog as a binary catalog using the extension .cat, or an XML format catalog using the extension .cat.xml.

An illustrative example

You are provided with a sample program that demonstrates how to use the Catalog API to create or modify a catalog - TestCatalogAPI.java that is located in `<install_root>\help\samples\APICatalog`.

Here, TestCatalogAPI.java has been used as an example to explain how the methods work in the Catalog API.

- **Creating a Designer object**

```
DesignerUserInfo userInfo=new DesignerUserInfo("UID",
"XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX");
desg = new Designer(path, catName, userInfo);
```

  If the catalog named catName (MyReports.cat) exists, it will be opened. Otherwise it will be created.

- **Getting a CatalogAPI instance**

```
CatalogAPI cat = dr.getCatalogAPI();
```

- **Creating a connection**

```
ConnectionInfo info = new ConnectionInfo("sa", "");
String con = cat.insert("", false,"Jinfonet demo", "demo db",
"jdbc:hsqldb:C:\\JReport\\Designer\\Demo\\db\\JRDemo", "org.hsqldb.jdbcDriver", info);
```

- **Inserting objects into a catalog**

  For example, insert a table named Customers:

```
tb.addTable("", null, "PUBLIC", "CUSTOMERS", "Customers");
```

- **Inserting a formula named "total"**

```
String formulahandles=cat.insert("total",cat.FORMULA);
FormulaInfo formulainfo=new FormulaInfo("total","total formula","@\"Unit Price\" *
@Quantity - @Discount");
cat.set(formulahandles,formulainfo);
```

- **Importing an SQL file**

```
cat.insert("SQL1","sqlfile.txt");
```

- **Saving**

```
cat.save();
```

- **Compiling and running the demo program**

  To compile and run the demo program, you should add the required classes to the class path.

  After running the program, a new catalog file MyReports.cat will be created in `C:\JReport\Designer\Demo\MyReports`. Be sure to create this empty directory first.

```
C:\JReport\Designer\help\samples\APICatalog>javac -classpath "C:\JReport\Designer\lib
\report.jar;C:\JReport\Designer\lib\sac.jar;C:\JReport\Designer\lib\JREngine.jar;C:
\JReport\Designer\lib\log4j-1.2.8.jar;C:\test\classes" TestCatalogAPI.java
```

```
C:\JReport\Designer\help\samples\APICatalog>java -classpath "C:\JReport\Designer\lib
\jrengine.jar;C:\JReport\Designer\lib\sac.jar;C:\JReport\Designer\lib\report.jar;C:
\JReport\Designer\lib\log4j-1.2.8.jar;C:\JReport\Designer\help;C:\test\classes" -
Dreporthome=C:\JReport\Designer TestCatalogAPI -path=C:\JReport\Designer\Demo
\MyReports -catalog=MyReports.cat -log=C:\JReport\Designer\logs\designer.log
```

•

JReport Design API

Besides editing a report with the design panel in JReport Designer, you can also edit a report with the Design API. By creating a Designer instance in your Java program, you can modify a report with the methods provided by the Design API. The Design API is for designing a report with your own application.

JReport Design API fully realizes designing a report programmatically. That is, with the Design APIs you can edit the layout of a report all in code. You can change the property values, or modify the report by inserting objects such as DBFields, formulas, text objects, and charts. You can also create and design a report with the Design API.

The following sections explain the Design API in detail:

- Design API packages & license
- Installing the Design API
- Getting started
- Programming with the Design API

Note: The Design API only supports Page Reports (.cls) and does not support Web Reports (.wls) and Library Components (.lc).

•

Design API packages & license

There are two JReport Design API packages: the workstation version and the server version. The workstation package is the Design API, and the server package is the Server Design API. The main differences between these two packages are:

- There are some special methods used for editing catalogs, queries, reports, and parameters. These methods are included in the Design API and Server Design API. When invoking these methods in the Server Design API, you also have to provide an additional parameter: UID (used to distinguish between different users, and can be a Number or a String). However, when invoking them in the Design API, you do not need to provide this parameter.
- The Design API can only edit one report at a time, while the Server Design API allows concurrent users to modify their reports.
- The Multi-User Design API included with Server Design API only supports V7 and earlier style report templates and not multi-tab page reports supported by Design API.

JReport Design API and JReport Server Design API use an independent license from other JReport products. You need to contact Jinfonet sales to obtain this special license key if you want to use either JReport Design API or JReport Server Design API on your machine with JReport Designer or JReport Server. The license will be applied to your program instead of being applied to the installation of JReport Design API or JReport Server Design API.

The keys for JReport Design API and JReport Server Design API are different. For example, you cannot use the JReport Design API key on JReport Server Design API, and vice versa. Which key is required is based on where you set your class path at runtime. If the classes are obtained from Server\lib you need to use the Server Design API key. If the classes are obtained from Designer\lib you need to use the Design API key.

To use the key you need to create a DesignerUserInfo instance. To create the DesignerUserInfo instance, use the following constructor providing the user ID and the Server Designer License Key or Designer License Key you received when you purchased JReport.

```
DesignerUserInfo userInfo=new DesignerUserInfo(Uid, key);
Designer desg = new Designer(catalogPath, catalogName, userInfo);
```

If you want to use JReport Design API or JReport Server Design API package for workstations without using JReport Designer and JReport Server, you will need to obtain a valid license for each work station.

•

Installing the Design API

The full Design API packages are included in JReport Designer and JReport Server. JReport Designer includes a single threaded Design API, while JReport Server includes a multi-threaded Design API. JReport Design API and JReport Server Design API packages can also be downloaded and installed to your local machine. After which, you can then program with them without having to install any other JReport products.

Installing the Design API by JReport Designer

When you install JReport Designer, the Design API will be installed at the same time. After installation, you will have the components listed below in `<designer_install_root>\lib`:

- JREngine.jar
- report.jar
- sac.jar
- commons-codec-1.2.jar
- log4j-1.2.8.jar (Jar file used for the JReport Engine logging system)

Installing the Design API by JReport Server

When you install JReport Server, the Server Design API will be installed at the same time. After installation, you will have the components listed below in `<server_install_root>\lib`:

- JREngine.jar
- JREServlets.jar
- sac.jar
- commons-codec-1.2.jar
- log4j-1.2.8.jar (Jar file used for the JReport Engine logging system)

Installing independent JReport Design API

Ask the Jinfonet Support (support@jinfonet.com) for the independent Design API installation file. Then, run the file and follow the prompts to install.

After you have installed JReport Design API by running the installation file, you will need to take the following two steps before you compile and run the program:

1. Set the classpath environment variable. Then, append the following string to your class path that starts the Design API:

   ```
   <designerAPI_install_root>\lib\designAPI.jar;<designerAPI_install_root>\lib\log4j-1.2.8.jar;<designerAPI_install_root>\lib\sac.jar;<designerAPI_install_root>\lib\commons-codec-1.2.jar;<designerAPI_install_root>\lib\hsqldb.jar
   ```

2. Set the -Dreporthome parameter equal to the root path of the Design API.

Installing independent JReport Server Design API

Ask the Jinfonet Support (support@jinfonet.com) for the independent Server Design API installation file. Then, run the file and follow the prompts to install. We don't recommend using these libraries, we recommend you use the Design API libraries included in JReport Server.

After you have installed JReport Server Design API by running the installation file, you will need to take the following two steps before you compile and run the program:

1. Set the classpath environment variable. Then, append the following string to your class path that starts the Server Design API:

   ```
   <svrdesignerAPI_install_root>\lib\serverDesignAPI.jar;<svrdesignerAPI_install_root>\lib\log4j-1.2.8.jar;<svrdesignerAPI_install_root>\lib\sac.jar;<svrdesignerAPI_install_root>\commons-codec-1.2.jar;<svrdesignerAPI_install_root>\hsqldb.jar
   ```

2. Set the -Dreporthome parameter to be the same as the root path of the Server Design API.

Note: The Design API classes are stored in the archive files: report.jar and JREngine.jar. You will need to take the following two steps before you can compile and run the program:

1. Set the classpath environment variable. Append the following string to your class path that starts the Design API:

   ```
   <designer_install_root>\lib\report.jar;<designer_install_root>\lib\JREngine.jar;<designer_install_root>\lib\log4j-1.2.8.jar;<designer_install_root>\lib\sac.jar;<designer_install_root>\commons-codec-1.2.jar;
   ```

2. Add the jar files for your DBMS JDBC drivers.
3. Set the -Dreporthome parameter to be the same as the root path of JReport Designer.
4. You will need to register the Design API licensed user and license key in your source code.

Note: The Server Design API classes are stored in the archive files: JRESServlets.jar and JREngine.jar. You will need to take the following two steps before you can compile and run the program:

1. Set the classpath environment variable. Append the following string to your class path that starts the Design API:

   ```
   <server_install_root>\lib\JRESServlets.jar;<server_install_root>\lib\JREngine.jar;<server_install_root>\lib\log4j-1.2.8.jar;<server_install_root>\lib\sac.jar;<server_install_root>\commons-codec-1.2.jar;
   ```

2. Add the jar files for our DBMS JDBC drivers.
3. Set the -Dreporthome parameter to be the same as the root path of JReport Server.
4. You will need to register the Server Design API licensed user and license key in your source code.

•

Getting started

To help you start using the Design API, here is a simple example which shows how to use the JReport Design API package to compile and run a Java class that returns the connection information from a catalog file.

1. Compile the sample code TellMeConnection.java in `<install_root>\help\samples\APICatalog`.

   ```
   javac -classpath <install_root>\lib\JREngine.jar;<install_root>\lib\sac.jar;
   <install_root>\lib\report.jar TellMeConnection.java
   ```

2. Set the license key for Design API using the method `dr.setUserInfo("UID","XXXXXXXXXXX")` in TellMeConnection.java.

3. Run the sample code.

   ```
   java -classpath ".;<install_root>\lib\JREngine.jar;<install_root>\lib\report.jar;
   <install_root>\lib\sac.jar;<install_root>\lib\log4j-1.2.8.jar" -
   Dreporthome=<install_root> TellMeConnection –path=<catalog path> -catalog=<catalog
   name>
   ```

   For example,

   ```
   java -classpath "c:\JReport\Designer\lib\JREngine.jar;c:\JReport\Designer\lib
   \report.jar;c:\JReport\Designer\lib\sac.jar;c:\JReport\Designer\lib\log4j-1.2.8.
   jar;C:\JReport\Designer\lib\hsqldb.jar" -Dreporthome=D:\JReport\Designer
   TellMeConnection -path=C:\JReport\Designer\Demo\Reports\TutorialReports -
   catalog=JinfonetGourmetJava.cat
   ```

   And the output should be as follows:

   Name: Jinfonet demo
   Driver: org.hsqldb.jdbcDriver
   URL: `jdbc:hsqldb:c:\JReport\Designer\Demo\db\JRDemo`
   User: sa
   Password
   TimestampFormat yyyy-MM-dd HH:mm:ss.SSS
   TimeFormat HH:mm:ss
   DateFormat yyyy-MM-dd
   ExtraNamePattern Default(JDBC)

•

Programming with the Design API

The class API is the root of the class hierarchy of JReport Design API (Design API and Catalog API). It is an abstract class and provides a series of editing methods for users. The Design API class has an API as a super class.

Designer and MultiUserDesigner are the principal classes that manipulate reports. The application creates a Designer instance and calls Designer methods to create or modify a report. For details of summary of FIELD, CONSTRUCTOR and METHOD, see JReport Javadoc jet.api package in `<install_root>\help\api`, which includes all documents generated by Javadoc.

The following topics explain how to use the Design API:

- Preparations
- Using the Design API
- Illustrative examples

Preparations

Before you can use the Design API to perform certain tasks, you need to first make the following preparations.

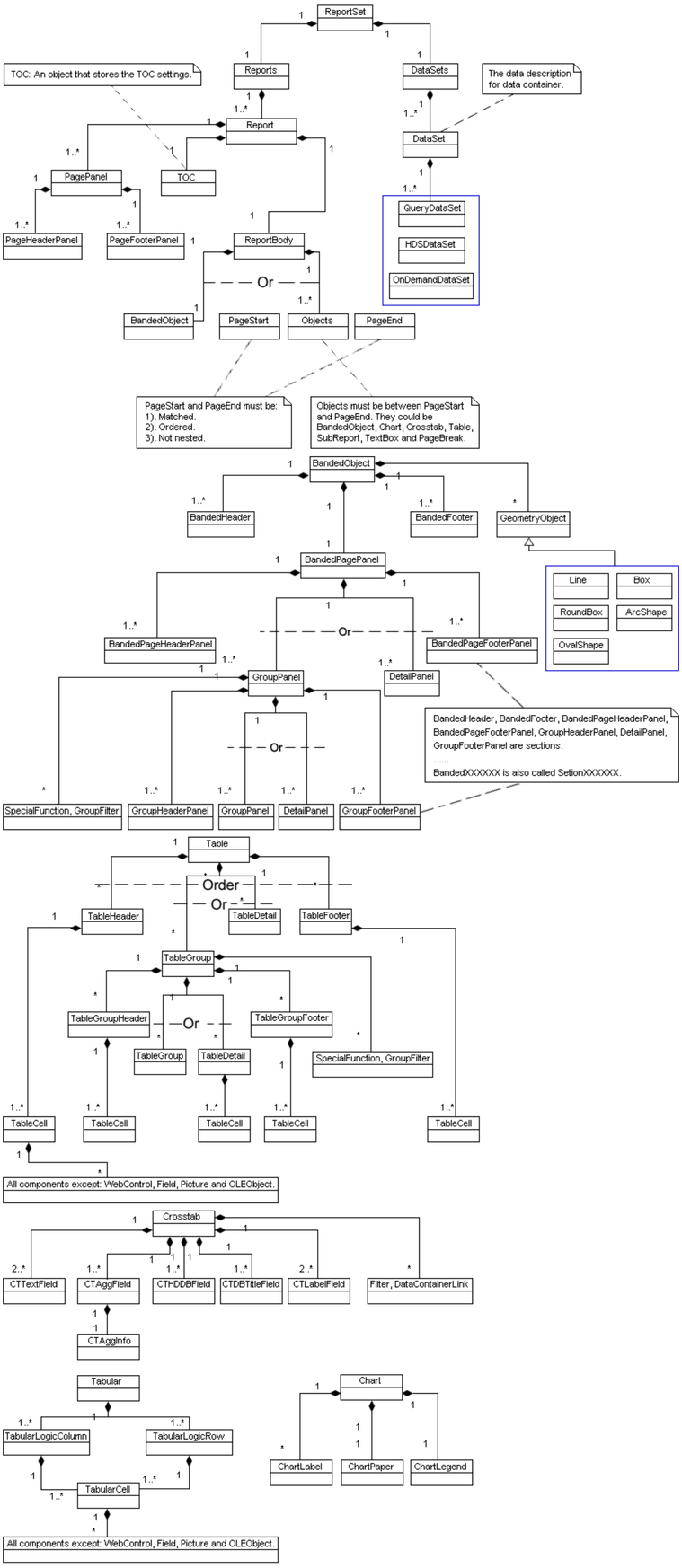
Setting the license key

The license information is set via a method called setUserInfo(String uid, String key), where UID is the user ID, and key is the license key. This method should be called before you can call any other Design API methods. The UID is the license user ID you received with the purchase of JReport Server and JReport Designer. The license key will be either Server Designer License Key or Designer License Key depending on your class path setting to Server\lib or Designer\lib respectively.

```
DesignerUserInfo userInfo=new DesignerUserInfo(Uid, key);
Designer desg = new Designer(catalogPath, catalogName, userInfo);
```

Getting handle information

All the objects in a report are organized into and browsed from a tree structure. For each report, the Report Inspector consists of a report objects tree at the top of the view panel and its corresponding reporting features or properties at the bottom. Selecting any object node on the tree also selects the same element in the report design area. The following diagrams cover all objects that can be used in a report with their relationships.



Every node and object in the above trees is identified by a HANDLE in JReport Design API. Therefore, before you can create or edit a report with the Design API, you will first have to get the handle information. Super class API provides the following methods to get handles of different nodes and different types.

- **getHandles()**
  Gets all the handles of a report and returns a handle array.
- **getHandles(String handle)**
  Gets handles of the sub node in the current node and returns a handle array. If it fails, it will return null.
- **getHandles(String handle, int type)**
  Gets sub node handles of the same types for the given node and returns a handle array. If it fails, it will return null.
- **getHandles(String handle, int type, int depth)**
  Gets sub node handles of the same types for the given node and returns a handle array. The depth is described below. If it fails, it will return null.
- **getParent(String handle)**
  Gets the parent handle of an object and returns the parent handle. If it fails, it will return null.

Parameters

- handle - Parent node handle when getting sub node handles. Object handle when getting parent handle.
- type - Class type value defined in the API class.
- depth - This is related to the current level in the report tree. When the depth is -1, all handles in the report will be returned. When the depth is 0, the handles of the current level will be returned. When the depth is number n, the handles of n levels and the current level will be returned.

Using the Design API

After making the required preparations, you can now use the Design API to perform tasks.

The following topics detail how to program with the Design API to achieve certain goal:

- Creating and deleting a report
- Opening and closing a report
- Inserting and deleting an object
- Getting property information
- Getting report metadata information
- Getting a Catalog API instance
- Getting objects and object information
- Getting the instance name and class type of an object
- Moving objects
- Getting editing information
- Exiting from the editing status
- Concurrent control

•

Creating and deleting a report

- **create(String name, String query)**
  Creates a new report with a specified name, and returns the handle of the new report. If it fails, a null value will be returned.
- **createReportSet(String name)**
  Creates a new report with a specified name.
- **deleteReport(java.lang.String name)**
  Deletes an existing report with the specified name, and returns true if the report has been successfully removed.

Parameters

- name - The specified report name that is to be created or deleted.
- query - The query name on which the report is created.

Notes:

- The report should be created based on the existing catalog file, otherwise you will get an error.
- The query should exist in the given catalog, otherwise you will get an error.

•

Opening and closing a report

To open or close a report, you can use the following methods:

- **open(String name)**
  Opens an existing report and returns the handle of the report.
- **close(String handle)**
  Saves any changes and closes the open report tab in the current report.
- **closeReportSet()**
  Saves any changes and closes the current report.
- **quit(String handle)**
  Closes an open report without saving any changes.

Parameters

- name - Name of the report that is to be opened.
- handle - Ancestor handle of the open report.

•

Inserting and deleting an object

To insert or delete an object into a report, you can use the methods and parameters below:

- **insert(String parent, int type, String name)**
  Inserts an object into the parent node.
- **insert(String parent, int type, String name, String mapping)**
  Inserts a DBField, parameter, formula or summary into the parent node.
- **insert(String parent, String name, CTRowColFieldInfo colInfo[], CTRowColFieldInfo rowInfo[], CTAggFieldInfo aggInfo[])**
  Insert a crosstab object and its children into the parent node.
- **insert(String parent, CTRowColFieldInfo colInfo, CTRowColFieldInfo rowInfo, CTAggFieldInfo aggInfo)**
  Inserts crosstab children into the crosstab.
- **insert(String parent, String name, String paperName, int charttype, String gName1, String gName2, String value, ChartLegendInfo chartLegendInfo, ChartLabelInfo chartLabelInfo)**
  Inserts a chart object and its children into the parent node.
- **insert (String parent, int type, String name, String topSection , String bottomSection)**
  Inserts a shape object into the report and returns the handle of the new shape.
- **insert(String parent, String name, GroupInfo groupInfo)**
  Inserts a group/sort into the parent node.
- **delete(String handle)**
  Deletes an object from its parent node.

Parameters

- parent - Parent handle node.
- type - Class type that you want to insert.
- name - Instance name of the new object.
- mapping - Mapping name of the database field, parameter, formula or summary.
- colInfo - Field information array of the Column.
- rowInfo - Field information array of the Row.
- aggInfo - Field information array of the Aggregate.
- paperName - Instance name of ChartCoordinatepaper object.
- charttype - Type of Chart.
- group1 - Mapping name of the first group.
- group2- Mapping name of the second group. It can be null.
- value - Mapping name of a summary.
- chartLegendInfo - Field information array of the chart legend.
- chartLabelInfo - Field information array of the chart title and notes.
- handle - Handle of the object that is to be deleted.
- topSection - Handle of the top section attached by the shape.
- bottomSection - Handle of the bottom section attached by the shape.
- groupInfo - The group information.

G-Z

Querying property information

Querying property information

The following functions are used to query the opening property values of different types and store the corresponding values.

- gmcopenproperty_boolean, String name
- gmcopenproperty_byte, String name
- gmcopenproperty_char, String name
- gmcopenproperty_double, String name
- gmcopenproperty_float, String name
- gmcopenproperty_int, String name
- gmcopenproperty_long, String name
- gmcopenproperty_short, String name
- gmcopenproperty_string, String name
- gmcopenproperty_vector3d, String name

Returns: The value of the property.

- scalar - Return Value
- vector - Return Value

Note: The gmcopenproperty_boolean, String name() function is used to get a boolean field value from the opening property, and the other functions are similar.

- gmcsetproperty_boolean, bool, String name
- gmcsetproperty_byte, byte, String name
- gmcsetproperty_char, char, String name
- gmcsetproperty_double, double, String name
- gmcsetproperty_float, float, String name
- gmcsetproperty_int, int, String name
- gmcsetproperty_long, long, String name
- gmcsetproperty_short, short, String name
- gmcsetproperty_string, string, String name
- gmcsetproperty_vector3d, vector3d, String name

Note: The gmcsetproperty_boolean, bool, String name() function is used to set a boolean field value for the opening property, and the other functions are similar.

Querying Vehicle Information

Querying Vehicle Information

The following functions are used to query Vehicle Property Values of various types.

- scalar - Return Value
- vector - Return Value

Note: The gmcgetvehicleproperty_boolean, int, int, int() function is used to get a boolean field value of a vehicle, and the other functions are similar.

- gmcvehicleproperty_boolean, int, int, int, bool
- gmcvehicleproperty_byte, int, int, int, byte
- gmcvehicleproperty_char, int, int, int, char
- gmcvehicleproperty_double, int, int, int, double
- gmcvehicleproperty_float, int, int, int, float
- gmcvehicleproperty_int, int, int, int, int
- gmcvehicleproperty_long, int, int, int, long
- gmcvehicleproperty_short, int, int, int, short
- gmcvehicleproperty_string, int, int, int, string
- gmcvehicleproperty_vector3d, int, int, int, vector3d

Parameters:

- int - Target object handle
- int - Vehicle Type
- int - Vehicle Attribute
- bool/byte/char/double/float/int/long/short/string/vector3d - Attribute value

Note: The gmcsetvehicleproperty_boolean, int, int, int, bool() function is used to set a boolean field value of a vehicle, and the other functions are similar.

Functions

Functions

The following functions are used to obtain the corresponding values in different types. They will return a null if the object is invalid.

- gmcgetactiveplayerhandle, int, int, int, int

	Kindergartenzone	Kindergarten is a school or school-like institution for children aged 3 to 6 years. Kindergarten is a school or school-like institution for children aged 3 to 6 years. Data type: Boolean
	Kinder	Kinder are children. Data type: Boolean
	Kinderbetreuungsangebot	Kinderbetreuungsangebot is a place where children are cared for, for example, in a kindergarten, nursery school, or day care center. Data type: Boolean
	Kinderbetreuungsangebotzone	Kinderbetreuungsangebotzone is a place where children are cared for, for example, in a kindergarten, nursery school, or day care center. Data type: Boolean
	Kinderbetreuung	Kinderbetreuung is the care of children by someone other than their parents or guardians, typically in a kindergarten, nursery school, or day care center. Data type: Boolean
	Kinderbetreuungszone	Kinderbetreuungszone is the care of children by someone other than their parents or guardians, typically in a kindergarten, nursery school, or day care center. Data type: Boolean
	Kinderbetreuungsstätte	Kinderbetreuungsstätte is a place where children are cared for, for example, in a kindergarten, nursery school, or day care center. Data type: Boolean
	Kinderbetreuungsstättezone	Kinderbetreuungsstättezone is a place where children are cared for, for example, in a kindergarten, nursery school, or day care center. Data type: Boolean
	Kinderbetreuungszentrum	Kinderbetreuungszentrum is a place where children are cared for, for example, in a kindergarten, nursery school, or day care center. Data type: Boolean
	Kinderbetreuungszentrumzone	Kinderbetreuungszentrumzone is a place where children are cared for, for example, in a kindergarten, nursery school, or day care center. Data type: Boolean
	Kindergarten	Kindergarten is a school or school-like institution for children aged 3 to 6 years. Data type: Boolean
	Kindergartenzone	Kindergartenzone is a school or school-like institution for children aged 3 to 6 years. Data type: Boolean
	Kindergeld	Kindergeld is a monthly payment to parents or guardians of children under 18 years of age, regardless of their income or employment status. Data type: Boolean
	Kindergeldzone	Kindergeldzone is a monthly payment to parents or guardians of children under 18 years of age, regardless of their income or employment status. Data type: Boolean
	Kindergrenze	Kindergrenze is the maximum age at which children are considered children for tax purposes. Data type: Boolean
	Kindergrenzezone	Kindergrenzezone

Ancestor		Ancestor
Ancestor		Ancestor
Ancestor		Ancestor

Rain-Gauge		Gauge for measuring the quantity of rain falling in a given time. Syn. Pluviometer. Ant. Anemometer. Adj. Rain-gauge.
Rain-gauge		Adj. Rain-gauge. Gauge for measuring the quantity of rain falling in a given time. Syn. Pluviometer. Ant. Anemometer.
Rain-gauging		Noun. Verbal. Measuring the quantity of rain falling in a given time.
Rainbow		Noun. Concrete. Arch of many colours, formed by the refraction of the sun's rays in a shower. Adj. Rainbow.
Rainbow-coloured		Adj. Rainbow. Arch of many colours, formed by the refraction of the sun's rays in a shower.
Rainbow-tinted		Adj. Rainbow. Arch of many colours, formed by the refraction of the sun's rays in a shower.
Raindrop		Noun. Concrete. Drop of water, formed by the condensation of the vapour in the atmosphere. Syn. Rain-drop.
Rainfall		Noun. Concrete. Quantity of water falling in a given time. Syn. Rain-fall

	WordWrap	Specifies whether to wrap the text to the width. Data type: Boolean
Subreport	AnchorDispValue	Specifies a string or formula to display for the TOC entry for this object in the TOC Browser. The TOCAnchor property must be set to true for this property to take effect. Data type: String
	Background	Specifies the background color of the object. Data type: String
	BorderColor	Specifies the color of the border of the object. Data type: String
	BorderWidth	Specifies the width of the border in inches or centimeters. Data type: Float
	BottomLine	Specifies the line style of the bottom border of the object. Data type: Enumeration
	Cache	Specifies whether to cache the dataset for this object in the data buffer so other objects which use the same dataset can share the data rather than perform their own SQL query. Data type: Boolean
	class	Specifies a CSS class to be applied to the object which is a valid class in the CSS file. Data type: String
	ColumnIndex	Specifies the X coordinate of the object relative to its parent container when exported to Excel or CSV, measured in cells. The Columned property at the report tab level must be set to true for this property to take effect. Data type: Integer
	Embedded	If false, the page header and footer of the subreport will not be shown. Data type: Boolean
	ExportToCSV	Specifies whether to include the object when exporting the report to CSV. Data type: Boolean
	ExportToDHTML	Specifies whether to include the object when exporting the report to Page Report Result. Data type: Boolean
	ExportToXLS	Specifies whether to include the object when exporting the report to Excel. Data type: Boolean
	ExternalCSSClassValue	Specifies a class selector to be applied to the object when exported as HTML. Data type: String
	ExternalID	It is mapped to the HTML attribute *id*. This attribute specifies a name to the object, which must be unique in the report. Data type: String
	ExternalStyle	It is mapped to the HTML attribute *style*. This attribute specifies style information for the object. Data type: String
	Height	Specifies the height of the object, in inches or centimeters. Data type: Float
	Invisible	Specifies whether to show the object in the design area and in the report result. All formulas and calculations will still be performed if the property is set to true. Data type: Boolean
	Lang	It is mapped to the HTML attribute *lang*. This attribute specifies the base language of the object's attribute values and text content. Data type: String
	LeftLine	Specifies the line style of the left border of the object. Data type: Enumeration
	MaxPageNum	Specifies the maximum number of pages in the data buffer. Data type: Integer
	OnNewSheet	Specifies whether to put the subreport in a new sheet when exporting the report to Excel. Data type: Boolean
	position	Specifies the position of the object. • **static** - The object is positioned at the location in which it is inserted. The X, Y and other position-related properties will be hidden or disabled. The position of the object is affected by preceding objects in the same container. • **absolute** - The object will be located at the position specified by dragging and dropping or by setting its X and Y property values. The position of the object is not affected by preceding objects in the same container. Data type: String **Note:** This property only affects objects with a parent container that is the report body, a tabular cell, or a text box.
	Postfix	Specifies how to add postfix number to the sheets in which the subreport will be put when exporting the report to Excel. • **Add postfix number to every page** - Specifies to add the postfix number to every sheet for the subreport (Subreport1, Subreport2, Subreport3, Subreport4, ...). • **Add postfix number only to additional pages** - Specifies to add the postfix number to the subreport sheets starting from the second one (Subreport, Subreport1, Subreport2, Subreport3, ...). It is meaningful to specify the property value when the OnNewSheet property is set to true. Data type: String
	RecPerPage	Specifies the number of records in each page in the data buffer. Data type: Integer
	RecordLocation	Specifies the calculation point for the properties which use formulas. • **Default** - The values of the properties are calculated in the default location where they are located. • **Page Header** - The values of the properties are calculated in the page header. • **Page Footer** - The values of the properties are calculated in the page footer. Data type: String
	RightLine	Specifies the line style of the right border of the object. Data type: Enumeration
	RowIndex	Specifies the Y coordinate of the object relative to its parent container when exported to Excel or CSV, measured in cells. The Columned property at the report tab level must be set to true for this property to take effect. Data type: Integer
	SheetName	Specifies the name of the new sheet in which the subreport will be put. Make sure the name specified here isn't used by any other sheets. It is meaningful to specify the property value when the OnNewSheet property is set to true. Data type: String **Note:** The characters "|", ":", "/", "?", "\", "*", "]", "[" cannot be displayed in the sheet name in Excel and the last character in the sheet name cannot be single quotation mark ('). If any one of them is used when you specify the property, it will be replaced by "_" in Excel.
	SRQueryName	Displays the name of the query used by the subreport. This property is read only.
	SRSecurity	Displays the security name (record level security) used by the subreport. This property is read only.
	SRSourceName	Displays the name of the data source used by the subreport. If nothing is displayed, the default data source will be used. This property is read only.
	Style	The property can be used in two ways. • Specifies a style to be applied to the object. This works when a StyleGroup has been specified at the report sheet level and when there are styles in the StyleGroup that can be applied to the object. • Specifies a CSS selector to be applied to the object. Data type: String
	Suppressed	Specifies whether to show the object in the design area and in the report result. All formulas and calculations will be skipped if the property is set to true. Data type: Boolean **Note:** When both the Invisible and Suppressed properties of an object are set to true, Suppressed has the higher priority.
	SuppressedIfNoRecords	Specifies whether to display the object in the report result when no record is returned to its parent data container. Data type: Boolean
	SuppressEmpty	Specifies whether to display the object in the report result when no record is returned to it. Data type: Boolean
	TOCAnchor	Specifies whether to include the object in the TOC Browser for the report. Data type: Boolean
	TopLine	Specifies the line style of the top border of the object. Data type: Enumeration
	Width	Specifies the width of the object, in inches or centimeters. Data type: Float
	X	Specifies the horizontal coordinate of the top left corner of the object, relative to its parent container. This property is ignored if the position property is set to static. Data type: Float
	Y	Specifies the vertical coordinate of the top left corner of the object, relative to its parent container. This property is ignored if the position property is set to static. Data type: Float
Line	BottomAttachCol	The four Excel properties, TopAttachCol, TopAttachRow, BottomAttachCol and BottomAttachRow, work together to control the position of a drawing object in the exported Excel file. BottomAttachCol specifies the X coordinate of the lower right corner of the drawing object in the exported Excel file, measured in cells. The Columned property at the report tab level must be set to true for this property to take effect. Data type: Float
	BottomAttachPosX	Specifies the horizontal coordinate of the bottom right point of the line in a banded panel. A drawing object can cross panels and the coordinate indicates the relative position in the involved panel. Data type: Float
	BottomAttachPosY	Specifies the vertical coordinate of the bottom right point of the line in a banded panel. A drawing object can cross panels and the coordinate indicates the relative position in the involved panel. Data type: Float
	BottomAttachRow	The four Excel properties, TopAttachCol, TopAttachRow, BottomAttachCol and BottomAttachRow, work together to control the position of a drawing object in the exported Excel file. BottomAttachRow specifies the Y coordinate of the lower right corner of the drawing object in the exported Excel file, measured in cells. The Columned property at the report tab level must be set to true for this property to take effect. Data type: Float
	class	Specifies a CSS class to be applied to the object which is a valid class in the CSS file. Data type: String
	ExportToDHTML	Specifies whether to include the object when exporting the report to Page Report Result. Data type: Boolean
	ExportToXLS	Specifies whether to include the object when exporting the report to Excel. Data type: Boolean
	ExternalCSSClassValue	Specifies a class selector to be applied to the object when exported as HTML. Data type: String
	ExternalID	It is mapped to the HTML attribute *id*. This attribute specifies a name to the object, which must be unique in the report. Data type: String
	ExternalStyle	It is mapped to the HTML attribute *style*. This attribute specifies style information for the object. Data type: String
	ExternalTitle	It is mapped to the HTML attribute *title*. This attribute offers advisory information about the object. Data type: String
	Invisible	Specifies whether to show the object in the design area and in the report result. All formulas and calculations will still be performed. Data type: Boolean
	Lang	It is mapped to the HTML attribute *lang*. This attribute specifies the base language of the object's attribute values and text content. Data type: String
	LineColor	Specifies the color of the line. Data type: String
	LineStyle	Specifies the style of the line. Data type: Enumeration
	LineWidth	Specifies the thickness of the line. Data type: Float
	RecordLocation	Specifies the calculation point for the properties which use formulas. • **Default** - The values of the properties are calculated in the default location where they are located. • **Page Header** - The values of the properties are calculated in the page header. • **Page Footer** - The values of the properties are calculated in the page footer. Data type: String
	Style	The property can be used in two ways. • Specifies a style to be applied to the object. This works when a StyleGroup has been specified at the report sheet level and when there are styles in the StyleGroup that can be applied to the object. • Specifies a CSS selector to be applied to the object. Data type: String
	SuppressedIfNoRecords	Specifies whether to display the object in the report result when no record is returned to its parent data container. Data type: Boolean
	TopAttachCol	The four Excel properties,TopAttachCol, TopAttachRow, BottomAttachCol and BottomAttachRow, work together to control the position of a drawing object in the exported Excel file. TopAttachCol specifies the X coordinate of the upper left corner of the drawing object in the exported Excel file, measured in cells. The Columned property at the report tab level must be set to true for this property to take effect. Data type: Float
	TopAttachPosX	Specifies the horizontal coordinate of the top left point of the line in a banded panel. A drawing object can cross panels and the coordinate indicates the relative position in the involved panel. Data type: Float
	TopAttachPosY	Specifies the vertical coordinate of the top left point of the line in a banded panel. A drawing object can cross panels and the coordinate indicates the relative position in the involved panel. Data type: Float
	TopAttachRow	The four Excel properties, TopAttachCol, TopAttachRow, BottomAttachCol and BottomAttachRow, work together to control the position of a drawing object in the exported Excel file. TopAttachRow specifies the Y coordinate of the upper left corner of the drawing object in the exported Excel file, measured in cells. The Columned property at the report tab level must be set to true for this property to take effect. Data type: Float
Box	Background	Specifies the background color of the object. Data type: String
	BorderColor	Specifies the color of the border of the object. Data type: String
	BorderStyle	Specifies the line style of the box border. Data type: Enumeration
	BorderWidth	Specifies the width of the object's border, in inches or centimeters. Data type: Float
	BottomAttachCol	The four Excel properties, TopAttachCol, TopAttachRow, BottomAttachCol and BottomAttachRow, work together to control the position of a drawing object in the exported Excel file. BottomAttachCol specifies the X coordinate of the lower right corner of the drawing object in the exported Excel file, measured in cells. The Columned property at the report tab level must be set to true for this property to take effect. Data type: Float
	BottomAttachPosX	Specifies the horizontal coordinate of the bottom right point of the box in a banded panel. A drawing object can cross panels and the coordinate indicates the relative position in the involved panel. Data type: Float
	BottomAttachPosY	Specifies the vertical coordinate of the bottom right point of the box in a banded panel. A drawing object can cross panels and the coordinate indicates the relative position in the involved panel. Data type: Float
	BottomAttachRow	The four Excel properties, TopAttachCol, TopAttachRow, BottomAttachCol and BottomAttachRow, work together to control the position of a drawing object in the exported Excel file. BottomAttachRow specifies the Y coordinate of the lower right corner of the drawing object in the exported Excel file, measured in cells. The Columned property at the report tab level must be set to true for this property to take effect. Data type: Float
	class	Specifies a CSS class to be applied to the object which is a valid class in the CSS file. Data type: String
	ExportToDHTML	Specifies whether to include the object when exporting the report to Page Report Result. Data type: Boolean
	ExportToXLS	Specifies whether to include the object when exporting the report to Excel. Data type: Boolean
	ExternalCSSClassValue	Specifies a class selector to be applied to the object when exported as HTML. Data type: String
	ExternalID	It is mapped to the HTML attribute *id*. This attribute specifies a name to the object, which must be unique in the report. Data type: String
	ExternalStyle	It is mapped to the HTML attribute *style*. This attribute specifies style information for the object. Data type: String
	ExternalTitle	It is mapped to the HTML attribute *title*. This attribute offers advisory information about the object. Data type: String
	Invisible	Specifies whether to show the object in the design area and in the report result. All formulas and calculations will still be performed. Data type: Boolean
	Lang	It is mapped to the HTML attribute *lang*. This attribute specifies the base language of the object's attribute values and text content. Data type: String
	RecordLocation	Specifies the calculation point for the properties which use formulas. • **Default** - The values of the properties are calculated in the default location where they are located. • **Page Header** - The values of the properties are calculated in the page header. • **Page Footer** - The values of the properties are calculated in the page footer. Data type: String
	Style	The property can be used in two ways. • Specifies a style to be applied to the object. This works when a StyleGroup has been specified at the report sheet level and when there are styles in the StyleGroup that can be applied to the object. • Specifies a CSS selector to be applied to the object. Data type: String
	SuppressedIfNoRecords	Specifies whether to display the object in the report result when no record is returned to its parent data container. Data type: Boolean
	TopAttachCol	The four Excel properties, TopAttachCol, TopAttachRow, BottomAttachCol and BottomAttachRow, work together to control the position of a drawing object in the exported Excel file. TopAttachCol specifies the X coordinate of the upper left corner of the drawing object in the exported Excel file, measured in cells. The Columned property at the report tab level must be set to true for this property to take effect. Data type: Float
	TopAttachPosX	Specifies the horizontal coordinate of the top left point of the box in a banded panel. A drawing object can cross panels and the coordinate indicates the relative position in the involved panel. Data type: Float
	TopAttachPosY	Specifies the vertical coordinate of the top left point of the box in a banded panel. A drawing object can cross panels and the coordinate indicates the relative position in the involved panel. Data type: Float
	TopAttachRow	The four Excel properties, TopAttachCol, TopAttachRow, BottomAttachCol and BottomAttachRow, work together to control the position of a drawing object in the exported Excel file. TopAttachRow specifies the Y coordinate of the upper left corner of the drawing object in the exported Excel file, measured in cells. The Columned property at the report tab level must be set to true for this property to take effect. Data type: Float
Round box	Background	Specifies the background color of the object. Data type: String
	BorderColor	Specifies the color of the object's border. Data type: String
	BorderStyle	Specifies the line style of the object's border. Data type: Enumeration
	BorderWidth	Specifies the width of the object's border, in inches or centimeters. Data type: Float
	BottomAttachCol	The four Excel properties, TopAttachCol, TopAttachRow, BottomAttachCol and BottomAttachRow, work together to control the position of a drawing object in the exported Excel file. BottomAttachCol specifies the X coordinate of the lower right corner of the drawing object in the exported Excel file, measured in cells. The Columned property at the report tab level must be set to true for this property to take effect. Data type: Float
	BottomAttachPosX	Specifies the horizontal coordinate of the bottom right point of the focus box in a banded panel. A drawing object can cross panels and the coordinate indicates the relative position in the involved panel. Data type: Float
	BottomAttachPosY	Specifies the vertical coordinate of the bottom right point of the focus box in a banded panel. A drawing object can cross panels and the coordinate indicates the relative position in the involved panel. Data type: Float
	BottomAttachRow	The four Excel properties, TopAttachCol, TopAttachRow, BottomAttachCol and BottomAttachRow, work together to control the position of a drawing object in the exported Excel file. BottomAttachRow specifies the Y coordinate of the lower right corner of the drawing object in the exported Excel file, measured in cells. The Columned property at the report tab level must be set to true for this property to take effect. Data type: Float
	class	Specifies a CSS class to be applied to the object which is a valid class in the CSS file. Data type: String
	CornerFactor	Specifies the relative radius of the object's corners. Data type: Float
	ExportToDHTML	Specifies whether to include the object when exporting the report to Page Report Result. Data type: Boolean
	ExportToXLS	Specifies whether to include the object when exporting the report to Excel. Data type: Boolean
	ExternalCSSClassValue	Specifies a class selector to be applied to the object when exported as HTML. Data type: String
	ExternalID	It is mapped to the HTML attribute *id*. This attribute specifies a name to the object, which must be unique in the report. Data type: String
	ExternalStyle	It is mapped to the HTML attribute *style*. This attribute specifies style information for the object. Data type: String
	ExternalTitle	It is mapped to the HTML attribute *title*. This attribute offers advisory information about the object. Data type: String
	Invisible	Specifies whether to show the object in the design area and in the report result. All formulas and calculations will still be performed. Data type: Boolean
	Lang	It is mapped to the HTML attribute *lang*. This attribute specifies the base language of the object's attribute values and text content. Data type: String
	RecordLocation	Specifies the calculation point for the properties which use formulas. • **Default** - The values of the properties are calculated in the default location where they are located. • **Page Header** - The values of the properties are calculated in the page header. • **Page Footer** - The values of the properties are calculated in the page footer. Data type: String
	Style	The property can be used in two ways. • Specifies a style to be applied to the object. This works when a StyleGroup has been specified at the report sheet level and when there are styles in the StyleGroup that can be applied to the object. • Specifies a CSS selector to be applied to the object. Data type: String
	SuppressedIfNoRecords	Specifies whether to display the object in the report result when no record is returned to its parent data container. Data type: Boolean
	TopAttachCol	The four Excel properties, TopAttachCol, TopAttachRow, BottomAttachCol and BottomAttachRow, work together to control the position of a drawing object in the exported Excel file. TopAttachCol specifies the X coordinate of the upper left corner of the drawing object in the exported Excel file, measured in cells. The Columned property at the report tab level must be set to true for this property to take effect. Data type: Float
	TopAttachPosX	Specifies the horizontal coordinate of the top left point of the focus box in a banded panel. A drawing object can cross panels and the coordinate indicates the relative position in the involved panel. Data type: Float
	TopAttachPosY	Specifies the vertical coordinate of the top left point of the focus box in a banded panel. A drawing object can cross panels and the coordinate indicates the relative position in the involved panel. Data type: Float
	TopAttachRow	The four Excel properties, TopAttachCol, TopAttachRow, BottomAttachCol and BottomAttachRow, work together to control the position of a drawing object in the exported Excel file. TopAttachRow specifies the Y coordinate of the upper left corner of the drawing object in the exported Excel file, measured in cells. The Columned property at the report tab level must be set to true for this property to take effect. Data type: Float
Oval	Background	Specifies the background color of the object. Data type: String
	BorderColor	Specifies the color of the object's border. Data type: String
	BorderStyle	Specifies the line style of the object's border. Data type: Enumeration
	BorderWidth	Specifies the width of the object's border, in inches or centimeters. Data type: Float
	BottomAttachCol	The four Excel properties, TopAttachCol, TopAttachRow, BottomAttachCol and BottomAttachRow, work together to control the position of a drawing object in the exported Excel file. BottomAttachCol specifies the X coordinate of the lower right corner of the drawing object in the exported Excel file, measured in cells. The Columned property at the report tab level must be set to true for this property to take effect. Data type: Float
	BottomAttachPosX	Specifies the horizontal coordinate of the bottom right point of the focus box in a banded panel. A drawing object can cross panels and the coordinate indicates the relative position in the involved panel. Data type: Float
	BottomAttachPosY	Specifies the vertical coordinate of the bottom right point of the focus box in a banded panel. A drawing object can cross panels and the coordinate indicates the relative position in the involved panel. Data type: Float
	BottomAttachRow	The four Excel properties, TopAttachCol, TopAttachRow, BottomAttachCol and BottomAttachRow, work together to control the position of a drawing object in the exported Excel file. BottomAttachRow specifies the Y coordinate of the lower right corner of the drawing object in the exported Excel file, measured in cells. The Columned property at the report tab level must be set to true for this property to take effect. Data type: Float
	class	Specifies a CSS class to be applied to the object which is a valid class in the CSS file. Data type: String
	ExportToDHTML	Specifies whether to include the object when exporting the report to Page Report Result. Data type: Boolean
	ExportToXLS	Specifies whether to include the object when exporting the report to Excel. Data type: Boolean
	ExternalCSSClassValue	Specifies a class selector to be applied to the object when exported as HTML. Data type: String
	ExternalID	It is mapped to the HTML attribute *id*. This attribute specifies a name to the object, which must be unique in the report. Data type: String
	ExternalStyle	It is mapped to the HTML attribute *style*. This attribute specifies style information for the object. Data type: String
	ExternalTitle	It is mapped to the HTML attribute *title*. This attribute offers advisory information about the object. Data type: String
	Lang	It is mapped to the HTML attribute *lang*. This attribute specifies the base language of the object's attribute values and text content. Data type: String
	Invisible	Specifies whether to show the object in the design area and in the report result. All formulas and calculations will still be performed. Data type: Boolean
	RecordLocation	Specifies the calculation point for the properties which use formulas. • **Default** - The values of the properties are calculated in the default location where they are located. • **Page Header** - The values of the properties are calculated in the page header. • **Page Footer** - The values of the properties are calculated in the page footer. Data type: String
	Style	The property can be used in two ways. • Specifies a style to be applied to the object. This works when a StyleGroup has been specified at the report sheet level and when there are styles in the StyleGroup that can be applied to the object. • Specifies a CSS selector to be applied to the object. Data type: String
	SuppressedIfNoRecords	Specifies whether to display the object in the report result when no record is returned to its parent data container. Data type: Boolean
	TopAttachCol	The four Excel properties, TopAttachCol, TopAttachRow, BottomAttachCol and BottomAttachRow, work together to control the position of a drawing object in the exported Excel file. TopAttachCol specifies the X coordinate of the upper left corner of the drawing object in the exported Excel file, measured in cells. The Columned property at the report tab level must be set to true for this property to take effect. Data type: Float
	TopAttachPosX	Specifies the horizontal coordinate of the top left point of the focus box in a banded panel. A drawing object can cross panels and the coordinate indicates the relative position in the involved panel. Data type: Float
	TopAttachPosY	Specifies the vertical coordinate of the top left point of the focus box in a banded panel. A drawing object can cross panels and the coordinate indicates the relative position in the involved panel. Data type: Float
	TopAttachRow	The four Excel properties, TopAttachCol, TopAttachRow, BottomAttachCol and BottomAttachRow, work together to control the position of a drawing object in the exported Excel file. TopAttachRow specifies the Y coordinate of the upper left corner of the drawing object in the exported Excel file, measured in cells. The Columned property at the report tab level must be set to true for this property to take effect. Data type: Float
Arc	ArcAngle	Specifies the angle of the arc, in degrees. The value ranges from 0 to 360. As it gets closer to 360, the arc begins to take on an elliptical shape. Data type: Float
	Background	Specifies the background color of the object. Data type: String
	BorderColor	Specifies the color of the object's border. Data type: String
	BorderStyle	Specifies the line style of the object's border. Data type: Enumeration
	BorderWidth	Specifies the width of the object's border, in inches or centimeters. Data type: Float
	BottomAttachPosX	Specifies the horizontal coordinate of the bottom right point of the focus box in a banded panel. A drawing object can cross panels and the coordinate indicates the relative position in the involved panel. Data type: Float
	BottomAttachPosY	Specifies the vertical coordinate of the bottom right point of the focus box in a banded panel. A drawing object can cross panels and the coordinate indicates the relative position in the involved panel. Data type: Float
	class	Specifies a CSS class to be applied to the object which is a valid class in the CSS file. Data type: String
	ExportToDHTML	Specifies whether to include the object when exporting the report to Page Report Result. Data type: Boolean
	ExternalCSSClassValue	Specifies a class selector to be applied to the object when exported as HTML. Data type: String
	ExternalID	It is mapped to the HTML attribute *id*. This attribute specifies a name to the object, which must be unique in the report. Data type: String
	ExternalStyle	It is mapped to the HTML attribute *style*. This attribute specifies style information for the object. Data type: String
	ExternalTitle	It is mapped to the HTML attribute *title*. This attribute offers advisory information about the object. Data type: String
	Invisible	Specifies whether to show the object in the design area and in the report result. All formulas and calculations will still be performed. Data type: Boolean
	Lang	It is mapped to the HTML attribute *lang*. This attribute specifies the base language of the object's attribute values and text content. Data type: String
	RecordLocation	Specifies the calculation point for the properties which use formulas. • **Default** - The values of the properties are calculated in the default location where they are located. • **Page Header** - The values of the properties are calculated in the page header. • **Page Footer** - The values of the properties are calculated in the page footer. Data type: String
	StartAngle	Specifies the start angle of the arc, in degrees. The value ranges from 0 to 360. At 0, the vertex of the arc starts at 12:15 clock position. As it gets closer to 360, the arc travels in a counterclockwise motion. Data type: Float
	Style	The property can be used in two ways. • Specifies a style to be applied to the object. This works when a StyleGroup has been specified at the report sheet level and when there are styles in the StyleGroup that can be applied to the object. • Specifies a CSS selector to be applied to the object. Data type: String
	SuppressedIfNoRecords	Specifies whether to display the object in the report result when no record is returned to its parent data container. Data type: Boolean
	TopAttachPosX	Specifies the horizontal coordinate of the top left point of the focus box in a banded panel. A drawing object can cross panels and the coordinate indicates the relative position in the involved panel. Data type: Float
	TopAttachPosY	Specifies the vertical coordinate of the top left point of the focus box in a banded panel. A drawing object can cross panels and the coordinate indicates the relative position in the involved panel. Data type: Float
Table	AnchorDispValue	Specifies a string or formula to display for the TOC entry for this object in the TOC Browser. The TOCAnchor property must be set to true for this property to take effect. Data type: String
	AutoExpands	Specifies whether the objects in cells of the table will expand to their empty neighbouring cells horizontally when the actual size of the objects is bigger than the cells due to no enough column or page width for auto fit. This property requires that the property AutoFit for the objects in the table cells is set to true. Data type: Boolean **Note:** The property does not apply to horizontal tables.
	Background	Specifies the background color of the object. Data type: String
	BorderColor	Specifies the color of the border of the object. Data type: String
	BottomLine	Specifies the line style of the bottom border of the object. Data type: Enumeration
	BorderWidth	Specifies the width of the border in inches or centimeters. Data type: Float
	Cache	Specifies whether to cache the dataset for this object in the data buffer so other objects which use the same dataset can share the data rather than perform their own SQL query. Data type: Boolean
	class	Specifies a CSS class to be applied to the object which is a valid class in the CSS file. Data type: String
	ExportToCSV	Specifies whether to include the object when exporting the report to CSV. Data type: Boolean
	ExportToDHTML	Specifies whether to include the object when exporting the report to Page Report Result. Data type: Boolean
	ExportToXLS	Specifies whether to include the object when exporting the report to Excel. Data type: Boolean
	ExternalCSSClassValue	Specifies a class selector to be applied to the object when exported as HTML. Data type: String
	ExternalID	It is mapped to the HTML attribute *id*. This attribute specifies a name to the object, which must be unique in the report. Data type: String
	ExternalStyle	It is mapped to the HTML attribute *style*. This attribute specifies style information for the object. Data type: String
	Height	Specifies the height of the object, in inches or centimeters. Data type: Float
	Horizontal	Specifies whether to show the table in horizontal style. Data type: Boolean
	Invisible	Specifies whether to show the object in the design area and in the report result. All formulas and calculations will still be performed if the property is set to true. Data type: Boolean
	LeftLine	Specifies the line style of the left border of the object. Data type: Enumeration
	position	Specifies the position of the object. • **static** - The object is positioned at the location in which it is inserted. The X, Y and other position-related properties will be hidden or disabled. The position of the object is affected by preceding objects in the same container. • **absolute** - The object will be located at the position specified by dragging and dropping or by setting its X and Y property values. The position of the object is not affected by preceding objects in the same container. Data type: String **Note:** This property only affects objects with a parent container that is the report body, a tabular cell, or a text box.
	RecordLocation	Specifies the calculation point for the properties which use formulas. • **Default** - The values of the properties are calculated in the default location where they are located. • **Page Header** - The values of the properties are calculated in the page header. • **Page Footer** - The values of the properties are calculated in the page footer. Data type: String
	RightLine	Specifies the line style of the right border of the object. Data type: Enumeration
	Style	The property can be used in two ways. • Specifies a style to be applied to the object. This works when a StyleGroup has been specified at the report sheet level and when there are styles in the StyleGroup that can be applied to the object. • Specifies a CSS selector to be applied to the object. Data type: String
	Summary	It is mapped to the HTML attribute *summary*. This attribute provides a summary of the object's purpose and structure. Data type: String
	Suppressed	Specifies whether to show the object in the design area and in the report result. All formulas and calculations will be skipped if the property is set to true. Data type: Boolean **Note:** When both the Invisible and Suppressed properties of an object are set to true, Suppressed has the higher priority.
	SuppressedIfNoRecords	Specifies whether to display the object in the report result when no record is returned to its parent data container. Data type: Boolean
	SuppressEmpty	Specifies whether to display the object in the report result when no record is returned to it. Data type: Boolean
	TOCAnchor	Specifies whether to include the object in the TOC Browser for the report. Data type: Boolean
	TopLine	Specifies the line style of the top border of the object. Data type: Enumeration
	Width	Specifies the width of the object, in inches or centimeters. Data type: Float
	X	Specifies the horizontal coordinate of the top left corner of the object, relative to its parent container. This property is ignored if the position property is set to static. Data type: Float
	Y	Specifies the vertical coordinate of the top left corner of the object, relative to its parent container. This property is ignored if the position property is set to static. Data type: Float
Table Column	AutoFit	Specifies whether to adjust the width of the object according to the contents. If the property is set to true, the width of the object will be adjusted automatically only when: • The position property for the field/label in this object is static or relative. • If the position property for the field/label in this object is absolute, then the AutoFit property for the field/label should be set to true. Data type: Boolean
	Invisible	Specifies whether to show the object in the design area and in the report result. All formulas and calculations will still be performed if the property is set to true. Data type: Boolean
Table Cell	Abbr	It is mapped to the HTML attribute *abbr*. This attribute specifies an abbreviated form of the content of the object. The abbreviated text may be rendered by user agents when appropriate in place of the object's content. Data type: String
	Axis	It is mapped to the HTML attribute *axis*. This attribute is used to place a cell into conceptual categories that can be considered to form axes in an n-dimensional space. The value of this attribute is a comma-separated list of category names. Data type: String
	Background	Specifies the background color of the object. Data type: String
	BorderColor	Specifies the color of the border of the object. Data type: String
	BorderWidth	Specifies the width of the border in inches or centimeters. Data type: Float
	BottomLine	Specifies the line style of the bottom border of the object. Data type: Enumeration
	class	Specifies a CSS class to be applied to the object which is a valid class in the CSS file. Data type: String
	ExportToCSV	Specifies whether to include the object when exporting the report to CSV. Data type: Boolean
	ExportToDHTML	Specifies whether to include the object when exporting the report to Page Report Result. Data type: Boolean
	ExportToXLS	Specifies whether to include the object when exporting the report to Excel. Data type: Boolean
	ExternalCSSClassValue	Specifies a class selector to be applied to the object when exported as HTML. Data type: String
	ExternalID	It is mapped to the HTML attribute *id*. This attribute specifies a name to the object, which must be unique in the report. Data type: String
	Headers	It is mapped to the HTML attribute *headers*. This attribute specifies the list of header cells that provide header information for the current data cell. The value of this attribute is a space-separated list of cell names; those cells must be named by setting their *id* attribute. Data type: String
	Height	Specifies the height of the object, in inches or centimeters. Data type: Float
	HorizontalAlignment	Specifies the horizontal justification of the content in the table cell. Data type: Enumeration **Note:** When the position property of the object in the table cell is set to absolute, this property does not take effect.
	JoiningMerge	Specifies whether to make all the sequential rows included in the cell vertically merged in the report result. It is meaningful to set the property only when the cell is merged with other cells in the same column. Data type: Boolean
	LeftLine	Specifies the line style of the left border of the object. Data type: Enumeration
	RecordLocation	Specifies the calculation point for the properties which use formulas. • **Default** - The values of the properties are calculated in the default location where they are located. • **Page Header** - The values of the properties are calculated in the page header. • **Page Footer** - The values of the properties are calculated in the page footer. Data type: String
	Repeat	Specifies whether to repeat the content of the cell in every page of the report result when the cell is split into pages. Available only for vertically merged cell. Data type: Boolean

	Bigfoot Gas	Gas type: the first number of the name of the Gas. Bigfoot
	Bigfoot Gas Mixture	Gas type: Mixture. Gas type: the first number of the name of the Gas. Bigfoot
	Bigfoot	Gas type: the first number of the name of the Gas. Bigfoot **Gas Mixture** - The name of the gas mixture. Bigfoot **Gas Mixture** - The name of the gas mixture. Bigfoot **Gas Mixture** - The name of the gas mixture

	ColumnNumber	Specifies the number of columns which will be the object's width in the exported Excel or CSV file. The Columned property at the report tab level must be set to true for this property to take effect. Data type: String
	ExportToCSV	Specifies whether to include the object when exporting the report to CSV. Data type: Boolean
	ExportToDHTML	Specifies whether to include the object when exporting the report to Page Report Result. Data type: Boolean
	ExportToXLS	Specifies whether to include the object when exporting the report to Excel. Data type: Boolean
	ExternalCSSClassValue	Specifies a class selector to be applied to the object when exported as HTML. Data type: String
	ExternalID	A representation of the standard HTML attribute *id*. This attribute specifies a name to the object, which must be unique in the report. Data type: String
	ExternalTitle	A representation of the standard HTML attribute *title*. This attribute offers advisory information about the object. Data type: String
	GotoDetail	Specifies whether to show the detailed information when you click the object in Page Report Studio. This property is available only when the object is in the group header/footer panel of a banded object. Data type: Boolean
	GroupValueFormat	Specifies the encoding format for values on the category axis. Data type: String
	Height	Specifies the height of the object, in inches or centimeters. Data type: Float
	HintFilter	Specifies the number format for the hint message. Data type: String
	HintPercentFilter	Specifies the format for each hint percentage to the total. Data type: String
	HyperLink	Specifies whether to add a hyperlink that refers to another report or a website to the chart. Data type: String
	HyperLinkTarget	Specifies the target window or frame in which to display the content the Hyperlink property specifies. • **<Server Setting>** Loads the linked file according to setting of the Pop Up New Window for Links option in the Customize Profile panel of the Profile dialog in JReport Server. • **New Window** Loads the linked file into a new window. The window is not named. • **Whole Window** Loads the linked file into the full browser window. • **Same Frame** Loads the linked file into the same frame as the link. • **Parent Frame** Loads the linked file into the parent frame of the frame that contains the link. • **Other Frame** Loads the linked file into some other specified frame. If the frame name does not exist, the linked file will be loaded into a new window. Data type: String
	Invisible	Specifies whether to show the object in the design area and in the report result. All formulas and calculations will still be performed if the property is set to true. Data type: Boolean
	LogicColumn	Specifies whether to show the object in the next visible table cell in the same row when the column which holds the object is hidden. Data type: Enumeration **Notes:** • The next visible table cell should be completely empty, that is, the cell should not hold any content including blank space. • When the property for several objects in the same row is set to next visible column, and the columns holding these objects are all hidden, then only the object in the rightmost column will be shown in the next visible cell.
	LongDesc	A representation of the standard HTML attribute *longdesc*. This attribute specifies a link to a long description of the object. Data type: String
	MotionFilter	Specifies the data format for the motion field values displayed on the motion bar. Data type: String
	PatternList	Specifies patterns of the data markers. The patterns can be one or more of the following: Automatic, Color, Texture, and Gradient. Data type: String
	PlatformBackgroundBorderPenColor	Specifies the color of the platform border. Data type: String
	PlatformBackgroundBorderPenEndCaps	Specifies the ending style of the platform border line. Data type: Enumeration
	PlatformBackgroundBorderPenLineJoint	Specifies the joint style of the platform border line. Data type: Enumeration
	PlatformBackgroundBorderPenLineJointRadius	Specifies the radius for the joint of the platform border line. The property takes effect only when PlatformBackgroundBorderPenLineJoint is set to joint round. Data type: Float
	PlatformBackgroundBorderPenOutlined	Specifies whether to show the platform border line in Outline Path. If the property is set to true, the platform border line will be shown in Outline Path; otherwise, in Fill Path. Data type: Boolean
	PlatformBackgroundBorderPenStyle	Specifies the line style of the platform border. Data type: Enumeration
	PlatformBackgroundBorderPenThickness	Specifies the width of the platform border, in inches or centimeters. Data type: Float
	PlatformBackgroundBorderPenTransparency	Specifies the transparency of the platform border, in percent. Data type: Integer
	PlatformBackgroundBorderPenVariabledDash	Specifies whether to resize the dash automatically. • **true** - If selected, the dash size will be adjusted automatically, and the option Auto Adjust Dash in the Format Platform dialog will take effect. • **false** - If selected, the dash size will be of fixed size, and the option Fixed Dash Size in the Format Platform dialog will take effect. Data type: Boolean
	PlatformBackgroundBorderStyle	Specifies the border type of the platform. Data type: Enumeration
	PlatformBackgroundBottom	Specifies the left position of the background area, measured in a percentage of the chart width, from the left edge of the chart. Data type: Float
	PlatformBackgroundFillPatternColor	Specifies the color of the platform background, applied when the property PlatformBackgroundFillPatternType is set to color. Data type: String
	PlatformBackgroundFillPatternGradientEndColor	Specifies the color of the point where the gradient ends, applied when the property PlatformBackgroundFillPatternType is set to gradient. Data type: String
	PlatformBackgroundFillPatternGradientEndX	Specifies the horizontal position where the gradient ends, measured in a percentage of the platform width. Applied when the property PlatformBackgroundFillPatternType is set to gradient. Data type: Integer
	PlatformBackgroundFillPatternGradientEndY	Specifies the vertical position where the gradient ends, measured in a percentage of the platform height. Applied when the property PlatformBackgroundFillPatternType is set to gradient. Data type: Integer
	PlatformBackgroundFillPatternGradientStartColor	Specifies the color of the point where the gradient begins, applied when the property PlatformBackgroundFillPatternType is set to gradient. Data type: String
	PlatformBackgroundFillPatternGradientStartX	Specifies the horizontal position where the gradient begins, measured in a percentage of the platform width. Applied when the property PlatformBackgroundFillPatternType is set to gradient. Data type: Integer
	PlatformBackgroundFillPatternGradientStartY	Specifies the vertical position where the gradient begins, measured in a percentage of the platform height. Applied when the property PlatformBackgroundFillPatternType is set to gradient. Data type: Integer
	PlatformBackgroundFillPatternGradientStyle	Specifies the gradient style for the platform background, applied when the property PlatformBackgroundFillPatternType is set to gradient. Data type: Enumeration
	PlatformBackgroundFillPatternImageFile	Specifies the file name of the image, a portion of which will be displayed as the platform background. Applied when PlatformBackgroundFillPatternType is set to image. Enter the file name with suffix in the value cell. If the image is outside of the current catalog, you should include the full path of the image correctly. Data type: String
	PlatformBackgroundFillPatternImageHeight	The four properties, PlatformBackgroundFillPatternImageX, PlatformBackgroundFillPatternImageY, PlatformBackgroundFillPatternImageWidth, and PlatformBackgroundFillPatternImageHeight, together are used to define a portion of the image (the portion is a rectangle), which will be displayed as the platform background. They take effect when the property PlatformBackgroundFillPatternType is set to image. Image Height specifies the height of the image portion, measured in a percentage of the image height. Data type: Integer
	PlatformBackgroundFillPatternImageLayout	Specifies the layout style for the image that will be displayed as the platform background, applied when the property PlatformBackgroundFillPatternType is set to image. Data type: Enumeration
	PlatformBackgroundFillPatternImageWidth	The four properties, PlatformBackgroundFillPatternImageX, PlatformBackgroundFillPatternImageY, PlatformBackgroundFillPatternImageWidth, and PlatformBackgroundFillPatternImageHeight, together are used to define a portion of the image (the portion is a rectangle), which will be displayed as the platform background. They take effect when the property PlatformBackgroundFillPatternType is set to image. Image Width specifies the width of the image portion, measured in a percentage of the image width. Data type: Integer
	PlatformBackgroundFillPatternImageX	The four properties, PlatformBackgroundFillPatternImageX, PlatformBackgroundFillPatternImageY, PlatformBackgroundFillPatternImageWidth, and PlatformBackgroundFillPatternImageHeight, together are used to define a portion of the image (the portion is a rectangle), which will be displayed as the platform background. They take effect when the property PlatformBackgroundFillPatternType is set to image. Image X specifies the distance between the left border of image and the portion that will contain the pattern, measured in a percentage of the image width. Data type: Integer
	PlatformBackgroundFillPatternImageY	The four properties, PlatformBackgroundFillPatternImageX, PlatformBackgroundFillPatternImageY, PlatformBackgroundFillPatternImageWidth, and PlatformBackgroundFillPatternImageHeight, together are used to define a portion of the image (the portion is a rectangle), which will be displayed as the platform background. They take effect when the property PlatformBackgroundFillPatternType is set to image. Image Y specifies the distance between the top border of image and the portion that will contain the pattern, measured in a percentage of the image height. Data type: Integer
	PlatformBackgroundFillPatternTextureBackColor	Specifies the background color of the texture, applied when the property PlatformBackgroundFillPatternType is set to texture. Data type: String
	PlatformBackgroundFillPatternTextureForeColor	Specifies the foreground color of the texture, applied when the property PlatformBackgroundFillPatternType is set to texture. Data type: String
	PlatformBackgroundFillPatternTextureStyle	Specifies the texture style of the platform background, applied when the property PlatformBackgroundFillPatternType is set to texture. Data type: Enumeration
	PlatformBackgroundFillPatternTransparency	Specifies the transparency of the platform background, in percent. Data type: Integer
	PlatformBackgroundFillPatternType	Specifies the fill pattern for the platform background. Data type: Enumeration
	PlatformBackgroundLeft	Specifies the left position of the background area, measured in a percentage of the chart width, from the left edge of the chart. Data type: Float
	PlatformBackgroundRight	Specifies the right position of the background area, measured in a percentage of the chart width, from the right edge of the chart. Data type: Float
	PlatformBackgroundTop	Specifies the top position of the background area, measured in a percentage of the chart height, from the top edge of the chart. Data type: Float
	position	Specifies the position of the object. • **static** - The object is positioned at the location in which it is inserted. If selected, the X, Y and other position-related properties will be hidden or disabled. The position of the object is affected by preceding objects in the same container. • **absolute** - The object will be located at the position specified by dragging and dropping or by setting its X and Y property values. The position of the object is not affected by preceding objects in the same container. Data type: Enumeration **Note:** This property only affects objects with a parent container that is the report body, a table cell, a tabular cell, or a text box.
	PrimaryDataFilter	Specifies the data format for the primary value axis (the Y1 axis) to display the tick mark labels in the way you choose. Data type: String
	ReverseCategory	Specifies whether to reverse the sequence of the category field value. Data type: Boolean
	ReverseSeries	Specifies whether to reverse the sequence of the series field value. Data type: Boolean
	RowIndex	Specifies the Y coordinate of the object relative to its parent container when exported to Excel or CSV, measured in cells. The Columned property at the report tab level must be set to true for this property to take effect. Data type: Integer
	RowNumber	Specifies the number of rows which will be the object's height in the exported Excel or CSV file. The Columned property at the report tab level must be set to true for this property to take effect. Data type: String
	SecondaryDataFilter	Specifies the data format for the secondary value axis (the Y2 axis) to display the tick mark labels in the way you choose. Data type: String
	SeriesEndOffset	The four properties work together to control the range of the data being displayed on a chart: CategoryStartOffset, CategoryEndOffset, SeriesStartOffset, and SeriesEndOffset. SeriesEndOffset specifies the ending offset of the series. Data type: Integer
	SeriesFilter	Specifies the data format for the series axis (the Z axis) to display the tick mark labels in the way you choose. Data type: String
	SeriesStartOffset	The four properties work together to control the range of the data being displayed on a chart: CategoryStartOffset, CategoryEndOffset, SeriesStartOffset, and SeriesEndOffset. SeriesStartOffset specifies the starting offset of the series. Data type: Integer
	SeriesValueEncoding	Specifies the encoding format for values on the series axis. Data type: String
	ShowLegend	Specifies whether to make the legend visible in a chart. Data type: Boolean
	SortCategory	Specifies the sorting order for the category field values. Data type: Enumeration
	SortSeries	Specifies the sorting order for the series field values. Data type: Enumeration
	Style	The property can be used in two ways. • Specifies a style to be applied to the object. This works when a StyleGroup has been specified at the report sheet level and when there are styles in the StyleGroup that can be applied to the object. • Specifies a CSS selector to be applied to the object. Data type: String
	Suppressed	Specifies whether to show the object in the design area and in the report result. All formulas and calculations will be skipped if the property is set to true. Data type: Boolean **Note:** When both the Invisible and Suppressed properties of an object are set to true, Suppressed has the higher priority.
	SuppressedIfNoRecords	Specifies whether to display the object in the report result when no record is returned to its parent data container. Data type: Boolean
	SwapGroup	Specifies to display values from different data fields by switching data between the category and series axes, the category and value axes of a chart. By default, this property is set to false, which means no switch will take place in the chart. • **Switching between the category and series axes** If you create a chart by adding fields to all of its axes, and set SwapGroup to true, data on the category and series axes will be switched. • **Switching between the category and value axes** If you create a chart by adding one field to the category axis and several to the value axis, and set SwapGroup to true, data on the category and value axes will be switched. Data type: Boolean **Note:** If the chart has only one field on the category axis and one on the value axis, even though the property value is set to true, there will be no switch take place on the two axes.
	TOCAnchor	Specifies whether to include the object in the TOC Browser for the report. Data type: Boolean
	Width	Specifies the width of the object, in inches or centimeters. Data type: Float
	X	Specifies the horizontal coordinate of the top left corner of the object, relative to its parent container. This property is ignored if the position property is set to static. Data type: Float
	XHyperLink	Specifies whether to add a hyperlink that refers to another report or a website to the X axis labels. Data type: String
	XHyperLinkTarget	Specifies the target window or frame in which to display the content the XHyperlink property specifies. • **<Server Setting>** Loads the linked file according to setting of the Pop Up New Window for Links option in the Customize Profile panel of the Profile dialog in JReport Server. • **New Window** Loads the linked file into a new window. The window is not named. • **Whole Window** Loads the linked file into the full browser window. • **Same Frame** Loads the linked file into the same frame as the link. • **Parent Frame** Loads the linked file into the parent frame of the frame that contains the link. • **Other Frame** Loads the linked file into some other specified frame. If the frame name does not exist, the linked file will be loaded into a new window. Data type: String
	Y	Specifies the vertical coordinate of the top left corner of the object, relative to its parent container. This property is ignored if the position property is set to static. Data type: Float
	ZHyperLink	Specifies whether to add a hyperlink that refers to another report or a website to the Z axis labels. Data type: String
	ZHyperLinkTarget	Specifies the target window or frame in which to display the content the ZHyperlink property specifies. • **<Server Setting>** Loads the linked file according to setting of the Pop Up New Window for Links option in the Customize Profile panel of the Profile dialog in JReport Server. • **New Window** Loads the linked file into a new window. The window is not named. • **Whole Window** Loads the linked file into the full browser window. • **Same Frame** Loads the linked file into the same frame as the link. • **Parent Frame** Loads the linked file into the parent frame of the frame that contains the link. • **Other Frame** Loads the linked file into some other specified frame. If the frame name does not exist, the linked file will be loaded into a new window. Data type: String
Chart Paper	Background	Specifies the properties of the chart paper background.
	BorderColor	Specifies the color of the paper border. Data type: String
	BorderWidth	Specifies the width of the paper border, in inches or centimeters. Data type: Float
	BottomLine	Specifies the line style of the bottom border of the object. Data type: Enumeration
	class	Specifies a CSS class to be applied to the object which is a valid class in the CSS file. Data type: String
	ColumnIndex	Specifies the X coordinate of the object relative to its parent container when exported to Excel or CSV, measured in cells. The Columned property at the report tab level must be set to true for this property to take effect. Data type: Integer
	ExportToCSV	Specifies whether to include the object when exporting the report to CSV. Data type: Boolean
	ExportToDHTML	Specifies whether to include the object when exporting the report to Page Report Result. Data type: Boolean
	ExportToXLS	Specifies whether to include the object when exporting the report to Excel. Data type: Boolean
	ExternalCSSClassValue	Specifies a class selector to be applied to the object when exported as HTML. Data type: String
	Height	Specifies the height of the object, in inches or centimeters. Data type: Float
	Invisible	Specifies whether to show the object in the design area and in the report result. All formulas and calculations will still be performed if the property is set to true. Data type: Boolean
	LeftLine	Specifies the line style of the left border of the object. Data type: Enumeration
	position	Specifies the position of the object. • **static** - The object is positioned at the location in which it is inserted. The X, Y and other position-related properties will be hidden or disabled. The position of the object is affected by preceding objects in the same container. • **absolute** - The object will be located at the position specified by dragging and dropping or by setting its X and Y property values. The position of the object is not affected by preceding objects in the same container. Data type: String **Note:** This property only affects objects with a parent container that is the report body, a tabular cell, or a text box.
	RecordLocation	Specifies the calculation point for the properties which use formulas. • **Default** - The values of the properties are calculated in the default location where they are located. • **Page Header** - The values of the properties are calculated in the page header. • **Page Footer** - The values of the properties are calculated in the page footer. Data type: String
	RightLine	Specifies the line style of the right border of the object. Data type: Enumeration
	RowIndex	Specifies the Y coordinate of the object relative to its parent container when exported to Excel or CSV, measured in cells. The Columned property at the report tab level must be set to true for this property to take effect. Data type: Integer
	Style	The property can be used in two ways. • Specifies a style to be applied to the object. This works when a StyleGroup has been specified at the report sheet level and when there are styles in the StyleGroup that can be applied to the object, or type in the style name. • Specifies a CSS selector to be applied to the object. Data type: String
	Suppressed	Specifies whether to show the object in the design area and in the report result. All formulas and calculations will be skipped if the property is set to true. Data type: Boolean **Note:** When both the Invisible and Suppressed properties of an object are set to true, Suppressed has the higher priority.
	SuppressedIfNoRecords	Specifies whether to display the object in the report result when no record is returned to its parent data container. Data type: Boolean
	TopLine	Specifies the line style of the top border of the object. Data type: Enumeration
	Width	Specifies the width of the object, in inches or centimeters. Data type: Float
	X	Specifies the horizontal coordinate of the top left corner of the object, relative to its parent container. This property takes effect only when the Placement property for the chart legend is set to auto. Data type: Float
	Y	Specifies the vertical coordinate of the top left corner of the object, relative to its parent container. This property takes effect only when the Placement property for the chart legend is set to auto. Data type: Float
Chart legend	AutoSize	Specifies whether to resize the legend automatically. Data type: Boolean
	BottomMargin	Specifies the distance in inches or centimeters between the legend labels and the bottom border of the legend. Data type: Float
	class	Specifies a CSS class to be applied to the object which is a valid class in the CSS file. Data type: String
	ExternalCSSClassValue	Specifies a class selector to be applied to the object when exported as HTML. Data type: String
	Height	Specifies the height of the object, in inches or centimeters. Data type: Float
	IconAlignment	Specifies the relative position of the icon to the legend. Data type: Enumeration
	IconBorderPenColor	Specifies the color of the icon borders. Data type: String
	IconBorderPenEndCaps	Specifies the ending style of the icon border line. Data type: Enumeration
	IconBorderPenLineJoint	Specifies the joint style of the icon border line. Data type: Enumeration
	IconBorderPenOutlined	Specifies whether to show the icon border line in Outline Path. If the property is set to true, the icon border line will be shown in Outline Path; otherwise, in Fill Path. Data type: Boolean
	IconBorderPenStyle	Specifies the line style of the icon borders. Data type: Enumeration
	IconBorderPenThickness	Specifies the width of the icon borders in inches or centimeters. Data type: Float
	IconBorderPenTransparency	Specifies the transparency of the icon borders, in percent. Data type: Integer
	IconBorderPenVariabledDash	Specifies whether to resize the dash automatically. If the property is set to true, the dash size will be adjusted automatically; otherwise, the dash size will be of fixed size. Data type: Boolean
	IconHeight	Specifies the height of the icon, in inches or centimeters. Data type: Float
	IconTextGap	Specifies the distance between each legend label and icon. Data type: Float
	IconWidth	Specifies the width of the icon, in inches or centimeters. Data type: Float
	LabelFilterSource	Specifies the data format option for the chart legend labels. • **Axis Data Format** - The data format of the legend labels will follow that of the axis data: if the chart has series axis, the legend label format will be the same as the series axis data format; if not, the category axis data format will be used to the legend labels. • **Legend Data Format** - The data format of the legend labels will be controlled by the property LegendFilter. Data type: Enumeration
	LabelHSpacing	Specifies the horizontal distance between two adjacent legend labels. Data type: Float
	LabelTextFontName	Specifies the font face for the legend labels. Data type: Enumeration
	LabelTextFontSize	Specifies the font size for the legend labels. Data type: Integer
	LabelTextFontStyle	Specifies the font style for the legend labels. Data type: Enumeration
	LabelTextPenStyle	Specifies the style for the label text outline. Data type: Enumeration
	LabelTextPatternColor	Specifies the color of the label, applied when the property LabelTextPatternType is set to color. Data type: String
	LabelTextPatternGradientEndColor	Specifies the color of the point where the gradient ends, applied when the property LabelTextPatternType is set to gradient. Data type: String
	LabelTextPatternGradientEndX	Specifies the horizontal position where the gradient ends, measured in a percentage of the label text width. Applied when the property LabelTextPatternType is set to gradient. Data type: Integer
	LabelTextPatternGradientEndY	Specifies the vertical position where the gradient ends, measured in a percentage of the label text height. Applied when the property LabelTextPatternType is set to gradient. Data type: Integer
	LabelTextPatternGradientStartColor	Specifies the color of the point where the gradient begins, applied when the property LabelTextPatternType is set to gradient. Data type: String
	LabelTextPatternGradientStartX	Specifies the horizontal position where the gradient begins, measured in a percentage of the label text width. Applied when the property LabelTextPatternType is set to gradient. Data type: Integer
	LabelTextPatternGradientStartY	Specifies the vertical position where the gradient begins, measured in a percentage of the label text height. Applied when the property LabelTextPatternType is set to gradient. Data type: Integer
	LabelTextPatternGradientStyle	Specifies the gradient style for the label, applied when the property LabelTextPatternType is set to gradient. The image fill type does not apply to the legend label. Data type: Enumeration
	LabelTextPatternTextureBackColor	Specifies the background color of the texture, applied when the property LabelTextPatternType is set to texture. Data type: String
	LabelTextPatternTextureForeColor	Specifies the foreground color of the texture, applied when the property LabelTextPatternType is set to texture. Data type: String
	LabelTextPatternTextureStyle	Specifies the texture style for the label, applied when the property LabelTextPatternType is set to texture. Data type: Enumeration
	LabelTextPatternTransparency	Specifies the transparency of the background, in percent. Data type: Integer
	LabelTextPatternType	Specifies the fill pattern for the label. Data type: Enumeration
	LabelTextPenColor	Specifies the color for the label text outline. Data type: String
	LabelTextPenEndCaps	Specifies the ending style for the label text outline. Data type: Enumeration
	LabelTextPenLineJoint	Specifies the joint style for the label text outline. Data type: Enumeration
	LabelTextPenOutlined	Specifies whether to show the label text outline in outline form. If the property is set to true, the outline will be shown in outline form; otherwise, in whole form. Data type: Boolean
	LabelTextPenThickness	Specifies the thickness for the label text outline. Data type: Float
	LabelTextPenTransparency	Specifies the transparency for the label text outline, in percent. Data type: Integer
	LabelTextPenVariabledDash	Specifies whether to resize the dash automatically. If the property is set to true, the dash size will be adjusted automatically; otherwise, the dash size will be of fixed size. Data type: Boolean
	LabelTextRotation	Specifies the rotation angle of each legend label around its center, in degrees. The default value is 0. Data type: Float
	LabelTextShearing	Specifies the shearing transformation of each legend label around its center. The default value is 0. Data type: Float
	LabelTextSpecialEffect1	Specifies the special effect for the label text. Data type: Enumeration
	LabelTextSpecialEffect2	Specifies whether the label text will in superscript or subscript form, or neither of them. Data type: Enumeration
	LabelTextStrikethrough	Specifies the style of the line by which the label texts are struck through. Data type: Enumeration
	LabelTextUnderline	Specifies the underline style for the legend labels. Data type: Enumeration
	LabelTextWordWrap	Specifies whether to enable the word wrap function for the label text, not applying to radar charts, bubble charts, scatter charts and gauge charts. If the text contains special characters (such as ",", "." and space), it will be broken at the position of one of the special characters. Data type: Boolean
	LabelVSpacing	Specifies the vertical distance between two adjacent legend labels. Data type: Float
	LeftMargin	Specifies the distance in inches or centimeters between the legend labels and the left border of the legend. Data type: Float
	LegendBackgroundBorderPenColor	Specifies the color of the legend border. Data type: String
	LegendBackgroundBorderPenEndCaps	Specifies the ending style of the legend border line. Data type: Enumeration
	LegendBackgroundBorderPenLineJoint	Specifies the joint style of the legend border line. Data type: Enumeration
	LegendBackgroundBorderPenOutlined	Specifies whether to show the legend border line in Outline Path. If the property is set to true, the legend border line will be shown in Outline Path; otherwise, in Fill Path. Data type: Boolean
	LegendBackgroundBorderPenStyle	Specifies the line style of the legend border. Data type: Enumeration
	LegendBackgroundBorderPenThickness	Specifies the width of the legend border, in inches or centimeters. Data type: Float
	LegendBackgroundBorderPenTransparency	Specifies the transparency of the legend border, in percent. Data type: Integer
	LegendBackgroundBorderPenVariabledDash	Specifies whether to resize the dash automatically. • **true** - If selected, the dash size will be adjusted automatically, and the option Auto Adjust Dash in the Format Legend dialog will take effect. • **false** - If selected, the dash size will be of fixed size, and the option Fixed Dash Size in the Format Legend dialog will take effect. Data type: Boolean
	LegendBackgroundBorderStyle	Specifies the border type of the legend. Data type: Enumeration
	LegendBackgroundBottom	Specifies the bottom position of the background area, measured in a percentage of the legend height, from the bottom edge of the legend. Data type: Float
	LegendBackgroundFillPatternColor	Specifies the color of the legend background, applied when the property LegendBackgroundFillPatternType is set to color. Data type: String
	LegendBackgroundFillPatternGradientEndColor	Specifies the color of the point where the gradient ends, applied when the property LegendBackgroundFillPatternType is set to gradient. Data type: String
	LegendBackgroundFillPatternGradientEndX	Specifies the horizontal position where the gradient ends, measured in a percentage of the legend width. Applied when the property LegendBackgroundFillPatternTypee is set to gradient. Data type: Integer
	LegendBackgroundFillPatternGradientEndY	Specifies the vertical position where the gradient ends, measured in a percentage of the legend height. Applied when the property LegendBackgroundFillPatternType is set to gradient. Data type: Integer
	LegendBackgroundFillPatternGradientStartColor	Specifies the color of the point where the gradient begins, applied when the property LegendBackgroundFillPatternType is set to gradient. Data type: String
	LegendBackgroundFillPatternGradientStartX	Specifies the horizontal position where the gradient begins, measured in a percentage of the legend width. Applied when the property LegendBackgroundFillPatternType is set to gradient. Data type: Integer
	LegendBackgroundFillPatternGradientStartY	Specifies the vertical position where the gradient begins, measured in a percentage of the legend height. Applied when the property LegendBackgroundFillPatternType is set to gradient. Data type: Integer
	LegendBackgroundFillPatternGradientStyle	Specifies the gradient style for the legend background, applied when the property LegendBackgroundFillPatternType is set to gradient. Data type: Enumeration
	LegendBackgroundFillPatternImageFile	Specifies the file name of the image, a portion of which will be displayed as the legend background. Applied when the property LegendBackgroundFillPatternType is set to image. Enter the file name with suffix in the value cell. If the image is outside of the current catalog, you should include the full path of the image correctly. Data type: String
	LegendBackgroundFillPatternImageHeight	The four properties, LegendBackgroundFillPatternImageX, LegendBackgroundFillPatternImageY, LegendBackgroundFillPatternImageWidth, and LegendBackgroundFillPatternImageHeight, together are used to define a portion of the image (the portion is a rectangle), which will be displayed as the legend background. They take effect when the property LegendBackgroundFillPatternType is set to image. LegendBackgroundFillPatternImageHeight specifies the height of the image portion, measured in a percentage of the image height. Data type: Integer
	LegendBackgroundFillPatternImageLayout	Specifies the layout style for the image that will be displayed as the legend background, applied when the property LegendBackgroundFillPatternType is set to image. Data type: Enumeration
	LegendBackgroundFillPatternImageWidth	The four properties, LegendBackgroundFillPatternImageX, LegendBackgroundFillPatternImageY, LegendBackgroundFillPatternImageWidth, and LegendBackgroundFillPatternImageHeight, together are used to define a portion of the image (the portion is a rectangle), which will be displayed as the legend background. They take effect when the property LegendBackgroundFillPatternType is set to image. LegendBackgroundFillPatternImageWidth specifies the width of the image portion, measured in a percentage of the image width. Data type: Integer
	LegendBackgroundFillPatternImageX	The four properties, LegendBackgroundFillPatternImageX, LegendBackgroundFillPatternImageY, LegendBackgroundFillPatternImageWidth, and LegendBackgroundFillPatternImageHeight, together are used to define a portion of the image (the portion is a rectangle), which will be displayed as the legend background. They take effect when the property LegendBackgroundFillPatternType is set to image. LegendBackgroundFillPatternImageX specifies the distance between the left border of image and the portion that will contain the pattern, measured in a percentage of the image width. Data type: Integer
	LegendBackgroundFillPatternImageY	The four properties, LegendBackgroundFillPatternImageX, LegendBackgroundFillPatternImageY, LegendBackgroundFillPatternImageWidth, and LegendBackgroundFillPatternImageHeight, together are used to define a portion of the image (the portion is a rectangle), which will be displayed as the legend background. They take effect when the property Fill Type is set to image. LegendBackgroundFillPatternImageY specifies the distance between the top border of image and the portion that will contain the pattern, measured in a percentage of the image height. Data type: Integer
	LegendBackgroundFillPatternTextureBackColor	Specifies the background color of the texture, applied when the property LegendBackgroundFillPatternType is set to texture. Data type: String
	LegendBackgroundFillPatternTextureForeColor	Specifies the foreground color of the texture, applied when the property LegendBackgroundFillPatternType is set to texture. Data type: String
	LegendBackgroundFillPatternTextureStyle	Specifies the texture style of the legend background, applied when the property LegendBackgroundFillPatternType is set to texture. Data type: Enumeration
	LegendBackgroundFillPatternTransparency	Specifies the transparency of the legend background, in percent. Data type: Integer
	LegendBackgroundFillPatternType	Specifies the fill pattern for the legend background. Data type: Enumeration
	LegendBackgroundLeft	Specifies the left position of the background area, measured in a percentage of the legend width, from the left edge of the legend. Data type: Float
	LegendBackgroundRight	Specifies the right position of the background area, measured in a percentage of the legend width, from the right edge of the legend. Data type: Float
	LegendBackgroundTop	Specifies the top position of the background area, measured in a percentage of the legend height, from the top edge of the legend. Data type: Float
	LegendFilter	Specifies the data format of the legend labels, applied when LabelFilterSource is set to Legend Data Format. Data type: String
	PercentFormat	Specifies the format of the percentage values on the legend. Applied when ShowPercent is set to true. Data type: String
	Placement	Specifies the position of the legend in the platform. Data type: Enumeration
	ReverseLabels	Specifies whether to reverse the order of the labels in the legend object. Data type: Boolean
	RightMargin	Specifies the distance in inches or centimeters between the legend labels and the right border of the legend. Data type: Float
	ShowPercent	Specifies whether to show the percentage of each entry value to the total on the legend. Applies to pie, bar and bench charts that have only one value axis. Data type: Boolean
	ShowScrollbar	Specifies whether to show a scrollbar on the legend to fully view the legend content when the content does not fit into the legend. Data type: Boolean
	ShowTips	Specifies whether to show the corresponding data information when the mouse pointer points at a target in the chart legend in Designer view mode, in HTML result, in Page Report Studio, or in Web Report Studio or JDashboard. Data type: Boolean
	ShowValues	Specifies whether to show the values for the legend entries. Applies to pie, bar and bench charts that have only one value axis. Data type: Boolean
	Style	The property can be used in two ways. • Specifies a style to be applied to the object. This works when a StyleGroup has been specified at the report sheet level and when there are styles in the StyleGroup that can be applied to the object, or type in the style name. • Specifies a CSS selector to be applied to the object. Data type: String
	SubPlacement	Specifies the position of the legend further based on the setting of the property Placement. Data type: Enumeration
	TopMargin	Specifies the distance in inches or centimeters between the legend labels and the top border of the legend. Data type: Float
	TotalValueText	Specifies the text of the total value label on the legend. Applies to pie, bar and bench charts that have only one value axis. Data type: String
	Truncate	Specifies whether to truncate the legend entry label text when the text overflow the labels. Data type: Boolean
	ValueFormat	Specifies the number format of the legend entry values. Applied when ShowValues is set to true. Data type: String
	Width	Specifies the width of the object, in inches or centimeters. Data type: Float
	X	Specifies the horizontal coordinate of the top left corner of the object, relative to its parent container. This property takes effect only when the Placement property is set to auto. Data type: Float
	Y	Specifies the vertical coordinate of the top left corner of the object, relative to its parent container. This property takes effect only when the Placement property is set to auto. Data type: Float
Chart Label	Alt	Specifies the alternate text of the image, which will be shown when the image cannot be displayed. This property applies only when the chart is exported to HTML as an image chart. Data type: String
	AnchorDispValue	Specifies a string or formula to display for the TOC entry for this object in the TOC Browser. The TOCAnchor property must be set to true for this property to take effect. Data type: String
	AntiAliasing	Specifies whether to make the edges in a chart smooth. Data type: Boolean
	Cache	Specifies whether to cache the dataset for this object in the data buffer so other objects which use the same dataset can share the data rather than perform their own SQL query. Data type: Boolean
	CategoryEndOffset	The four properties work together to control the range of the data being displayed on a chart: CategoryStartOffset, CategoryEndOffset, SeriesStartOffset, and SeriesEndOffset. CategoryEndOffset specifies the ending offset of the categories. It does not apply to the charts which involve only one group. Data type: Integer
	CategoryFilter	Specifies the data format for the category axis (the X axis) to display the tick mark labels in the way you choose. Data type: String

CategoryStartOffset	The four properties work together to control the range of the data being displayed on a chart: CategoryStartOffset, CategoryEndOffset, SeriesStartOffset, and SeriesEndOffset. CategoryStartOffset specifies the starting offset of the categories. It does not apply to the charts which involve only one group. Data type: Integer
class	Specifies a CSS class to be applied to the object which is a valid class in the CSS file. Data type: String
ColumnIndex	Specifies the X coordinate of the object relative to its parent container when exported to Excel or CSV, measured in cells. The Columned property at the report tab level must be set to true for this property to take effect. Data type: Integer
ColumnNumber	Specifies the number of columns which will be the object's width in the exported Excel or CSV file. The Columned property at the report tab level must be set to true for this property to take effect. Data type: String
ExportToCSV	Specifies whether to include the object when exporting the report to CSV. Data type: Boolean
ExportToDHTML	Specifies whether to include the object when exporting the report to Page Report Result. Data type: Boolean
ExportToXLS	Specifies whether to include the object when exporting the report to Excel. Data type: Boolean
ExternalCSSClassValue	Specifies a class selector to be applied to the object when exported as HTML. Data type: String
ExternalID	It is mapped to the HTML attribute *id*. This attribute specifies a name to the object, which must be unique in the report. Data type: String
ExternalTitle	It is mapped to the HTML attribute *title*. This attribute offers advisory information about the object. Data type: String
GotoDetail	Specifies whether to show the detailed information when you click the object in Page Report Studio. This property is available only when the object is in the group header/footer panel of a banded object. Data type: Boolean
GroupValueFormat	Specifies the encoding format for values on the category axis. Data type: String
Height	Specifies the height of the object, in inches or centimeters. Data type: Float
HintFilter	Specifies the number format for the hint message. Data type: String
HintPercentFilter	Specifies the format for each hint percentage to the total. Data type: String
HyperLink	Specifies whether to add a hyperlink that refers to another report or a website to the chart. Data type: String
HyperLinkTarget	Specifies the target window or frame in which to display the content the Hyperlink property specifies. • **<Server Setting>** Loads the linked file according to setting of the Pop Up New Window for Links option in the Customize Profile panel of the Profile dialog in JReport Server. • **New Window** Loads the linked file into a new window. The window is not named. • **Whole Window** Loads the linked file into the full browser window. • **Same Frame** Loads the linked file into the same frame as the link. • **Parent Frame** Loads the linked file into the parent frame of the frame that contains the link. • **Other Frame** Loads the linked file into some other specified frame. If the frame name does not exist, the linked file will be loaded into a new window. Data type: String
Invisible	Specifies whether to show the object in the design area and in the report result. All formulas and calculations will still be performed if the property is set to true. Data type: Boolean
LogicColumn	Specifies whether to show the object in the next visible table cell in the same row when the column which holds the object is hidden. Data type: Enumeration **Notes:** • The next visible table cell should be completely empty, that is, the cell should not hold any content including blank space. • When the property for several objects in the same row is set to next visible column, and the columns holding these objects are all hidden, then only the object in the rightmost column will be shown in the next visible cell.
LongDesc	A representation of the standard HTML attribute *longdesc*. This attribute specifies a link to a long description of the object. Data type: String
MotionFilter	Specifies the data format for the motion field values displayed on the motion bar. Data type: String
PatternList	Specifies patterns of the data markers. The patterns can be one or more of the following: Automatic, Color, Texture, and Gradient. Data type: String
PlatformBackgroundBorderPenColor	Specifies the color of the platform border. Data type: String
PlatformBackgroundBorderPenEndCaps	Specifies the ending style of the platform border line. Data type: Enumeration
PlatformBackgroundBorderPenLineJoint	Specifies the joint style of the platform border line. Data type: Enumeration
PlatformBackgroundBorderPenLineJointRadius	Specifies the radius for the joint of the platform border line. The property takes effect only when PlatformBackgroundBorderPenLineJoint is set to joint round. Data type: Float
PlatformBackgroundBorderPenOutlined	Specifies whether to show the platform border line in Outline Path. If the property is set to true, the platform border line will be shown in Outline Path; otherwise, in Fill Path. Data type: Boolean
PlatformBackgroundBorderPenStyle	Specifies the line style of the platform border. Data type: Enumeration
PlatformBackgroundBorderPenThickness	Specifies the width of the platform border, in inches or centimeters. Data type: Float
PlatformBackgroundBorderPenTransparency	Specifies the transparency of the platform border, in percent. Data type: Integer
PlatformBackgroundBorderPenVariabledDash	Specifies whether to resize the dash automatically. • **true** - If selected, the dash size will be adjusted automatically, and the option Auto Adjust Dash in the Format Platform dialog will take effect. • **false** - If selected, the dash size will be of fixed size, and the option Fixed Dash Size in the Format Platform dialog will take effect. Data type: Boolean
PlatformBackgroundBorderStyle	Specifies the border type of the platform. Data type: Enumeration
PlatformBackgroundBottom	Specifies the left position of the background area, measured in a percentage of the chart width, from the left edge of the chart. Data type: Float
PlatformBackgroundFillPatternColor	Specifies the color of the platform background, applied when the property PlatformBackgroundFillPatternType is set to color. Data type: String
PlatformBackgroundFillPatternGradientEndColor	Specifies the color of the point where the gradient ends, applied when the property PlatformBackgroundFillPatternType is set to gradient. Data type: String
PlatformBackgroundFillPatternGradientEndX	Specifies the horizontal position where the gradient ends, measured in a percentage of the platform width. Applied when the property PlatformBackgroundFillPatternType is set to gradient. Data type: Integer
PlatformBackgroundFillPatternGradientEndY	Specifies the vertical position where the gradient ends, measured in a percentage of the platform height. Applied when the property PlatformBackgroundFillPatternType is set to gradient. Data type: Integer
PlatformBackgroundFillPatternGradientStartColor	Specifies the color of the point where the gradient begins, applied when the property PlatformBackgroundFillPatternType is set to gradient. Data type: String
PlatformBackgroundFillPatternGradientStartX	Specifies the horizontal position where the gradient begins, measured in a percentage of the platform width. Applied when the property PlatformBackgroundFillPatternType is set to gradient. Data type: Integer
PlatformBackgroundFillPatternGradientStartY	Specifies the vertical position where the gradient begins, measured in a percentage of the platform height. Applied when the property PlatformBackgroundFillPatternType is set to gradient. Data type: Integer
PlatformBackgroundFillPatternGradientStyle	Specifies the gradient style for the platform background, applied when the property PlatformBackgroundFillPatternType is set to gradient. Data type: Enumeration
PlatformBackgroundFillPatternImageFile	Specifies the file name of the image, a portion of which will be displayed as the platform background. Applied when PlatformBackgroundFillPatternType is set to image. Enter the file name with suffix in the value cell. If the image is outside of the current catalog, you should include the full path of the image correctly. Data type: String
PlatformBackgroundFillPatternImageHeight	The four properties, PlatformBackgroundFillPatternImageX, PlatformBackgroundFillPatternImageY, PlatformBackgroundFillPatternImageWidth, and PlatformBackgroundFillPatternImageHeight, together are used to define a portion of the image (the portion is a rectangle), which will be displayed as the platform background. They take effect when the property PlatformBackgroundFillPatternType is set to image. Image Height specifies the height of the image portion, measured in a percentage of the image height. Data type: Integer
PlatformBackgroundFillPatternImageLayout	Specifies the layout style for the image that will be displayed as the platform background, applied when the property PlatformBackgroundFillPatternType is set to image. Data type: Enumeration
PlatformBackgroundFillPatternImageWidth	The four properties, PlatformBackgroundFillPatternImageX, PlatformBackgroundFillPatternImageY, PlatformBackgroundFillPatternImageWidth, and PlatformBackgroundFillPatternImageHeight, together are used to define a portion of the image (the portion is a rectangle), which will be displayed as the platform background. They take effect when the property PlatformBackgroundFillPatternType is set to image. Image Width specifies the width of the image portion, measured in a percentage of the image width. Data type: Integer
PlatformBackgroundFillPatternImageX	The four properties, PlatformBackgroundFillPatternImageX, PlatformBackgroundFillPatternImageY, PlatformBackgroundFillPatternImageWidth, and PlatformBackgroundFillPatternImageHeight, together are used to define a portion of the image (the portion is a rectangle), which will be displayed as the platform background. They take effect when the property PlatformBackgroundFillPatternType is set to image. Image X specifies the distance between the left border of image and the portion that will contain the pattern, measured in a percentage of the image width. Data type: Integer
PlatformBackgroundFillPatternImageY	The four properties, PlatformBackgroundFillPatternImageX, PlatformBackgroundFillPatternImageY, PlatformBackgroundFillPatternImageWidth, and PlatformBackgroundFillPatternImageHeight, together are used to define a portion of the image (the portion is a rectangle), which will be displayed as the platform background. They take effect when the property PlatformBackgroundFillPatternType is set to image. Image Y specifies the distance between the top border of image and the portion that will contain the pattern, measured in a percentage of the image height. Data type: Integer
PlatformBackgroundFillPatternTextureBackColor	Specifies the background color of the texture, applied when the property PlatformBackgroundFillPatternType is set to texture. Data type: String
PlatformBackgroundFillPatternTextureForeColor	Specifies the foreground color of the texture, applied when the property PlatformBackgroundFillPatternType is set to texture. Data type: String
PlatformBackgroundFillPatternTextureStyle	Specifies the texture style of the platform background, applied when the property PlatformBackgroundFillPatternType is set to texture. Data type: Enumeration
PlatformBackgroundFillPatternTransparency	Specifies the transparency of the platform background, in percent. Data type: Integer
PlatformBackgroundFillPatternType	Specifies the fill pattern for the platform background. Data type: Enumeration
PlatformBackgroundLeft	Specifies the left position of the background area, measured in a percentage of the chart width, from the left edge of the chart. Data type: Float
PlatformBackgroundRight	Specifies the right position of the background area, measured in a percentage of the chart width, from the right edge of the chart. Data type: Float
PlatformBackgroundTop	Specifies the top position of the background area, measured in a percentage of the chart height, from the top edge of the chart. Data type: Float
position	Specifies the position of the object. • **static** - The object is positioned at the location in which it is inserted. If selected, the X, Y and other position-related properties will be hidden or disabled. The position of the object is affected by preceding objects in the same container. • **absolute** - The object will be located at the position specified by dragging and dropping or by setting its X and Y property values. The position of the object is not affected by preceding objects in the same container. Data type: Enumeration **Note:** This property only affects objects with a parent container that is the report body, a table cell, a tabular cell, or a text box.
PrimaryDataFilter	Specifies the data format for the primary value axis (the Y1 axis) to display the tick mark labels in the way you choose. Data type: String
ReverseCategory	Specifies whether to reverse the sequence of the category field value. Data type: Boolean
ReverseSeries	Specifies whether to reverse the sequence of the series field value. Data type: Boolean
RowIndex	Specifies the Y coordinate of the object relative to its parent container when exported to Excel or CSV, measured in cells. The Columned property at the report tab level must be set to true for this property to take effect. Data type: Integer
RowNumber	Specifies the number of rows which will be the object's height in the exported Excel or CSV file. The Columned property at the report tab level must be set to true for this property to take effect. Data type: String
SecondaryDataFilter	Specifies the data format for the secondary value axis (the Y2 axis) to display the tick mark labels in the way you choose. Data type: String
SeriesEndOffset	The four properties work together to control the range of the data being displayed on a chart: CategoryStartOffset, CategoryEndOffset, SeriesStartOffset, and SeriesEndOffset. SeriesEndOffset specifies the ending offset of the series. Data type: Integer
SeriesFilter	Specifies the data format for the series axis (the Z axis) to display the tick mark labels in the way you choose. Data type: String
SeriesStartOffset	The four properties work together to control the range of the data being displayed on a chart: CategoryStartOffset, CategoryEndOffset, SeriesStartOffset, and SeriesEndOffset. SeriesStartOffset specifies the starting offset of the series. Data type: Integer
SeriesValueEncoding	Specifies the encoding format for values on the series axis. Data type: String
ShowLegend	Specifies whether to make the legend visible in a chart. Data type: Boolean
SortCategory	Specifies the sorting order for the category field values. Data type: Enumeration
SortSeries	Specifies the sorting order for the series field values. Data type: Enumeration
Style	The property can be used in two ways. • Specifies a style to be applied to the object. This works when a StyleGroup has been specified at the report sheet level and when there are styles in the StyleGroup that can be applied to the object. • Specifies a CSS selector to be applied to the object. Data type: String
Suppressed	Specifies whether to show the object in the design area and in the report result. All formulas and calculations will be skipped if the property is set to true. Data type: Boolean **Note:** When both the Invisible and Suppressed properties of an object are set to true, Suppressed has the higher priority.
SuppressedIfNoRecords	Specifies whether to display the object in the report result when no record is returned to its parent data container. Data type: Boolean
SwapGroup	Specifies to display values from different data fields by switching data between the category and series axes, the category and value axes of a chart. By default, this property is set to false, which means no switch will take place in the chart. • **Switching between the category and series axes** If you create a chart by adding fields to all of its axes, and set SwapGroup to true, data on the category and series axes will be switched. • **Switching between the category and value axes** If you create a chart by adding one field to the category axis and several to the value axis, and set SwapGroup to true, data on the category and value axes will be switched. Data type: Boolean **Note:** If the chart has only one field on the category axis and one on the value axis, even though the property value is set to true, there will be no switch take place on the two axes.
TOCAnchor	Specifies whether to include the object in the TOC Browser for the report. Data type: Boolean
Width	Specifies the width of the object, in inches or centimeters. Data type: Float
X	Specifies the horizontal coordinate of the top left corner of the object, relative to its parent container. Data type: Float
XHyperLink	Specifies whether to add a hyperlink that refers to another report or a website to the X axis labels. Data type: String
XHyperLinkTarget	Specifies the target window or frame in which to display the content the XHyperlink property specifies. • **<Server Setting>** Loads the linked file according to setting of the Pop Up New Window for Links option in the Customize Profile panel of the Profile dialog in JReport Server. • **New Window** Loads the linked file into a new window. The window is not named. • **Whole Window** Loads the linked file into the full browser window. • **Same Frame** Loads the linked file into the same frame as the link. • **Parent Frame** Loads the linked file into the parent frame of the frame that contains the link. • **Other Frame** Loads the linked file into some other specified frame. If the frame name does not exist, the linked file will be loaded into a new window. Data type: String
Y	Specifies the vertical coordinate of the top left corner of the object, relative to its parent container. Data type: Float
ZHyperLink	Specifies whether to add a hyperlink that refers to another report or a website to the Z axis labels. Data type: String
ZHyperLinkTarget	Specifies the target window or frame in which to display the content the ZHyperlink property specifies. If the specified frame name does not exist, the linked document will be loaded into a new window. • **<Server Setting>** Loads the linked file according to setting of the Pop Up New Window for Links option in the Customize Profile panel of the Profile dialog in JReport Server. • **New Window** Loads the linked file into a new window. The window is not named. • **Whole Window** Loads the linked file into the full browser window. • **Same Frame** Loads the linked file into the same frame as the link. • **Parent Frame** Loads the linked file into the parent frame of the frame that contains the link. • **Other Frame** Loads the linked file into some other specified frame. If the frame name does not exist, the linked file will be loaded into a new window. Data type: String

Getting report metadata information

Dataset information can be got by the method getDataSet(), which returns the dataset object. Then the following report metadata information can be got from the dataset:

- Basic information
- Query information
- Business view information
- Business/Report cube information (Deprecated)
- Data buffer information

Getting basic information

- **getName()**
  Returns the dataset name.
- **getDataSourceName()**
  Returns the name of the data source used to build the data component.
- **getResourceType()**
  Returns the resource type. 1 means query, 2 means business view, and 3 means business/report cube.
- **getBlName()**
  Returns the information of the query, imported SQL, stored procedure, user defined data source, business view, or business/report cube used to build the data component.
- **getMaxRecords()**
  Returns the maximum number of records you want to display for all of the data containers using this dataset.
- **setMaxRecords(int maxRecords)**
  Sets the maximum number of records.
- **getDataDriver()**
  Returns the path of the cached query result file that is used as the data resource.
- **setDataDriver(String dataDriver)**
  Sets the path of the cached query result file.
- **getSecurityPolicyName()**
  Returns the security policy name that is applied to the data component.
- **setSecurityPolicyName(String securityPolicyName)**
  Sets the security policy name.

Getting query information

The following methods are used to get information of the query, imported SQL or other data objects that are used to create the dataset.

- **getQueriableName()**
  Returns the name of the query, imported SQL, stored procedure, or user defined data source.
- **getDataFieldNames()**
  Returns a collection of data field mapping names that the dataset uses.

Getting business view information

The following methods are used to get the business view information that is used to create the dataset.

- **getBVName()**
  Returns the business view name.
- **getQueriableName()**
  Returns the name of the query, imported SQL, stored procedure, or user defined data source that is used to create the business view.
- **getElementNames()**
  Returns a collection of the qualified names of the business view elements that the dataset uses.
- **getDisplayName(String qualifiedName)**
  Returns the display name of the business view element that has the qualified name.
- **getMappingName(String qualifiedName)**
  Returns the mapping name of the queriable object data field used to create business view element that has the qualified name.

Getting business/report cube information (Deprecated)

The following methods are used to get the business/report cube information that is used to create the dataset.

- **getCubeName()**
  Returns the business/report cube name.
- **getQueriableName()**
  Returns the name of the query, imported SQL, stored procedure, or user defined data source that is used to create the business/report cube.
- **getElementNames()**
  Returns a collection of the qualified names of the business/report cube elements that the dataset uses.
- **getDisplayName(String qualifiedName)**
  Returns the display name of the business/report cube element that has the qualified name.
- **getMappingName(String qualifiedName)**
  Returns the mapping name of the queriable object data field used to create business/report cube element that has the qualified name.

Getting data buffer information

- **getMaxPageNumber()**
  Returns the maximum number of pages in the data buffer.
- **setMaxPageNumber(int maxPageNumber)**
  Sets the maximum number of pages.
- **getRecordsPerPage()**
  Returns the number of records in each page in the data buffer.
- **setRecordsPerPage(int recordsPerPage)**
  Sets the number of records in each page.

•

Getting a Catalog API instance

The method public CatalogAPI getCatalogAPI() is used for getting a Catalog API instance.

•

Getting objects and object information

- **getDBFields(String handle)**
  Gets DBFields that can be used in the report and returns a mapping name array.
- **getControlFields(String handle, String name)**
  Gets fields that control property values at runtime.
- **getFormulae(String handle)**
  Gets formulas that can be used in the report and returns a mapping name array.
- **getSummaries(String handle)**
  Gets summaries that can be used in the report and returns a mapping name array.
- **getParameter()**
  Gets parameters in the catalog and returns a mapping name array.
- **getQueries()**
  Gets queries in the catalog and returns a mapping name array.
- **getQueryName(String handle)**
  Gets the query name of a report.
- **getProcedures()**
  Gets stored procedures in the catalog and returns a mapping name array.
- **getSections(String handle, boolean visible), getSections(String handle)**
  Gets handles of sections in a report.
- **getGroups(String handle)**
  Gets handles of groups in a report.
- **public GroupInfo[] getGroupInfo(String handle)**
  Gets the group information in a report.
- **public ReportSortInfo getReportSortInfo(String handle)**
  Gets the sort information of a report.
- **public SortInfo[] getSortInfo(java.lang.String handle)**
  Gets the sort information of a group.
- **getSQLs()**
  Gets the SQLs in the catalog.
- **getUDS()**
  Gets the UDSs in the catalog.
- **getViews()**
  Gets the VIEWs in the catalog.
- **getObjectType(String handle)**
  Gets the type information of an object.
- **getObjectInfo(String handle)**
  Gets the handle of an object in a report.

Parameters

- handle - Report node handle.
- name - The property name.
- visible - A Boolean type parameter which specifies that the section is visible.

Getting the instance name and class type of an object

- **getInstanceName(String handle)**
  Gets the instance name of an object, and returns the instance name. If it fails, it will return null.

- **getClassType(String handle)**
  Gets the class type of an object and returns one of the class type values that are defined in the Designer class. If an error or a warning occurs, it will return UNKNOWN.

Parameter

- handle - Object handle.

•

Moving objects

The method public boolean changeZOrder(String objectHandle, int zorderType) is used to move an object to front/back, or forward/backward.

Parameters

- objectHandle - Handle of the object whose z-order will be moved.
- zorderType - Direction to which the object will be moved.

•

Getting editing information

- **getWarning()**
  Gets and returns the warning message. If the operation is successful, it will return null.
- **clearWarning()**
  Clears the warning message.
- **getError()**
  Gets and returns the error message. If the operation is successful, it will return null.
- **clearError()**
  Clears the error message.
- **clearMsg()**
  Clears all warning and error messages.
- **setLog(OutputStream log, String encoding)**
  Sets the log to get error, debug and other information.
- **writeLog(String msg)**
  Writes all messages to the log.
- **closeLog() throws IOException**
  Closes the log.

•

Exiting from the editing status

- **exit()**
  Exits from the editing status, releases all the resources and saves all reports and the catalog.

- **quit()**
  Quits the editing status without saving and releases all resources. It will return false if it fails, otherwise true.

•

Concurrency control

Most of the methods in JReport Design API can also be used when programming with JReport Server Design API. There are however, some methods which are unique to JReport Server Design API. When programming with it, you should use these methods by using the parameters listed in MultiUserCatalogAPI and MultiUserDesigner, which can be found in the JReport Javadoc stored in `<install_root>\help\api`. The methods have parameter UID, and they are used to access private catalog resources while another user is editing the catalog in MultiUserCatalogAPI with the same UID; however, all the standard Catalog and Design API methods without the UID are still supported.

Notes:

- The number of concurrent reports and users will depend on your license.
- A user does not mean a person but means a session. Concurrent users mean concurrent sessions.
- When using JReport Server Design API, you must make sure that each user can only edit one report and that each report can only be edited by one user at the same time.
- MultiUserDesigner allows multiple users to edit a report; however, it only supports report sets with a single tab as designed in V7 or earlier JReport Designer. It can not be used with report sets that contain multiple tabs.
- For details about JReport Server Design API special methods, see the JReport Javadoc in `<install_root>\help\api`.

Illustrative examples

JReport provides you with some sample programs which demonstrate how to create and edit reports using the Design API in `<insatll_root>\help\samples\APIDesign`. They are:

- **TestDesignEditInvoice.java**
  An example of how to modify an existing page report template.
- **TestDesignGraph.java**
  An example of designing a report with a chart.
- **TestDesignInvoice.java**
  An example of designing a report containing a banded object.
- **TestDesignSubreport.java**
  An example of designing a report with a subreport.
- **TestInsertCrossTabIntoBanded.java**
  An example of inserting a crosstab into a banded object.
- **TestInsertCrossTabIntoReport.java**
  An example of designing a report with a crosstab.
- **TestMutiUserDesign.java**
  An example of creating two reports at the same time.

The following are three specific examples of using the Design API:

- Example 1: TestDesignInvoice.java
- Example 2: TestMutiUserDesign.java
- Example 3: TestDesignEditInvoice.java

Example 1: TestDesignInvoice.java

TestDesignInvoice.java in `<insatll_root>\help\samples\APIDesign` is primarily designed for creating a report with the methods provided by the Design API. Here is an explanation of how TestDesignInvoice.java works.

- **Creating an instance of Designer**
  ```
  desg = new Designer(path, cat);
  ```
  The constructor of Designer has two parameters: path and name. They are respectively the path and name of the catalog with which the report will be created.

- **Setting log if necessary**
  ```
  desg.setLog(new FileOutputStream(log), "8859_1");
  ```
  The output stream log points to the path and file in which to write log messages. Parameter "8859_1" is the encoding for the messages.

- **Getting handles**
  ```
  handles = desg.getHandles(report, Designer.PAGE);
  ```
  This method gets all handles of PAGE type and returns a handle array. Parameter report is the parent handle, and Designer.PAGE is a class type value which is defined in the Designer class.

- **Creating a report set**
  ```
  // Create a report set. Use the name from the -name parameter.
  reportset = designer.createReportSet(name);
  ```
  This method creates a new report and returns its handle. Parameter name is the name of the report that is to be created.

- **Adding a connection**
  ```
  // Register the existing query "WorldWideSales" to this report set.
  dataset = designer.addDataset(reportset, "Data Source 1",
  "WorldWideSales","dsWorldWideSales", "",Designer.DATASET_TYPE_QUERY);
  ```
  This method adds a dataset to the report. The dataset must exist in the referenced catalog.

- **Adding a report tab**
  ```
  // Add a report tab named "Invoice" to the report set.
  report = designer.addReport(reportset, "Invoice");
  ```
  This method creates a new report and returns its handle. Parameter name is the name of the report that is to be created.

- **Adding a banded object**
  ```
  // Insert the banded object for the invoice report
  banded = designer.insertBandedObject(reportbody, true, true, true);
  ```
  This method adds a banded object into our report tab.

- **Inserting a group panel**
  ```
  gGroupInfo ginfo =new GroupInfo( "Customers_Customer ID", GroupInfo.DESCENDING );
  groupCust = designer.insertSectionGroup( handles[0], ginfo );
  ```
  This method inserts a new group panel and returns its handle. Parameter handles[0] is the parent node handle. The newly inserted object will be in this node. Customers_Customer ID is the mapping name to group by.

- **Inserting a label**
  ```
  handle = desg.insert(header, Designer.LABEL, "Label1");
  ```
  This method inserts a label named label1 into the group header. Parameter header is the parent handle, the newly inserted object will be in this node. desg.LABEL is the class type that you want to insert, and Label1 is the object name to be inserted.

- **Inserting DBField**
  ```
  handle = desg.insert(header, Designer.DBFIELD, "DBField1", "Orders_Order ID");
  ```
  This method inserts a DBField named DBField1 into the group header. Parameter header is handle of the parent node; Designer.DBFIELD is class type that is to be inserted, and Orders_Orders ID is the mapping name of the database field.

- **Inserting a formula, parameter and summary**
  To insert a formula, parameter or summary, use the same method as for inserting a DBField. Note that the formula, parameter or summary that is to be inserted should already exist in the catalog.

  For example, in the sample program, it is made sure that the parameter PToday, formula customeraddress12, and summary Sum_total9 already exist in the catalog by the following:

  ```
  handle = desg.insert(header, Designer.PARAMETER, "Parameter1", "PToday");
  handle = desg.insert(header, Designer.FORMULA, "formula1", "customeraddress12");
  handle = desg.insert(footer, Designer.SUMMARY, "summary1", "Sum_total9");
  ```

- **Inserting an image, box and line**
  To insert an image, box or line, use the same method as for inserting a label.

  For example, in the sample program, you can insert the image, box and line as follows:

  ```
  handle = desg.insert(header, Designer.IMAGE, "Image1");
  handle = desg.insert(report, Designer.BOX, "Box1");
  handle = desg.insert(report, Designer.LINE, "Line1");
  ```

- **Setting the property value**
  To set the property value, you should provide three parameters: object handle, property name and property value.

  In the sample program, you can set the property value as follows:

  ```
  desg.set(handle, "Width", 1.74f);desg.set(handle, "Height", 0.35f);desg.set(handle,
  "Text", "Invoice");desg.set(handle, "Alignment", desg.LEFTCENTER);desg.set(handle,
  "Foreground", new Color(0x900000));desg.set(handle, "Bold", true);desg.set(handle,
  "FontSize", 0.31f);
  ```

- **Exiting and closing all reports**
  ```
  desg.exit();
  ```
  This method ends the editing status, releases all resources and saves all of the reports and the catalog.

- **Compiling and running the sample program**
  To compile and run the sample program, you should add report.jar, sac.jar, log4j-1.2.8.jar and JREngine.jar with their path into the class path.

  For example, you can use the following command to compile the sample program:

  ```
  C:\JReport\Designer\help\samples\APIDesign>javac -classpath "C:\JReport\Designer\lib
  \report.jar;C:\JReport\Designer\lib\sac.jar;C:\JReport\Designer\lib\JREngine.jar;C:
  \JReport\Designer\lib\log4j-1.2.8.jar" TestDesignInvoice.java
  ```

  When you run the sample program, you should provide two or three parameters. If you want to use two parameters, they should be catalog path and catalog name, and the command line should be:

  ```
  C:\JReport\Designer\help\samples\APIDesign>java -Dreporthome="C:\JReport\Designer" -
  classpath "C:\JReport\Designer\lib\jrengine.jar;C:\JReport\Designer\lib\sac.jar;C:
  \JReport\Designer\lib\resource_en_US.jar;C:\JReport\Designer\lib\report.jar;C:\JReport
  \Designer\lib\log4j-1.2.8.jar;C:\JReport\Designer\lib\hsqldb.jar" TestDesignInvoice -
  path=C:\JReport\Designer\Demo\Reports\TutorialReports -catalog=JinfonetGourmetJava.
  cat -report=TestInvoice.cls
  ```

  If you want to use three parameters, they should be catalog path, catalog name and log file with a full path name, and the command line should be:

  ```
  C:\JReport\Designer\help\samples\APIDesign>java -Dreporthome="C:\JReport\Designer" -
  classpath "C:\JReport\Designer\lib\JREngine.jar;C:\JReport\Designer\lib\sac.jar;C:
  \JReport\Designer\lib\resource_en_US.jar;C:\JReport\Designer\lib\report.jar;C:\JReport
  \Designer\lib\log4j-1.2.8.jar;C:\JReport\Designer\lib\hsqldb.jar" TestDesignInvoice -
  path=C:\JReport\Designer\Demo\Reports\TutorialReports -catalog=JinfonetGourmetJava.
  cat -report=TestInvoice.cls -log=C:\JReport\Designer\logs\designer.log
  ```

  Here, it is assumed that JReport Designer has been installed to `C:\JReport\Designer`, and the current directory when you execute these commands is `C:\JReport\Designer\help\samples\APIDesign` (location of the sample programs). After running the program, you will find a newly created file TestInvoice.cls in `C:\JReport\Designer\Demo\Reports\TutorialReports`.

Example 2: TestMutiUserDesign.java

TestMutiUserDesign.java in `<insatll_root>\help\samples\APIDesign` is primarily designed for creating two reports at the same time using the methods provided by the Design API. The MultiUserDesign API can only create page reports in the V7 and earlier format. These run in later versions of JReport but can not include multiple page tabs, each template can only contain a single report tab.

With this sample program, two reports are created:

- **Creating the first instance of MultiUserDesigner**
  ```
  desg = new MultiUserDesigner(path, cat, userInfo);
  ```

  The constructor of MultiUserDesigner has three parameters: path, name and userInfo. They are respectively the path and name of the catalog with which the reports will be created, and the user ID and license key which are provided by JReport.

- **Creating the first report**
  ```
  report = desg.create(name, "invoice", User)
  ```

  This method creates a new report and returns its handle. Parameter name is the name of the report that is to be created, Invoice is the name of query in the catalog file SampleReports.cat on which the report is created, and User is the user who creates the report.

- **Getting handles**
  ```
  handles = desg.getHandles(report, Designer.PAGE);
  ```

  This method gets all handles of PAGE type and returns a handle array. Parameter report is the parent handle, and Designer.PAGE is the class type value defined in the Designer class.

- **Inserting a parameter, Dbfield, formula**
  To insert a parameter, Dbfield or Formula. The following methods are examples of the special methods mentioned before. User is the name of each concurrent user.

  ```
  handle = desg.insert(User, header, Designer.PARAMETER, "Parameter1", "PToday")
  handle = desg.insert(User, header, Designer.DBFIELD, "DBField1", "Orders_Order ID")
  handle = desg.insert(User, header, Designer.FORMULA, "formula1", "customeraddress12")
  ```

- **Creating the other report**
  User is the name of each concurrent user. There are two users in this demo.

  ```
  DesignerReport DESInvoice = new DesignerReport("Myinvoice", test.desg, "designer1");
  DesignerReport DESregionsalesgraph = new DesignerReport("Myregionsalesgraph", test.
  desg, "designer3");
  ```

Example 3: TestDesignEditInvoice.java

TestDesignEditInvoice.java in `<insatll_root>\help\samples\APIDesign` shows how to modify a report using the methods of the Design API. It demonstrates how to change the image in the report TestInvoice.cls.

Here is an explanation of how TestDesignEditInvoice.java works with the methods provided by the Design API.

- **Creating an instance**
  ```
  desg = new Designer(path, cat);
  ```
  The constructor of Designer has two parameters: path and name. They are respectively path and name of the catalog with which the report will be created.

- **Setting log**
  ```
  desg.setLog(new FileOutputStream(log), "8859_1");
  ```
  The output stream log points the path and file in which to write log messages. Parameter "8859_1" is the encoding for the messages.

- **Opening an existing report**
  ```
  report = desg.open(name);
  ```
  This method opens an existing report corresponding to the name specified, and returns the handle of the report.

- **Getting handles**
  ```
  handle = desg.getHandles(report, Designer.IMAGE, -1)[0];
  ```
  This allows the handle of the object to be modified.

- **Changing the property value**
  ```
  // change the image "coffee.gif" to "CoffeeLogo.gif"
  desg.set(handle, "PictureName", "CoffeeLogo.gif");
  // change the image position
  desg.set(handle, "X", 4.67f);
  // change the image width and height
  desg.set(handle, "Width", 1.70f);
  desg.set(handle, "Height", 0.81f);
  ```

- **Exiting and closing all reports**
  ```
  desg.exit();
  ```

- **Compiling and running the sample program**
  To compile and run the sample program, you should add report.jar, sac.jar, log4j-1.2.8.jar and JREngine.jar with their path into the class path.

  For example, you can use the following command to compile the sample program:

  ```
  C:\JReport\Designer\help\samples\APIDesign>javac -classpath "C:\JReport\Designer\lib
  \report.jar;C:\JReport\Designer\lib\sac.jar;C:\JReport\Designer\lib\JREngine.jar;C:
  \JReport\Designer\lib\log4j-1.2.8.jar" TestDesignEditInvoice.java
  ```

  When you run the sample programs, you should provide two or three parameters. If you want to use two parameters, they should be catalog path and catalog name. The command line should be:

  ```
  C:\JReport\Designer\help\samples\APIDesign>java -Dreporthome="C:\JReport\Designer" -
  classpath "C:\JReport\Designer\lib\JREngine.jar;C:\JReport\Designer\lib\sac.jar;C:
  \JReport\Designer\lib\report.jar;C:\JReport\Designer\lib\log4j-1.2.8.jar;C:\JReport
  \Designer\lib\hsqldb.jar;" TestDesignEditInvoice -catalog=JinfonetGourmetJava.cat -
  report=TestInvoice.cls
  ```

  If you want to use three parameters, they should be catalog path, catalog name and log file with a full path name. The command line should be:

  ```
  C:\JReport\Designer\help\samples\APIDesign>java -Dreporthome="C:\JReport\Designer" -
  classpath "C:\JReport\Designer\lib\JREngine.jar;C:\JReport\Designer\lib\sac.jar;C:
  \JReport\Designer\lib\report.jar;C:\JReport\Designer\lib\log4j-1.2.8.jar;C:\JReport
  \Designer\lib\hsqldb.jar;" TestDesignEditInvoice -path=C:\JReport\Designer\Demo
  \Reports\TutorialReports -catalog=JinfonetGourmetJava.cat -report=TestInvoice.cls -
  log=C:\JReport\Designer\logs\design.log
  ```

  Here, it is assumed that JReport Designer has been installed to `C:\JReport\Designer`, and that the directory that is current when you execute these commands is `C:\JReport\Designer\help\samples\APiDesign` (location of the sample programs). After running the program, you will find a modified file TestInvoice.cls in `C:\JReport\Designer\Demo\Reports\TutorialReports`.

JReport Result Viewer Bean

JReport Result Viewer Bean enables you to:

- Show the report result files (.rst extension).
- Export the report result files to multiple file formats (HTML, PDF, XLS, RTF, Postscript, Text, CSV, Applet, E-mail) and to printer.

Reference: See also the JReport Javadoc jet.bean.JRResultViewer class in `<install_root>\help\api`.

The following sections provide a detailed introduction about JReport Result Viewer Bean:

- Installing the Result Viewer Bean
- Result Viewer Bean properties
- Programming with the Result Viewer Bean

Installing the Result Viewer Bean

When you install JReport Designer, which by default is located in `C:\JReport\Designer`, JReport Result Viewer Bean is also installed at the same time. After installation, you will have the main components in `<install_root>` as follows:

- **lib\JRResultViewer.jar***
  Jar file for the Result Viewer Bean.
- **lib\log4j-1.2.8.jar***
  Jar file used for the JReport Engine logging system.
- **lib\activation.jar**
  Jar file for exporting to mail.
- **lib\mail.jar**
  Jar file for exporting to mail.
- **lib\resource_en_US.jar**
  Archive file used when exporting to XML.
- **lib\xml-apis.jar, xercesImpl.jar**
  Jar files used when exporting to XML.
- **lib\itext_1.5.4.jar**
  Jar file used when exporting to PDF.
- **lib\iTextAsian.jar**
  Jar file used for the Asian fonts.
- **lib\poiHSSF_151.jar**
  Jar file used when exporting to Excel.
- **SwingDesigner.jar**
  If RPTExecutor UDO has been inserted into a report, to view the correct result, this jar file should be added to the class path that starts the Result Viewer Bean.
- **Readme.htm**, **License.htm**
  Release information.
- **help\samples\APIViewResult\TestResultBean.java**
  A sample program to demonstrate how to display the result.

Notes:

- * means that these files are required when you call the Result Viewer Bean.
- Other files without * should be added to your class path to call the Result Viewer Bean.

  For example, when you want to use the Result Viewer Bean to export a result file to mail, you should add mail.jar and activation.jar with a valid path to your class path, and when you want to export the result file to XML, you should add resource_en_US.jar with a valid path to your class path.

Result Viewer Bean properties

JReport Result Viewer Bean has four properties: reportName, showInfoLevel, logFile and loadReport. The following methods are used to set the values of these properties:

- setReportName ("ReportName")
- setTempPath("theTemporaryPath");
- setShowInfoLevel (vDebug/vError/vOff)
- setLogFile ("LogFileName")
- setLoadReport ("Yes/No")

Parameters

- ReportName - Specifies the name of the JReport result file that is to be viewed. It contains both the path information and the file name.

  > **Note:** Here the report name is the result file name with an .rst extension (NOT the report file .cls).

- theTemporaryPath - The temporary file path. When the bean runs, the temporary files will be written in this path. You should set the full path of the temporary files in this method as:

  ```
  bean.setTempPath("c:\\temp");
  ```

  If you do not set the temporary path, the default path is the directory from where the Bean will be run.

- vDebug/vError/vOff - These are used for showing and turning off debug and error information. LogFileName specifies where the debug, error and other information will go. LogFileName contains both the path and the file name. If no path or file is specified, the default will be used.

- Yes/No - Indicates whether or not to start to view the result of the report (the file specified by ReportName).

Programming with the Result Viewer Bean

JReport Result Viewer can be used to view local JReport result files generated by JReport Engine. You can show the report result file (.rst) and export the report result to multiple file formats and to printer, by calling JReport Result Viewer Bean in three different ways: by JReport Designer, by JReport Server, or by JReport Engine Bean.

The following topics describe how to program with JReport Result Viewer Bean:

- Preparations
- Using the Result Viewer Bean
- An illustrative example

Preparations

Before programming with the Result Viewer Bean, you need to import the classes of the Result Viewer Bean by adding the following import statements to the beginning of your program. The first import statement is added to import the bean class:

```
import jet.bean.*;
import java.awt.*;
import java.awt.event.*;
import javax.swing.*;
```

Then, you need to create an instance.

To run with the JReport Result Viewer Bean, you should first create an instance of it. The constructor of the Result Viewer Bean has no parameter. After the object has been created, you should then set the report name and temp path (if not set, the current path will be used).

```
bean = new JRResultViewer();
bean.setShowInfoLevel(bean.vDebug | bean.vError);
bean.setReportName("c:\\test\\customerlist.rst");
bean.runReport("yes");
```

With the JReport Result Viewer Bean object, you can show the report result and export it to other formats or to printer. You do not need to create a bean for each report. The Result Viewer Bean provides many useful methods.

Using the Result Viewer Bean

After making the required preparations, you can now use the Result Viewer Bean to perform tasks.

The following topics detail how to run with the Result Viewer Bean to achieve certain goal:

- Exporting report result to different file formats
- Printing reports
- Getting the page orientation of a report page
- Browsing report pages
- Zooming reports
- Getting field results by listening to the click event
- Setting the report communicator
- Setting the user input stream
- Exiting from the Result Viewer Bean

Exporting report result to different file formats

There are several methods with which JReport Result Bean can export a result file (.rst extension) to different file formats (.htm, .pdf, .txt, and so on):

- **Export to HTML**
  - exportToHtml(string HTMLFileName)
  - exportToHtml(string HTMLFileName, HtmlParameter Parameter)
  - exportToHtml(string HTMLFileName, boolean bChartApplet, boolean isMultiFile, boolean bUsingTable, boolean bHyperlink, boolean bPageNum, boolean bAbsolute, int iBrowser)
  - exportToHtml(string HTMLFileName, boolean bChartApplet, boolean isMultiFile, boolean bUsingTable, boolean bHyperlink, boolean bPageNum, boolean bAbsolute, int iBrowser, Drilldownfile)
  - exportToHtml(string RSTFileName, string HTMLFileName, boolean bChartApplet, boolean isMultiFile, boolean UsingTable, boolean Hyperlink, boolean PageNum, boolean bAbsolute, int iBrowser, boolean bincdrilldownfile, boolean noMargin)
  - exportToHtml(string rstFileName, string HTMLFileName, boolean bChartApplet, boolean isMultiFile, boolean bUsingTable, boolean bHyperlink, boolean bPageNum, boolean bAbsolute, int iBrowser, boolean bincdrilldownfile, boolean noMargin, int resolution)
- **Export to Page Report Result**
  - exportToDhtml(string HTMLFileName)
  - exportToDhtml(string RSTFileName, string HTMLFileName, boolean bChartApplet, boolean isMultiFile, boolean bUsingTable, boolean bHyperlink, boolean bPageNum, boolean bAbsolute, int iBrowser, boolean bincdrilldownfile, boolean noMargin)
  - exportToDhtml(string HTMLFileName, boolean bChartApplet, boolean isMultiFile, boolean bUsingTable, boolean bHyperlink, boolean bPageNum, boolean bAbsolute, int iBrowser, boolean bincdrilldownfile, boolean noMargin, int resolution)
  - exportToDhtml(string HTMLFileName, boolean bChartApplet, boolean isMultiFile, boolean bUsingTable, boolean bHyperlink, boolean bPageNum, boolean bAbsolute, int iBrowser)
  - exportToDhtml(string HTMLFileName, boolean bChartApplet, boolean isMultiFile, boolean bUsingTable, boolean bHyperlink, boolean bPageNum, boolean bAbsolute, int iBrowser, boolean bincdrilldownfile)
  - exportToDhtml(string HTMLFileName, boolean bChartApplet, boolean isMultiFile, boolean bUsingTable, boolean bHyperlink, boolean bPageNum, boolean bAbsolute, int iBrowser, boolean bincdrilldownfile, boolean noMargin)
- **Export to PDF**
  - exportToPdf(string PDFFileName)
  - exportToPdf(string FileName, PDFParameter Parameter)
  - exportToPdf(string PDFFileName, boolean noMargin)
  - exportToPdf(string PDFFileName, boolean noMargin, boolean isSimPrintMode)
  - exportToPdf(string PDFFileName, boolean noMargin, boolean isSimPrintMode, int isCompressImage)
  - exportToPdf(string RSTFileName, string PDFFileName, boolean noMargin, boolean isPrintMode, int isCompressImage, boolean isTransparent)
  - exportToPdf(string PDFFileName, boolean noMargin, boolean isSimPrintMode, int isCompressImage, boolean isTransparent, boolean bToc, boolean bDrill, string password)
  - exportToPdf(string PDFFileName, boolean noMargin, boolean isSimPrintMode, int isCompressImage, boolean isTransparent, boolean bToc, boolean bDrill)
- **Export to Text**
  - exportToText(string TextFileName, boolean isNormalText)
  - exportToText(string TextFileName, TextParameter Parameter)
  - exportToText(string TextFileName, boolean isNormalText, int udchWidth, int unchHeight)
  - exportToText(string TextFileName, boolean isNormalText, boolean isRepeat, char delimiter)
  - exportToText(string TextFileName, boolean isNormalText, boolean isRepeat, char delimiter, boolean compress)
  - exportToText(string TextFileName, boolean isNormalText, boolean isRepeat, char delimiter, boolean compress, boolean quoteMark)
  - exportToText(string TextFileName, boolean isNormalText, boolean isRepeat, char delimiter, int udchWidth, int udchHeight, boolean compress, int resolution, boolean quoteMark, boolean hasHeadfoot)
  - exportToText(string TextFileName, boolean isNormalText, boolean isRepeat, char delimiter, int udchWidth, int udchHeight, boolean compress, int resolution, boolean quoteMark, boolean hasHeadfoot, boolean forWindows)
  - exportToText(string TextFileName, boolean isNormalText, int udchWidth, int udchHeight, boolean compress, int resolution)
  - exportToText(string TextFileName, boolean isNormalText, int udchWidth, int udchHeight, boolean compress, int resolution, boolean quoteMark)
- **Export to Excel**
  - exportToXls(string XLSFileName)
  - exportToXls(string XLSFileName, ExcelParameter Parameter)
  - exportToXls(string XLSFileName, int isWordWrap)
  - exportToXls(string XLSFileName, int isWordWrap, boolean isBiff8)
  - exportToXls(string XLSFileName, int isWordWrap, boolean isBiff8, int param)
- **Export to PS**
  - exportToPS(string PSFileName)
  - exportToPS(string PSFileName, boolean noMargin)
  - exportToPS(string PSFileName, PSParameter Parameter)
- **Export to RTF**
  - exportToRtf(string RTFFileName)
  - exportToRtf(string RTFFileName, boolean noMargin)
  - exportToRtf(string RTFFileName, RTFParameter Parameter)
- **Export to XML**
  - exportToXML(string XMLFileName, boolean isMultiFile, boolean isOnlyData)
  - exportToXML(string XMLFileName, boolean isDataOnly, string xsdFile)
  - exportToXML(string XMLFileName, boolean isDataOnly)

Parameters

- HTMLFileName - The name of the HTML format result file with a full path.
- bChartApplet - A boolean value which specifies whether or not the chart in an HTML file is an applet. The default value is true.
- isMultiFile - A boolean value which specifies whether or not HTML is generated to a single file or to multiple files. The default value is true.
- bUsingTable - A boolean value which specifies whether or not to use HTML table format when exporting HTML files. The default value is false.
- bHyperLink - A boolean value which specifies whether or not a hyperlink is generated. The default value is true.
- bPageNum - A boolean value which specifies whether or not to generate page numbers. The default value is true.
- bAbsolute - A boolean value which specifies whether the font size is absolute or relative. The default value is true.
- iBrowser - An int value which specifies whether or not the web browser is IE. 0 means IE. The default value is 0.
- Drilldownfile - A boolean value which specifies whether or not to generate the drilldown HTML files in the file list. The default value is false.
- RSTFileName - The name of the result file (.rst extension) that is to be converted to the required format.
- PDFFileName - The name of the PDF format result file with a full path.
- isSimPrintMode - A boolean value which specifies whether or not to use Simulated Printing Mode (true) or Standard Mode (false) (To PDF). By checking the Standard Mode, the exporting speed will be much improved. The default value is false.
- isCompressImage - An int value which specifies the compression ratio of the image in a report (To PDF). The values range from 1 to 100. The default value is 20.
- isTransparent - A boolean value which specifies whether or not to keep the background of the GIF image transparent.
- TextFileName - The name of the Text format result file with a full path. true/false is to set whether this text file is in normal Text format or standard Data format. The standard Data format text file is a text file where each row represents a single record in the report. It can be used as a text datasource for exchanging data with other applications.
- isNormalText - A boolean value which specifies whether or not the text file is in normal text format. When it is set to false, the output format is in standard data format. The default value is true.
- isRepeat - A boolean value which specifies whether or not the same contents are repeated. The default value is false.
- delimiter - A character used in SDF (Standard Data Format) format. The delimiter will only be used when the "isNormalText" parameter is set to false. The default delimiter is a blank space.
- compress - A boolean value which specifies whether or not to remove empty lines or redundant delimiters in the exported text file. The default value is false.
- XLSFileName - The name of the Excel format report result file with a full path.
- PSFileName - The name of the Postscript format result file with a full path.
- RTFFileName - The name of the Rich Text Format file with a full path.
- noMargin - A boolean value which specifies whether or not to generate a file without a page margin. The default value is false.
- chWidth - User defined char width used in plain text format. It would be ignored if the value is 0 or less than 0. When exporting to Text, the default value is -1.
- chHeight - User defined char height which is used in plain text format. It would be ignored if the value is 0 or less than 0. When exporting to Text, the default value is -1.
- XMLFileName - The XML file name.
- isMultiFile - A boolean value which specifies whether XML is generated to a single file or to multiple files.
- isOnlyData - A boolean value which specifies whether or not only the database column information is exported to an XML file. Otherwise, all the report details, including all the properties will be exported. The default value is false.
- parameter - A parameter contains some special settings for exporting to different file formats.
- resolution - An int value which is used to specify resolution for different exported file formats.
- password - A string value which is used to set the password when exporting to a PDF file that contains a password.
- bToc - A boolean value which specifies whether or not to generate TOC for the PDF file.
- bDrill - A boolean value which specifies whether or not to apply drill-down function to the exported PDF file.
- quoteMark - A boolean value specifies whether or not quote characters will be kept in standard format text file.
- hasHeadfoot - A boolean value specifies whether or not page header and footer will be contained when exporting to TEXT.
- forWindows - A boolean value specifies whether the CR-LF in Windows convention or the LF in Unix convertion will be used as the end-of-line character.
- udchWidth - An int value which is specified for each unit of the horizontal density between columns.
- udchHeight - An int value which is specified for each unit of the vertical density between columns.
- isWordWrap - An int value which specifies a word wrap setting option.
  - KEEPORIGIN - Keeps all the original settings of each component's Word Wrap property as specified in the report.
  - NOWORDWRAP - Disables the Word Wrap property for all components.
  - WORDWRAPALL - Enables the Word Wrap property for all components.
- isBiff8 - A boolean value specifies whether or not to export reports to a Data Format file, which means only the report data will be exported without format.
- param - An int value specifies whether or not to include drawing objects in the exported Excel file. If isBiff8 is set to true, drawing objects will not be included in the result file, because they cannot be exported to Data Format files, which means only the report data will be exported without format.
  - NOEXPORTSHAPE - Drawing objects will not be exported to Excel files.
  - EXPORTSHAPE - Drawing objects will be exported to Excel files.
- XSDFile - A string value indicates the schema file on which the exported XML file is based. If it is set to null, a new XSD file will be output.

Notes:

- Method exportToHtml(string HtmlFileName) will use the method exportToHtml(string HTMLFileName, boolean bChartApplet, boolean isMultiFile, boolean bUsingTable, boolean bHyperlink, boolean bPageNum, boolean bAbsolute, int iBrowser) using default parameter values, such as: exportToHtml ("HtmlFileName", false, true, false, true, true, true, JRResultViewer.IE).
- Method exportToText(string TextFileName, boolean isNormalText) will use the method exportToText (string TextFileName, boolean isNormalText, boolean isRepeat, char delimiter) using default parameter values such as: exportToText("TextFileName", isNormalText, false, ' ').

Printing reports

There are five methods provided for printing a report.

- printReport()
- printReport(Object printJob, Object pageFormat, boolean bInteractive, boolean bInBackground, boolean bUseJDK11)
- printReport(Object printJob, Object pageFormat, boolean bInteractive, boolean bInBackground, boolean bUseJDK11, boolean bSeparateLargePage)
- printReport(java.lang.Object job, boolean bInteractive, java.lang.Object pf, boolean bInBackground, JReportPrintListener jreportprintlistener, int useJDK, java.lang.String jobName, java.util.Vector attributeInfo, java.lang.String printer)
- getPrintStatus()

Parameters

- bInteractive - A Boolean value. If you set it to false, the Print Setup dialog will not be shown.
- bInBackground - A Boolean value. If you set it to true, the Print dialog will not be shown.
- bUseJDK11 - A Boolean value. If you set it to true, it will use JDK1.1.x to print the report. Otherwise if you set it to false, it will use the JDK that you are running to print the report. The JDK 1.1 printing method is quick in speed but the printing quality cannot be guaranteed. The JDK 1.2 method will provide a satisfactory result, even for a .gif file, but it is slow in speed.
- bSeparateLargePage - A Boolean value. If the report page size is larger than the print paper size, and you want to separate a large report page during printing, you should set this property to true. If the print page size you choose is less than the report page size, the report will be printed in multiple pages serially. This means a large report will be separated automatically. For JDK1.2 or later, after you have set this parameter to true, the Page Format dialog will be shown. In this dialog, you can choose the page format in which you want to print. This page size will then be used for printing. For JDK1.1.x, however, you must first change the page format in the Print dialog (click the Property button, then choose the page format).
- job - The print job of this print action. It is an instance of PrintJob or PrinterJob.
- bUseJDK - An Integer value. If you set it to 0, it will use the method JDK1.1 to print the report. If you set it to 1, it will use the method JDK1.2 to print the report. If you set it to 2, it will use the method JDK1.4 to print the report. The JDK 1.1 printing method is quick in speed but the printing quality cannot be guaranteed. However, the JDK 1.2 method will return you with a satisfactory result, even for a gif file, but it is slow in speed. The method JDK1.4 provides further print options, such as specifying the paper tray.
- attributeInfo - A vector with String elements. Each string has been defined in interface PrintConfig which is located in package jet.bean. For example, you can implement PrintConfig and call its fields as shown below:

  ```
  attributeInfo.add(TRAY_MANUAL);
  attributeInfo.add(ORIENTATION_PORTRAIT).
  ```

Example 1

With JDK1.2 or later, if you use the JDK1.2 printing method (boolean bUseJDK11), you can call the printReport method as shown below:

```
PrinterJob printJob=PrinterJob.getPrinterJob();
bean.printReport(printJob, printJob.defaultPage(), false, false, false);
```

Or, if you do not want to use getPrinterJob, you can simply call:

```
bean.printReport(null, null, false, false, false);
```

Example 2

With JDK1.2 or later, if you use the JDK1.1 printing method, you are best recommended not to use getPrintJob, instead you can call:

```
bean.printReport(null, null, false, false, true);
```

Notes:

- The first method printReport() is called to start printing. It will take the third method using default parameter values such as:
  printReport(null, null, true, false, false, false)
- The second and third method allows you to pass the parameters - printJob, pageFormat, bInteractive, bInbackground, bUseJDK11 and bSeparateLargePage.
- The third printing method (with the bSeparateLargePage parameter) is not recommended. It may cause some unexpected errors, for example, you may get an extra blank page.
- After printing has started, the method getPrintStatus() can be used to get the print job status. Returned values of getPrintStatus() are:
  "No print job"
  "Busy"
  "OK"
  "No report load"
  Other error messages if there are errors.

Getting the page orientation of a report page

The method getPageOrientation() is used to get the orientation of the report page. You should call this method after calling the runReport method. The return value is an int value, and can be LANDSCAPE or PORTRAIT. If it is -1, it makes no sense, and is incorrect. For example, you may have called it before calling runReport method.

Browsing report pages

The methods listed here are also available to JReport Result Viewer Bean.

- The methods getCurrentPageNumber() and getTotalPageNumber() are used to return the current page number and the total number of pages of the current report onto the screen.
- The method int gotoPage(int pageNumber) is used to display the page specified by PageNumber on the screen. If PageNumber is out of the range of the current report, the first page (if the pageNumber is less than 0), or the last page will be shown on screen. The returned value is the current page number.
- The methods firstPage(), prevPage(), nextPage() and lastPage() are used to show the specified page on the screen.

The return value of the above methods is a string, and can be:

- "OK"
- "No more page"
- Other error messages such as "No report load".

Zooming reports

If you want to zoom in or out of a report, there is a method named zoom(ratio) that will meet this requirement.

Parameter

- ratio - Integer type, representing a percentage value.

Example 1

If you want to enlarge the report by two times the original size, you can use the method in this way:

```
zoom(200)
```

Example 2

If you want to reduce the report size by half, you can use the method in this way:

```
zoom(50)
```

Getting field results by listening to the click event

When you click a field in the viewer, you can get the entire record of the field. The interface in package jet.thinviewer named ClickActionListener can implement this function. This interface is defined as:

```
interface ClickActionListener
{
        void clickOnField(jet.connect.Record currentRecord);
}
```

If you have a class which wants to listen to the click event of a field, this class can implement this interface. Then, after you have instantiated a bean, you can add the ClickActionListener.

Note: This action must be implemented before running the report. Finally after the report has been shown and you click on a field, the ClickActionListener's method clickOnField will be called, and you will then get the values of all columns in that field from the parameter currentRecord.

Setting the report communicator

The method SetReportCommunicator(Communicator) is used to set the report communicator.

Parameter

- communicator - The Communicator here is a class defined by JReport. After users set the communicator, the viewer will load the report from this communicator, not the result file.

You are recommended to use getReportCommunicator() in JReport Engine Bean, and use the method setCommunicator() in JReport Result Viewer Bean. That is,

```
resultViewerBean.setCommunicator(engineBean.getReportCommunicator());
```

Setting the user input stream

To view the result file with JReport Result Viewer, you will usually set the report name to a physical .rst file, or if you want to set the stream of the .rst file, you can use the methods below:

- setInputStream(InputStream ins, boolean zip)

  Parameters

  - ins - The user input stream.
  - zip - Whether the result is compressed or not. To see whether the result is compressed or not, you can call the method below:

- boolean isZipStream(InputStream ins)

  Parameter

  - ins - The user input stream.

  Return

  - true - Compressed stream.
  - false - Uncompressed stream.

Exiting from the Result Viewer Bean

The method exit() should be called when you want to quit from JReport Result Viewer Bean. It will perform some clean up work, such as deleting temporary files. It is important to free up the resources of the result file, so remember to follow this method.

An illustrative example

There is a sample program called TestResultBean.java in `<install_root>\help\samples\APIViewResult` included in the JReport Server and JReport Designer installation.

The properties for the report are very simple, and you will only need to set the ReportName. Note that here the report name is the report result .rst file generated by JReport Engine. It is not a .cls file. By calling the runReport and show() methods, the report result will be displayed.

The report result can be exported to other formats such as HTML, PDF, RTF or to a printer. You will need to use the appropriate methods and parameters.

To run TestResultBean.java, you can use the following command to compile the sample program first (assume that JReport Designer has been installed to `C:\JReport\Designer`):

```
C:\JReport\Designer\help\samples\APIViewResult>javac -classpath "C:\JReport\Designer\lib
\JRResultviewer.jar;C:\JReport\Designer\lib\log4j-1.2.8.jar;C:\JReport\Designer\lib
\itext_1.5.4.jar" TestResultBean.java
```

JReport Catalog Bean

JReport Catalog Bean provides a programming API for manipulating and accessing a JReport catalog. A catalog is used for holding all the resources used for building reports in JReport.

JReport Catalog Bean enables you to:

- Create an empty catalog.
- Add objects to a catalog.
- Remove objects from a catalog.
- Get an object description in a catalog.

The following are some specific topics about JReport Catalog Bean:

- Installing the Catalog Bean
- Catalog Bean classes
- Programming with the Catalog Bean

Installing the Catalog Bean

When you install JReport Designer, which by default is located in `C:\JReport\Designer`, JReport Catalog Bean is also installed at the same time. After installation, you will have the main components in `<install_root>` as follows:

- **lib\JREngine.jar***
  Class library that contains the Catalog Bean.
- **lib\log4j-1.2.8.jar***
  Jar file used for the JReport Engine logging system.
- **bin\jslc.dat***
  License control file.
- **bin\report.ini***
  Configuration file.
- **help\samples\APICatalog\TestCatalogBean.java**
  The test tool of the Catalog Bean with a graphics user interface.

Catalog Bean classes

JReport Catalog Bean contains several classes: JRCatalog class, JRCatalogException class and ConnectionDesc class. Read this section to get detailed information.

JRCatalog class

The JReport Catalog is the principal class that manipulates and accesses a catalog. The application creates a JReport Catalog instance, and uses JReport Catalog methods to manipulate the catalog.

JReport Catalog has two properties:

- reportHome - The string that contains the installation path of JReport Catalog Bean.
- catName - The string that contains the full file path of the catalog. This path is used when loading or creating a catalog.

JRCatalogException class

JRCatalogException is the class that describes the error information. This exception is thrown when an error occurs. This exception describes many kinds of errors. The error code in the exception represents the exception type. The error code and error message can be viewed through public means.

ConnectionDesc class

ConnectionDesc is the class that describes the JDBC connection information. This class is used to create/modify connections and get connection information.

Reference: JReport Javadoc jet.bean.JRCatalog, jet.bean.JRCatalogException and jet.universe. ConnectionDesc classes in `<install_root>\help\api`, which includes all the documents generated by Javadoc.

Programming with the Catalog Bean

A catalog is used to contain all resources used for building reports in JReport Designer, and JReport Catalog Bean provides a programming API for manipulating and accessing JReport catalogs. JReport Catalog Bean allows to achieve functions such as, creating an empty catalog, adding objects to a catalog, removing objects from a catalog, and getting object description in a catalog.

Go through the following topics to further understand how to program with JReport Catalog Bean:

- Preparations
- Using the Catalog Bean
- An illustrative example
- Test tool program

Preparations

Before programming with the Catalog Bean, you need to import the Catalog Bean classes by adding the following import statements to the beginning of your program. The first two import statements are added to import the Catalog Bean, and exception classes. The third is added to import the connection description class.

```
import jet.bean.JRCatalog;
import jet.bean.JRCatalogException;
import jet.universe.ConnectionDesc;
```

Then you need to create an instance of the Catalog Bean. The constructor of the Catalog Bean has no parameter. After the object has been created, you should then set the report home to the object.

```
JRCatalog jrCatalog = new JRCatalog();
jrCatalog.setReportHome("c:\\jreport") ;
```

With the Catalog Bean object, you can manipulate and access many catalogs. You do not need to create a bean for each catalog.

You can load an existing catalog or create a new catalog. After a catalog has been created or loaded, you can manipulate it before closing it. The closeCatalog method closes a catalog, releases resource and clears up the temporary file. Also, the program requires using the closeCatalog method when operations on it have been completed.

Using the Catalog Bean

After making the required preparations, you can now use the Catalog Bean to perform tasks.

Creating a catalog

The method newCatalog creates an empty catalog according to the catalog name that is set in catName, and saves it to disk.

When creating a new catalog, the directory of the catalog name should exist and contain no catalog files. You should make sure of this before you start creating the new catalog.

```
Try
{
    jrCatalog.setCatName("c:\\a.cat");
    newCatalog();
}
catch(JRCatalogException e)
{
    System.out.println(e);
}
```

Loading an existing catalog

The method loadCatalog is used to load a catalog according to the catalog name that is set in catName.

```
try
{
    jrCatalog.setCatName("c:\\a.cat");
    loadCatalog();
}
catch(JRCatalogException e)
{
    System.out.println(e);
}
```

Manipulating a catalog

A catalog can be manipulated by the actions of adding, deleting and editing of objects within it. A catalog contains many kinds of objects. These objects include connections, tables, views, synonyms, file queries, stored procedures, queries, WHERE portions, formulas, summaries and parameters.

For example, you can modify the connection, the name, the URL, the user, the password and the driver.

```
try
{
    ConnectionDesc conDesc=new ConnectionDesc();
    conDesc.strName="Demo";
    conDesc.strURL="jdbc:oracle:thin:@myhost:1521:orcl";
    conDesc.strUser="system";
    conDesc.strPassword="manager";
    conDesc.strDriver="oracle.jdbc.driver.OracleDriver";
    jrCatalog.modifyConnection(conDesc);
}
catch(JRCatalogException e)
{
  System.out.println(e);
}
```

Getting information

You can get information about objects in a catalog via some methods.

For example, you can get the connection information via the method getConnectionDesc().

```
Try
{
    ConnectionDesc conDesc = bean.getConnectionDesc();
}
catch(JRCatalogException e)
{
    System.out.println(e);
}
```

Saving a catalog

Changes to a catalog are saved to a disk only when the method saveCatalog is used.

```
Try
{
    jrCatalog.saveCatalog();
}
catch(JRCatalogException e)
{
    System.out.println(e);
}
```

Handling exceptions

When an error occurs, the JRCatalogException is thrown out. Your program will catch the exception and handle the error.

The exception describes the error code and detailed message as follows:

- **Code: 0**
  Message: "System initialize failed."
  Description: The system initialization process failed. The Catalog Bean needs to be initialized to resolve the environment. This error is caused by reasons such as wrong report home, and bad installation.
  Detail: The reason why the initialization failed.
- **Code: 1**
  Message: "The catalog name is null."
  Description: The catalog name is null. The catalog name should be set before the loadReport() or newReport() methods are used.
  Detail: None.
- **Code: 2**
  Message: "Creating the catalog failed, because the specified directory does not exist or there is an existing catalog in that directory."
  Description: Cannot create a catalog in the specified directory, because the directory does not exist or there is a catalog in the directory.
  Detail: None
- **Code: 3**
  Message: "Cannot open the catalog file."
  Description: Cannot open the specified catalog file. The reason is file not found, file corrupt, format error or version mismatch.
  Detail: The reason why the opening of a catalog file failed.
- **Code: 4**
  Message: "Save catalog failed."
  Description: Cannot save the catalog file successfully. The cause is a file writing error, or a formula file save has failed.
  Detail: The reason why the saving of a catalog failed.
- **Code: 5**
  Message: "No catalog loaded."
  Description: The loadCatalog() or newCatalog() methods should be used before other methods, and closeCatalog() should be used in the end.
  Detail: None
- **Code: 6**
  Message: "The name of the WHERE portion is invalid or already exists."
  Description: The name specified is invalid or has already been used by another WHERE portion.
  Detail: None
- **Code: 7**
  Message: "The specified WHERE portion does not exist."
  Description: The specified WHERE portion does not exist.
  Detail: None.
- **Code: 8**
  Message: "A catalog can contain only one connection."
  Description: A catalog can contain only one connection, and this error occurs when adding a second connection.
  Detail: None.
- **Code: 9**
  Message: "The name of the connection is invalid or already exists."
  Description: The name specified is invalid or has already been used.
  Detail: None
- **Code: 10**
  Message: "Connection failure."
  Description: The JDBC connection failed when trying to create a connection to the database.
  Detail: The message of SQLException or ClassNotFoundException.
- **Code: 11**
  Message: "The name of the file query (customer SQL) is invalid or already exists."
  Description: The name of file query is either invalid or already exists.
  Detail: None.
- **Code: 12**
  Message: "Cannot create the file query."
  Description: Cannot create file query object.
  Detail: The reason why the file query cannot be created.
- **Code: 13**
  Message: "There is no connection in the catalog."
  Description: There is no connection in the catalog. If you are deleting or modifying a connection, the problem is that no such connection exists. If you are adding a file query, you cannot create it without a connection.
  Detail: None
- **Code: 14**
  Message: "The specified file query doesn't exist."
  Description: The specified file query does not exist.
  Detail: None.
- **Code: 15**
  Message: "Unknown Error."
  Description: The unexpected error.
  Detail: None
- **Code: 16**
  Message: "Invalid user ID or password"
  Description: The user ID or password is invalid.
  Detail: None
- **Code: 17**
  Message: "Only Tables or Views can be added into catalog."
  Description: The table type users specified is not supported.
  Detail: None
- **Code: 18**
  Message: "Close connection failure."
  Description: When closing connection, an error occurred.
  Detail: None

An illustrative example

In the following program, the Catalog Bean will be used to change the connection of a catalog:

```
try
{
// create catalog bean
JRCatalog jrCatalog = new JRCatalog();
// set report home
jrCatalog.setReportHome( "c:\\jrcbean");
// load catalog
jrCatalog.setCatName("c:\\a.cat">);
loadCatalog();
// modify connection
ConnectionDesc conDesc = new ConnectionDesc();
conDesc.strName = "Demo";
conDesc.strURL = "jdbc:oracle:thin:@myhost:1521:orcl";
conDesc.strUser = "System";
conDesc.strPassword = "Manager";
conDesc.strDriver = "oracle.jdbc.driver.OracleDriver";
jrCatalog.modifyConnection(conDesc);
// save catalog
jrCatalog.saveCatalog();
// close catalog
jrCatalog.closeCatalog();
}
catch(JRCatalogException e)
{
System.out.println(e);
}
```

Test tool program

TestCatalogBean.java located in `<install_root>\help\samples\APICatalog`, is a test tool for JReport Catalog Bean. It is primarily designed and built for you to test all the methods of JReport Catalog Bean.

The test tool program has a user interface, and all the functions of the Catalog Bean can be invoked using the menu items. Depending on success or failure of each test, you will get either a return result or an exception message respectively.

To run the JReport Catalog Bean test tool program, you should:

1. Compile the source code of TestCatalogBean.java. For example, JReport Designer is installed in `C:\JReport\Designer`, you can compile the it using the following command:

   ```
   C:\JReport\Designer\help\samples\APICatalog>javac -classpath "C:\JReport\Designer
   \lib\report.jar;C:\JReport\Designer\lib\sac.jar;C:\JReport\Designer\lib\JREngine.
   jar;" TestCatalogBean.java
   ```

2. Run the test tool:

   ```
   C:\JReport\Designer\help\samples\APICatalog>java -Dreporthome="C:\JReport
   \Designer" -classpath "C:\JReport\Designer\lib\JREngine.jar;C:\JReport\Designer\lib
   \sac.jar;C:\JReport\Designer\lib\report.jar;C:\JReport\Designer\lib\log4j-1.2.8.
   jar;C:\JReport\Designer\lib\hsqldb.jar" TestCatalogBean
   ```

   After activating the program, a window will be displayed.

3. Before starting other commands, you should first set the home path and the catalog name.

   Select **File** > **Set Home Directory** to set the home path and **File** > **Set Catalog Name** to set the catalog name.

4. Then, you can create a new catalog with the specified catalog name, or load an existing catalog in the same way.

5. The test tool also provides the source code of each method, and you can copy and paste it into your program from the text window.

JReport Exit functions

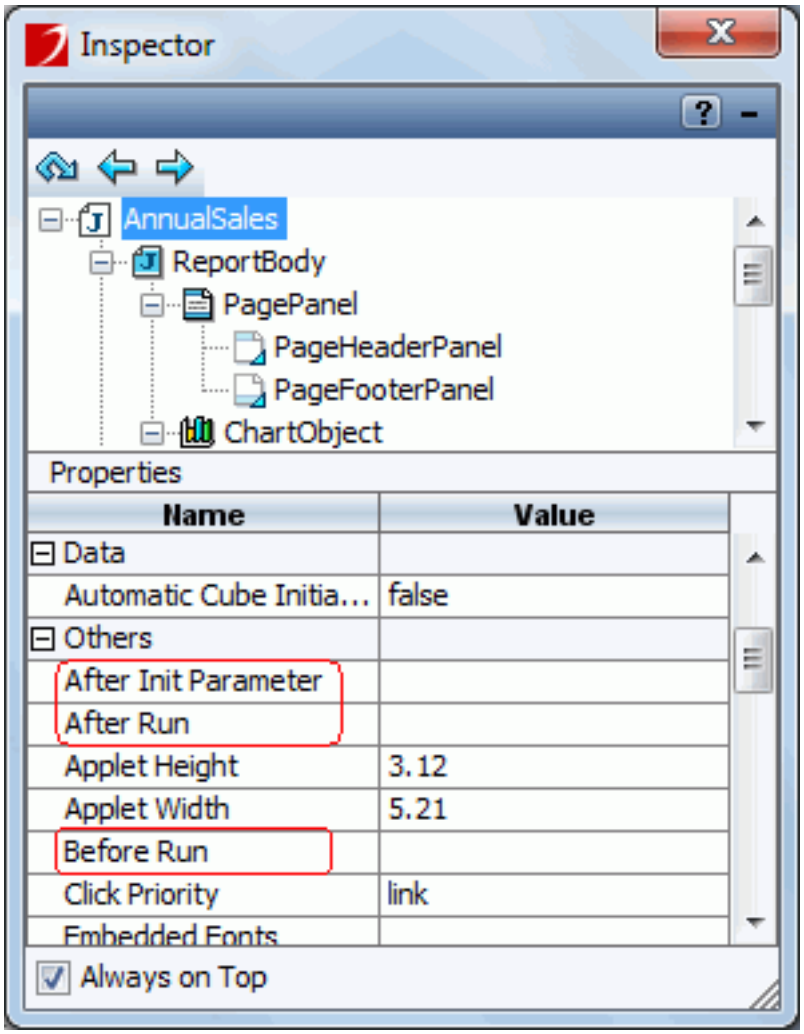
JReport has developed three Exit functions: After Init Parameter, After Run and Before Run. These functions enable you to develop an action to be called before, during and after the process of running a report. A status will then be returned. If the returned status is true, the JReport Engine will go on running. If false, the JReport Engine will stop at this point.

The following topics explain the JReport Exit functions in detail:

- Properties
- Executive interfaces
- Using the Exit functions

Properties

Open a report in JReport Designer. In the Report Inspector, select the node that represents the report tab and you will find the three Exit function properties: After Init Parameter, After Run and Before Run.



- **After Init Parameter**
  When running the report, after you enter parameters in the parameter dialog, this function allows you to call a program developed by yourself.

- **After Run**
  Allows you to call a program developed by yourself after running a report.

- **Before Run**
  Calls a program developed by yourself just before running a report.

Reference: For API information, see JReport Javadoc jet.util package and jet.exception package in `<install_root>\help\api`.

Executive interfaces

We provide three interfaces for you to implement:

jet.util.JURLExecutor

This interface provides the following method, which is used to get the status of the JReport Engine:

public boolean exec(String[] params) throws JURLExecuterException

- **Parameter**
  params - Array of strings, each element is defined and parsed by yourself. You can enter the PARAMETERs in the Report Inspector. If no PARAMETER is entered, params.length will be zero.
- **Return**
  Boolean value. If true, the JReport Engine will continue running. If false, JReport Engine will stop at this point.

jet.util.JURLEngineExecutor

This interface provides the following method, which is used to get the status of the JReport Engine:

public boolean exec(String[] params, EngineExecutor executor) throws JURLExecutorException

- **Parameter**
  params - Array of strings, each element of which is defined by yourself and also parsed by yourself. You can enter the PARAMETERs in the Report Inspector. If no PARAMETERs are entered, then params. length is zero.
  executor - Executor of EngineExecutor, which is another interface JReport provides for exporting a report to HTML, TXT or PDF format within the Exit functions.
- **Return**
  boolean value. If true, the JReport Engine will continue running. If false, the JReport Engine will stop at this point.

jet.util.EngineExecutor

This interface provides three methods:

- **public boolean exportToHtml(String htmlFileName, boolean bChartApplet, boolean isMultiFile, boolean bUsingTable, boolean bHyperlink, boolean bPageNum, boolean bAbsolute, int iBrowser)**

  This method is used to export the report result to HTML format.

  - **Parameter**
    HtmlFileName - Name of the HTML format result file with full path.
    bChartApplet - A Boolean value which specifies whether or not a chart in an HTML file is an applet.
    isMultiFile - A Boolean value which specifies whether or not the HTML is generated to one or multiple files.
    bUsingTable - A Boolean value which specifies whether or not to use the HTML table format for exporting HTML files.
    bHyperLink - A Boolean value which specifies whether or not to generate hyperlinks.
    bPageNum - A Boolean value which specifies whether or not to generate page number.
    bAbsolute - A Boolean value which specifies whether or not the font size is absolute or relative.
    iBrowser - An int value which specifies whether or not the web browser is IE.
  - **Return**
    Boolean value for exporting status. If true, the report has been successfully exported and vice versa.

- **public boolean exportToPdf(String pdfFileName)**

  This method is used to export the report result to PDF format.

  - **Parameter**
    PdfFileName - The name of the PDF format result file with a full path.
  - **Return**
    Boolean value for exporting status. If true, the report has been successfully exported and vice versa.

- **public boolean exportToText(String textFileName, boolean isNormalText, boolean isRepeat, char delimiter)**

  This method is used for exporting the report result to text format.

  - **Parameter**
    TextFileName - Name of the TEXT format result file with full path, where "true/false" sets whether or not the text file is of normal text format or of standard data format. A standard data format text file is a text file where each row represents a single record in a report. It can be used as a text data source for exchanging data with other applications.
    isNormalText - A Boolean value which specifies whether or not a text file is in normal text format.
    isRepeat - A Boolean value which specifies whether or not the contents will be repeated.
    delimiter - A character which is be used in SDF (Standard Data Format). The delimiter will only be used when the "true/false" parameter is set to false. Delimiters can be ',' (the CSV format) or any other character. The default is ' ' (a blank), which is the SSV format.
  - **Return**
    Boolean value for exporting status. If true, the report has been successfully exported and vice versa.

Reference: For API information, see JURExecutor, JUREngineExecutor and EngineExecutor interfaces in the JReport Javadoc jet.util package located at `<install_root>\help\api`.

Using the Exit functions

This section provides you with two examples to further understand the Exit functions.

Example 1: Using parameters in the Exit functions

In this example, we provide a simple application which will run at all the three times, Before Run, After Init Parameter, and After Run.

1. Develop your Java files to implement the methods. In the example, copy the following code and save it as **testa.java** to `C:\JReport\Designer\help` (here it is assumed that you have installed your JReport Designer to the default directory).

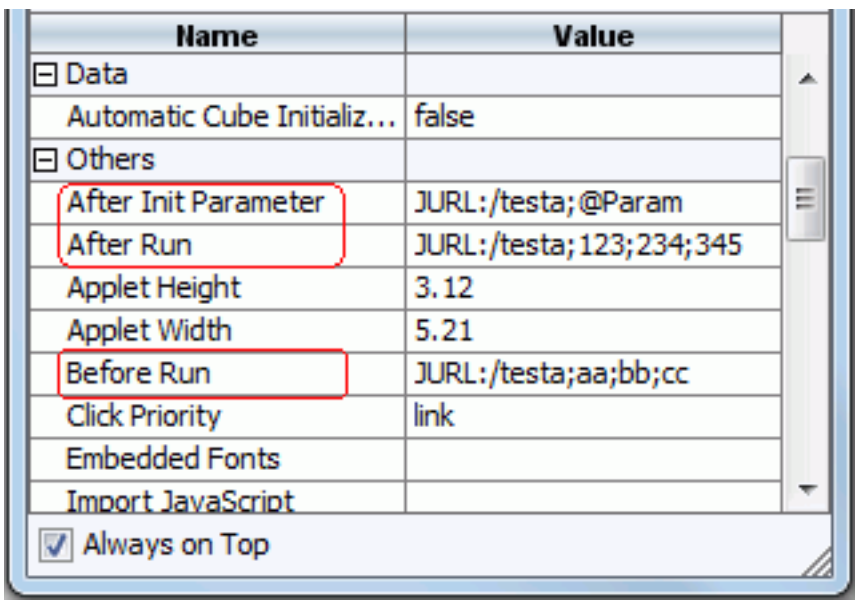
```
import java.io.*;
import java.awt.*;
import java.net.*;
import jet.util.*;

public class testa implements JURLExecuter
{
        public boolean exec(String[] params)
        {
                System.out.println("testa : ");
                for (int i = 0; i < params.length; i++)
                {
                        System.out.print("\"" + params[i] + "\",");
                }
                System.out.println();
                return true;
        }
}
```

2. Compile testa.java to generate the class file testa.class in `C:\JReport\Designer\help`.
3. Modify the batch file setenv.bat in `C:\JReport\Designer\bin` by appending the path `C:\JReport\Designer\help` into the batch file's ADDCLASSPATH variable:

   `set ADDCLASSPATH=%JAVAHOME%\lib\tools.jar;`**`C:\JReport\Designer\help;`**

4. Start JReport Designer, then open a report with parameter.
5. In the Report Inspector, select the node that represents the report tab and specify the class and give parameters for the three functions, for example:



Name	Value
⊟ Data	
Automatic Cube Initializ...	false
⊟ Others	
After Init Parameter	JURL:/testa;@Param
After Run	JURL:/testa;123;234;345
Applet Height	3.12
Applet Width	5.21
Before Run	JURL:/testa;aa;bb;cc
Click Priority	link
Embedded Fonts	
Import JavaScript	
☑ Always on Top	

JURL:/ - The standard format, and followed by the class name.
testa - The class name in this example.
aa;bb;cc and **123;234;345** - The string type parameters of the class, which are divided by semicolons.
Param - A predefined JReport parameter used in the current report, which is introduced by the symbol @.

As shown above, two types of Exit function parameters are supported, they are:

- String type parameters, such as aa, C:\ and so on. In this case, the string will be passed directly to the Exit function.
- JReport predefined parameters used in the current report, which are introduced by the @ symbol. In this case, the parameter value will be passed to the Exit function.

6. Run the report, the report engine will call the class that you specified in the Report Inspector.

Example 2: Using formulas in the Exit functions

JReport allows you to use formulas in the Exit functions. To do this, you can use the Data Container Link function which can return value of any formula to a parameter. And using the parameter in the Exit functions, you will get the value of the formula in the Exit functions.

1. Create a formula and insert it into a report. It should be put in the child data container.
2. Create a parameter with the same data type of the formula, leaving other options blank, and then insert it into the parent data container.
3. Right-click the parent data container and click **Data Container Link** on the shortcut menu.
4. In the Data Container Link dialog, click the **Return Value** tab, add the formula in the Field in Child Data Container to the Return Value box, then click **OK**.

   Now, you have finished passing value of the formula to a parameter.
5. Use this parameter in the Exit functions with the method described in Example 1, and then the final value of formula will be returned in the Exit functions.

Notes:

- When a JReport parameter is used as the parameter of the Before Run function, no JReport parameter value will be returned, and the returned value will be the JReport parameter name with the @ symbol.
- For the JReport parameter that are used as Exit function parameters, they must be used in the report. Otherwise, no value will be returned.
- If you want to publish the report to JReport Server and run it there, you should publish both the report and the catalog file to JReport Server. However, do not forget to add the path to the `-classpath` option in the batch file that starts JReport Server.
- Using formulas in Exit functions can also be realized by using returning values in subreport. Thus, you need to pass value of a formula in the subreport to a parameter in the primary report.

Accessibility

JReport supports accessibility attributes and a built-in accessible JReport Server console. The implementation standard is based on HTML specification 4.01 http://www.w3.org/TR/WCAG10-HTML-TECHS/ and information on Section 508 Standards: http://www.section508.gov and http://www.access-board.gov.

Making HTML format report results accessible

When designing a report in JReport Designer, you can add the accessibility related HTML attributes to the report elements in order to make the HTML format result more readable and accessible. Those attributes are provided in the Accessibility category of the Report Inspector.

To add accessibility to a report in the HTML format result:

1. Predefine necessary accessibility attributes when designing the report.

   When a report element is selected, the corresponding accessibility attributes are available in the Report Inspector. For details about the accessibility attributes for each report element, see Properties in the Report Inspector.

2. Enable Section 508 compliant output when exporting the report to HTML format.

   In the HTML export option UI, check the option **Section 508 Compliant Output**.

   If you only want to convert table/crosstab components into HTML data table in the HTML format report result, check the option **Use HTML Data Table**.

   The above two options are available on all HTML export option UIs in JReport Designer and JReport Server.

The accessible version of JReport Server

JReport Server enables the disabled users to visit the accessible version with simplified functionality to read reports by reader agent or other assistive tools. In the accessible version, reports are displayed in the HTML format with accessibility attributes, and table/crosstab components are output as HTML data tables.

JReport Server provides an option Use Accessible Version for setting the accessible version of JReport Server as the default portal UI for an end user after the user logs in. The default port for accessible version is 8888 which is also the default port for JReport Console page in the normal version. That is to say, the Use Accessible Version option controls switching between normal version and accessible version of JReport Server UI when logging onto port 8888. By default this option is unselected and end users are directed to the JReport Console page in normal version.

True Type Fonts

Users might find that well designed reports in JReport Designer become truncated when running in JReport Server on a different platform. The reason being that various platforms use different font systems, and by mapping the fonts to a different platform by JDK, the report result is shown with a discrepancy. To solve this problem, Java provides a True Type Font (TTF) administrative system, which allows the fonts to be published together with a report into the runtime environment independent of the platform fonts.

In addition, in cases where end users do not have the fonts used in a report, JReport provides a PDF embedded fonts solution. That is, the fonts are embedded in the PDF file, and end users do not need to install the fonts to their local machines.

Notes:

- When you design reports that will be run on a platform different to where they have been designed, it is best to use TTF fonts.
- JReport uses the JDK TTF, however the JDK has a problem in printing reports with rs_song.ttf (some characters are incorrect). You are therefore recommended to use other Chinese TTF fonts such as SimSun.ttf, PMingLiu.ttf and LF_FangSong.ttf.

The following topics describe how to use TTF in your reports.

Designing reports with TTF

To design reports with TTF, follow the steps below:

1. Copy True Type Font files `*.ttf` to the `<install_root>\font` subdirectory. You can find the TTF fonts in your system's Font directory or from the Internet.
2. Running JReport Designer. In the design area, click on the drop-down arrow of the font list on the toolbar, and scroll down to the bottom part of the list. You will then find the fonts with an ***** before their name. These are the TTF fonts you have published.
3. Preview the report result and save the report until you are satisfied with the report design.

Publishing TTF to the production environment

If you want to publish report files that use True Type Fonts to other operating systems, you will need to place the relevant font files (other files in the font path) in the `<install_root>\font` directory of the operating system of where you want to publish the report. Make sure that you copy or ftp (in binary mode) the .ttf files to the directory.

Delivering the result in Excel/RTF/HTML/TXT

For files of HTML/RTF/Excel formats, they will find the fonts from the system font directory. If you want to deliver the report result in the above formats to end users, you should choose the TTF fonts that are commonly used on Windows systems for your reports. If you have to use fonts that do not exist in a client system, you can choose the embedded TTF font solution for a good print result in PDF format.

Delivering the result in PDF

To view PDF reports with other character sets, for example, Chinese fonts, you will need to install special language packages. Also, if you design reports with fonts that are not within Acrobat's internal fonts, they will be mapped differently. The above two cases will cause trouble to end users when viewing or printing a report with Acrobat Reader. However, differing from HTML and RTF, PDF allows the embedding of the font description within a file. JReport therefore provides the embedded TTF font solution for a PDF result. You will find it extremely helpful in the following cases:

- When you select TTF fonts to design a report, and these fonts are not within the client's existing font systems.
- When you have used the TTF and Acrobat Reader has to be updated with the language extension package to view the result, but you do not want to trouble the end user with it.

To take advantage of the PDF embedded TTF font feature, follow the steps below:

1. Select the TTF to design reports as specified previously.
2. Open the report, go to the Report Inspector, and in the node representing the report, find a property called Embedded Fonts. The drop-down list will show all the TTF fonts used by the report. You can select multiple fonts by holding down the Ctrl or Shift key.
3. Save the report and export it to a PDF file. The selected fonts will be embedded in the exported PDF file.

By embedding TTF fonts, the file size will increase a little. However, it is still of reasonable size to be transferred over a network.

Applying Bold and Italic properties to TTF

JReport supports the Bold and Italic properties when you design report with TTF. When you select a TTF font and set the Bold/Italic property to true, JReport will first automatically search the `<install_root>\font` directory to check all the TTF files related with the TTF font you have selected previously. If there is such a TTF file holding the Bold/Italic property, JReport will use the font style defined in this file to match your setting and make the Bold/Italic function take effect; if not, JReport will find the specified font in your platform fonts and apply the Bold/Italic function directly.

Moreover, when you select an embedded font while using TTF to design reports, and set the Bold/Italic property to true, if JReport find the related TTF file to match your setting, the matched font with the Bold/Italic property will be embedded when you want to export the report result to a PDF file.

Logging

JReport provides a robust, flexible and configurable logging system which is based on log4j version 1.2.8, and supports versions from log4j version 1.2.8 to 1.3alpha7.

The logging system is used for obtaining meaningful and helpful JReport logging information in a convenient way. It is easy to configure and manage. The following are the significant benefits of the JReport logging system:

- It is compatible with your application which is based on log4j.
- It can be configured to output the logging information of different modules separately or to output them all together.
- It supports setting trace and error types' levels separately.

Configuring logs

Log configuration tasks can be achieved in two ways. They are:

- Using the LogConfig.properties file located in `<install_root>\bin` directory.
- Using command options.

  In comparison with earlier versions, the command options used for log configuration have undergone a considerable change. The following are descriptions of these options:

Options	Description
-vDebug	Sets engine log file's trace level to INFO and error level to WARN.
-vError	Sets engine log file's trace level to OFF and error level to ERROR.
-log[:<file>]	Outputs message to the file as specified and uses the -vDebug level.

> **Note:** Any settings made using the command options will override the trace and error type level settings in the LogConfig.properties file. To log on console when using the two commands above: -vDebug or -vError, configure the appender in LogConfig.properties first.

Configuring output log appenders

JReport logging system provides certain appenders to which log content can be output and by default, the output file is located in `<install_root>\logs` directory. For example:

- JReportRollingFile
- JReportDailyRollingFile
- JReportConsole

By default, all log categories use one appender, JReportRollingFile. However, you can also define different output appenders or files for different log categories in the LogConfig.properties file, which is located in `<install_root>\bin`. For example, to add another log appender - JReportConsole for the Designer, edit the LogConfig.properties file as follows:

```
log4j.logger.com.jinfonet.designer=, JReportRollingFile, JReportConsole
...
log4j.appender.RootConsole=org.apache.log4j.ConsoleAppender
log4j.appender.RootConsole.layout=com.jinfonet.util.JRPatternLayout
log4j.appender.RootConsole.layout.ConversionPattern=%m [%t][%p][%d{HH:mm:ss,SSS}]%n
log4j.appender.RootConsole.threshold=ON
log4j.appender.RootConsole.target=System.out
...
```

Logging information using the trace and error type

The JReport logging system enables you to log situations by using the trace and error type. Trace is used for logging something expectable or regular, such as tracing a program workflow, logging runtime information and associated elements. Error is used for logging something unexpected or irregular. For example, when a URL is unreachable, a file does not exist, or a table cannot be found in a data source.

Trace type is divided into four levels, and is ordered according to the amount of information logged, from the least to the most. These levels are OFF, OUTLINE, INFO, and TRIVIAL. Similarly, error type is also divided into four levels, and is ordered according to the amount of information logged, from the least to the most. These levels are OFF, FATAL, ERROR, and WARN.

If you want the least amount of information to be logged, you can set trace level to OUTLINE and error level to ERROR. If you want the most amount of information to be logged, you can set trace level to TRIVIAL and error level to WARN. Setting these to the highest level could affect system performance as well as disk usage.

Trace levels

- **OFF**
  Specifies not to use the trace level.
- **OUTLINE**
  The OUTLINE level specifies an outline of program workflow, and dump global variables, including a single thread, multiple threads, the time for when to begin to fetch data, and success in exporting to the specified result format.
- **INFO**
  The INFO level specifies informational messages that highlight the application progress at a coarse-grained level, and important local variables, such as query, parameter, formula value used when running reports, connection information, and SQL statement.
- **TRIVIAL**
  The TRIVIAL Level specifies fine-grained informational events most useful in tracing an application. Such as the report structure dump, or the result set dump.

Error levels

- **OFF**
  Specifies not to use the Error level.
- **FATAL**
  The FATAL level specifies severe error events that will presumably lead the application to abort, for example, failing to read the valid key, and exceptions that result in feature incompleted.
- **ERROR**
  The ERROR level specifies error events that may allow the application to continue running, such as failing to load a report, failing to find a catalog file, failing to parse a parameter, and failing to create db buffer.
- **WARN**
  The WARN level specifies potentially harmful situations, such as failing to find the resource, or having found invalid query or formula with grammar error when loading a catalog.

Configuration settings - effect on performance

The following are configuration properties which have an effect on performance:

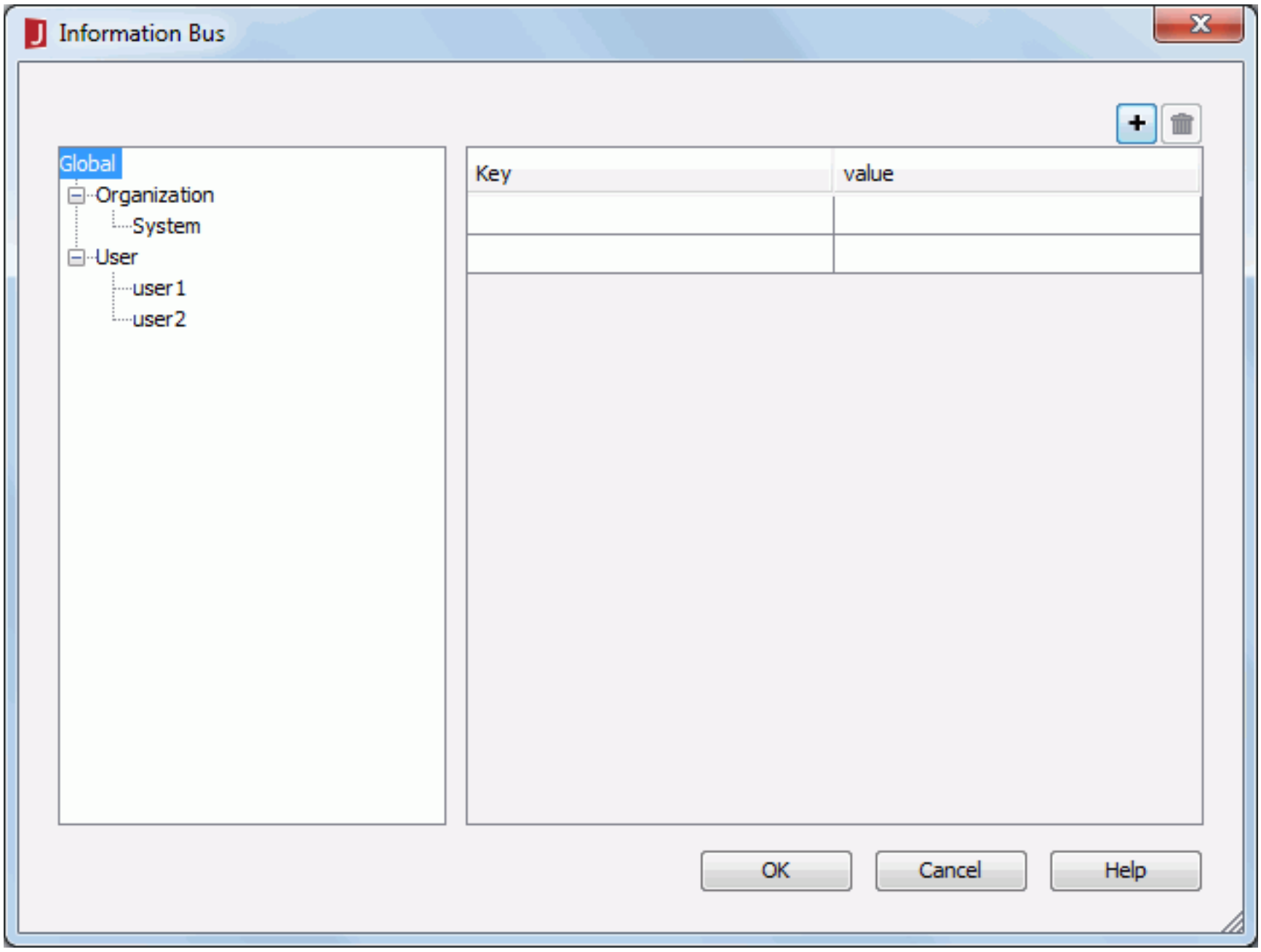
- If the showTitle property is set to true in the LogConfig.properties file, the log file size will increase. However, the effect on performance is negligible.
- The levels of trace and error affect the amount of logging output. The higher the level, the more the output. However, the effect on performance is negligible.
- The use of the following conversion characters in ConversionPattern property results in the slow generation of caller class or location information: %C, %F, %L and %M. Their use should be avoided unless execution speed is not an issue. For information about the ConversionPattern property, see comments in the LogConfig.properties file, which is located in the `<install_root>\bin` directory.

Information Bus

Information Bus is a built-in object in JReport containing information from global level, organization level and user level, which are key-value pairs of information that can be put or got by authorized users when calling APIs or writing formulas.

To edit content in the Information Bus:

1. Click **File** > **Information Bus** on the menu bar to bring out the Information Bus dialog.



2. The users are listed in the left panel. Select one to edit the information. The users here are the same as the users created for member-level security.

   The information from global level, organization level and user level are listed here. System admin can get or put information in all information containers. Organization admin can get or put information in the global level information container, its own organization level information container and the user level information containers that belong to its organization. Organization user can get or put information in the global level information container, the organization level information container it belongs and its own user level information container.

3. Click + to add an information line.
4. Double-click in the Key column to input the key for the information. Then double-click in the Value column to input the value for the information.
5. If needed, click + to add a new information line and repeat the step above to add new information. If any information is not required, select it and click 🗑 to remove it.
6. Click **OK** to apply the settings and close the dialog.

Dialog Reference

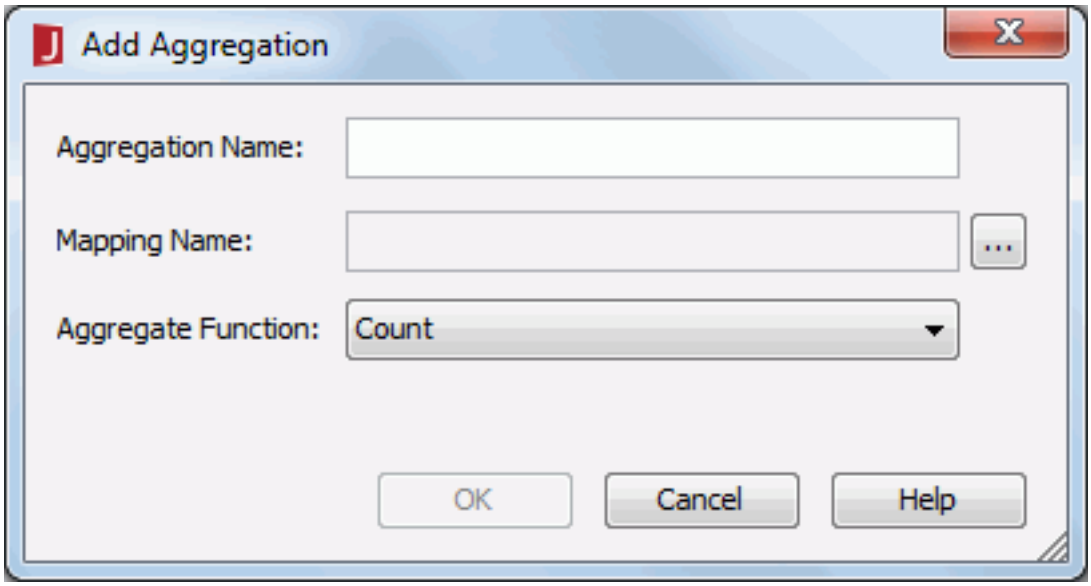
This chapter covers dialogs that you will come across when using *Report Designer*. It mainly provides information about the options available in these dialogs. The purpose is to make the use of *Report Designer* much easier and more convenient.

The following are dialogs included in this chapter:

Add Aggregation dialog
Add Collection dialog
Add Display dialog
Add Font dialog
Add Format dialog
Add Formula Field dialog
Add General Hierarchical Data Source dialog
Add Group dialog
Add Language dialog
Add Role dialog
Add Sorted Frameboxes dialog
Add Sorters dialog
Add Table dialog
Add Table/View/Query dialog
Add User Defined Data Source dialog
Add User dialog
Add View dialog
Add View Element dialog
Advanced dialog
Advanced Export Settings dialog
Aggregate On dialog
Apply Option dialog
Apply Parameter dialog
Apply Sorting dialog
Banded Wizard
Barcode Expert dialog
Bind Data dialog
Bursting dialog
Business View Editor window
Calendar dialog
Category Options dialog
Category Property dialog
Change Property - Web Action Builder dialog
Change Property - Web Behavior dialog
Change Property dialog
Chart Wizard for page report
Chart Wizard for web report/library component
Choose Data dialog
Choose Page Report dialog
Click Priority dialog for page report
Click Priority dialog for web report/library component
Color List dialog
Comparison Function dialog
Conditional Formatting dialog
Connect to Report Server dialog
Create Banded Object wizard
Create Chart wizard for page report
Create Chart wizard for web report/library component
Create Composed Column dialog
Create Crosstab wizard for page report
Create Crosstab wizard for web report/library component
Create Data Mapping File dialog
Create Geographic Map wizard for library component
Create Geographic Map wizard for web report/library component
Create Geographic Map wizard for web report
Create Parameter dialog
Create Query Username and Password dialog
Create Table wizard for page report
Create Table wizard for web report/library component
Crosstab Formula Editor window
Crosstab Wizard for page report
Crosstab Wizard for web report/library component
CSS Editor dialog
CSS Style Definition dialog
Custom Sort dialog
Customize Dataset dialog
Customize Toolbars dialog
Customize Value dialog
Customized Control - Web Action Builder dialog
Customized Control Manager dialog
Customized Function dialog
Data Buffer Information dialog
Data Format dialog
Data Manager dialog
Data Mapping Editor dialog
Dataset Editor dialog
Dataset Filter dialog
Dataset Management dialog
Data Source Driver Manager dialog
Data Source Editor dialog
Display Attributes dialog
Display Type dialog
Download Customized Control dialog
Download from *Report Server* dialog
Edit Additional Value dialog
Edit Aggregation dialog
Edit Business View Security dialog
Edit Color dialog
Edit Conditions dialog
Edit Display Name dialog
Edit Display Name for Component dialog
Edit Filter Control dialog
Edit Filter dialog for page report
Edit Filter dialog for web report/library component
Edit Gradient Color dialog
Edit Group dialog
Edit Hierarchical Data Source dialog
Edit Link dialog for page report
Edit Link dialog for web report/library component
Edit Operation Calling Property dialog
Edit Parameter dialog
Edit Role dialog
Edit Shapes dialog
Edit Slider dialog
Edit Sort and Filter dialog
Edit Style Group dialog
Edit Summary dialog
Edit User Defined Data Source dialog
Edit User dialog
Edit Values dialog
Edit View Element dialog
Encode dialog
Error Parameter Values dialog
Error Values dialog
Excel Option dialog
Export to Excel dialog
Export to Fax dialog
Export to *Report Server* dialog
Export to Mail dialog
Export to PDF dialog
Export to PostScript dialog
Export to RTF dialog
Export to Text dialog
Export to XML dialog
Expression dialog
FO Effects dialog
Filter - Web Action Builder dialog
Filter - Web Behavior dialog
Filter dialog
Filter Options dialog
Filter View dialog
Find dialog
Find/Replace dialog
Flash Parameter dialog
Format Area dialog for library component
Format Area dialog for page report
Format Area dialog for web report
Format Bar dialog for library component
Format Bar dialog for page report
Format Bar dialog for web report
Format Bar Gauge dialog for library component
Format Bar Gauge dialog for page report
Format Bar Gauge dialog for web report
Format Bubble Gauge dialog for library component
Format Bubble Gauge dialog for page report
Format Bubble Gauge dialog for web report
Format Category (Y) Axis dialog for library component
Format Category (Y) Axis dialog for page report
Format Category (Y) Grid/line dialog
Format Cell dialog
Format Dial Gauge dialog for library component
Format Dial Gauge dialog for page report
Format Dial Gauge dialog for web report
Format Dial Gauge dialog for library component
Format Dial Gauge dialog for page report
Format Field dialog
Format Field/Parameter dialog
Format Line dialog for library component
Format Line dialog for page report
Format Line dialog for web report
Format Page dialog
Format Pie dialog
Format Pie dialog for library component
Format Pie dialog for page report
Format Pie dialog for web report
Format Plot/Point dialog
Format Polar dialog
Format Piechart/Piegraph dialog for library component
Format Piechart/Piegraph dialog for web report
Format Piechart/Piegraph dialog for page report
Format Series (Z) Axis dialog
Format Series (Z) Grid/line dialog
Format Scale dialog
Format Surface dialog
Format Text dialog
Format Value (Y) Axis dialog
Format Value (Y) Grid/line dialog
Format Value (Y2) Axis dialog
Format Value (Y2) Grid/line dialog
Format Wall dialog
Formula Editor window
Get JDBC Connection Information dialog
Geographic Map Wizard for library component
Geographic Map Wizard for page report
Geographic Map Wizard for web report
Group Filter dialog
Group Options dialog
Hierarchical Banded Wizard
HTML Option dialog
Image Button dialog
Import from *Report Server* dialog
Import XML Hierarchical Data Source dialog
Incremental Condition dialog
Information Bus dialog
Insert Aggregation dialog
Insert Detail Column dialog
Insert Fields dialog
Insert Filter Control dialog
Insert Group Column dialog
Insert Link dialog for page report
Insert Link dialog for web report/library component
Insert Parameter Form Control dialog
Insert Slider dialog
Insert Summary Column dialog
Insert Summary dialog
Insert Tabular dialog
Insert UDO dialog
Jaxi Options dialog
JSON Connection Wizard
Library Component Setting dialog
Line Pattern List dialog
Link Data Container dialog
Mailing Label Wizard
Merge dialog
Metagalib Connection Wizard
New Catalog dialog
New CSS Style dialog
New Customized Control File dialog
New Dataset dialog
New Data Source dialog
New Library Component dialog
New Page Report dialog
New Page Report Tab dialog
New Summary dialog
NLS Editor dialog
Node Factory List dialog
Open As dialog
Open Report dialog
Optimize Dataset dialog
Options dialog
Page Panel dialog
Page Report Option dialog
Page Setup dialog
Parameter - Web Action Builder dialog
Parameter dialog
Parameter Run Time Order dialog
Parameters dialog
PDF Option dialog
Pick a Color dialog
Pie/Donut Option dialog
Pie-pop Option window
Pieopen Settings dialog
Predefined Filter dialog
Preview Options dialog
Publish Additional Resources dialog
Publish to *Report Server* dialog
Publish to Local Directory dialog
Query Editor window
Query Wizard for Editor dialog
Query Option dialog
Rank Expert dialog
Read Flash Parameter dialog
Receive Message dialog
Record Level Security Information dialog
Reference Table Configuration dialog
Remote Datatype Display Name dialog
Replace Formula dialog
Reset All dialog
RESTful Data Source Options dialog
RTF Option dialog
Save CSS As dialog
Save Library Component As dialog
Save Pre-run Path dialog
Save Report As dialog for page report
Save Report As dialog for web report
Save Style dialog
Scrollable Chart dialog
Search Condition dialog
Search dialog
Search Repository dialog
Security Configuration Setting dialog
Security dialog
Select a Report dialog for page report
Select a Report dialog for web report/library component
Select Area Name dialog
Select Field dialog
Select Folder dialog
Select Group Function dialog
Select N dialog
Select Pre-join dialog
Select Report Tab dialog
Select Slice Group dialog
Select Table dialog
Send Message - Web Action Builder dialog
Send Message dialog
Series Options dialog
Set Grids dialog
Shape Map Area Conditional Formatting dialog
Shape Map Centers Setup dialog
Shape Map Data Binding Wizard
Shape Map Editor window
Show Column dialog
Show Fields dialog
Sign Digital ID dialog
Sort - Web Action Builder dialog
Sort - Web Behavior dialog
Sort By dialog
Sort dialog
Sort Fields By dialog
Special Function dialog
Stored Procedure Parameters dialog
Subreport dialog
Suppress Column Subtotal dialog
Suppress Row Subtotal dialog
Table Type dialog
Table Wizard for page report
Table Wizard for web report/library component
Tabular Wizard
Text Option dialog
Time Zone and Locale Options dialog
UDT Column dialog
Unique Key dialog
User Defined Group dialog
Web Action List dialog
Web Behaviors dialog
Web Options dialog
Web Report Option dialog
Web Service Data Source dialog
Web Wizard - Expert to HTML
Wizard dialog
Windows Media Player Parameter dialog
XML Connection Wizard
XML Option dialog

Add Aggregation dialog

The dialog appears when you click the <Add Aggregation...> item in the Dynamic Resources > Aggregations node in the Resource View panel of the main window, or in the Resources box of the report wizards. It helps you to create a dynamic aggregation which can be used in the current web report or library component.



Aggregation Name

Specifies the name of the aggregation.

Mapping Name

Specifies the field or formula on which the aggregation is based. Click the button ... to select the desired one.

Aggregate Function

Specifies the function for the aggregation. For details about each function, see Math functions.

OK

Creates the aggregation and closes the dialog.

Cancel

Cancels the creation and closes the dialog.

Help

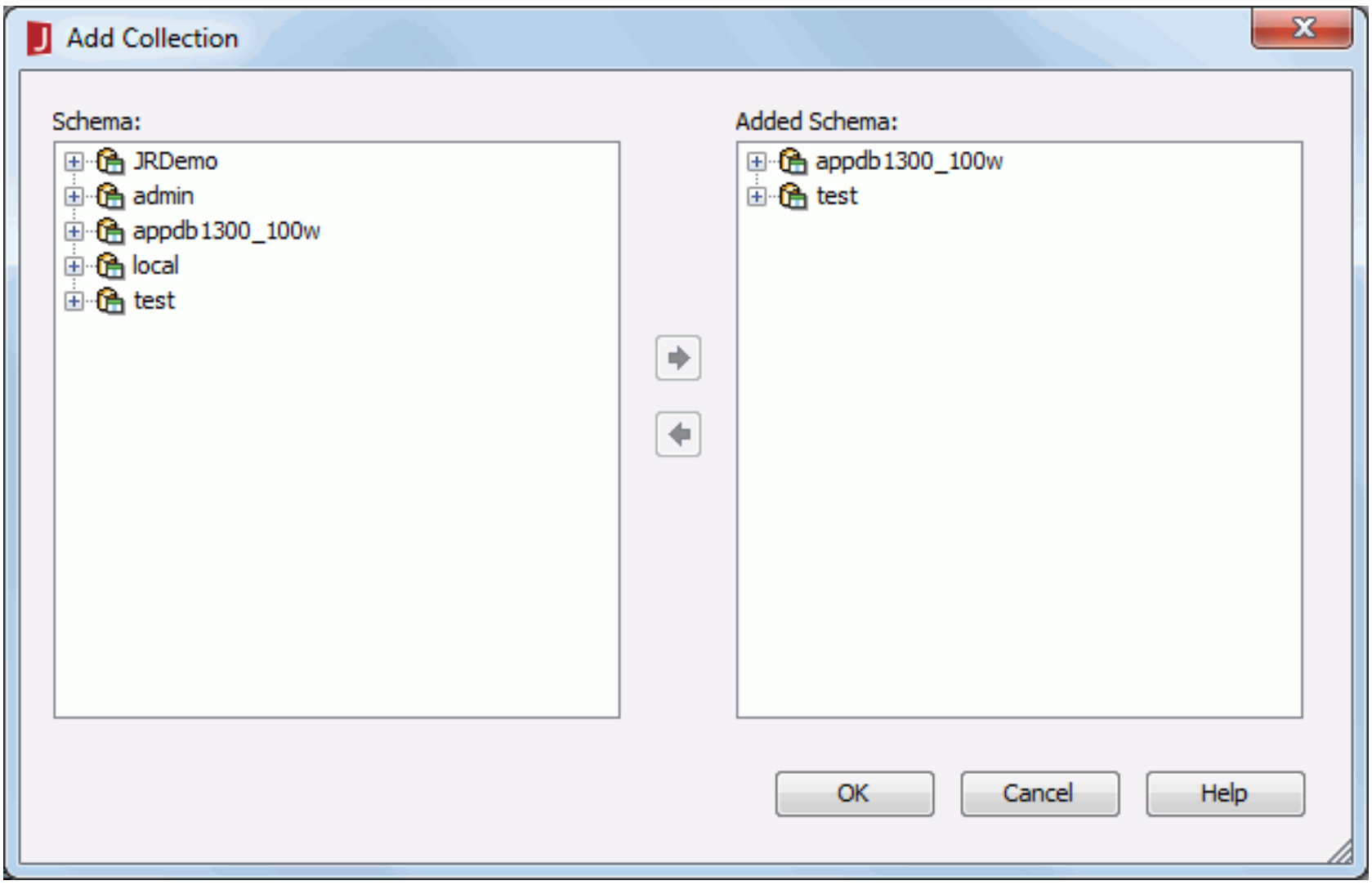
Displays the help document about this feature.

Related topics:

- Using dynamic resources in a library component

Add Collection dialog

This dialog appears when you right-click the Schemas node or a database node in the Schemas node of a MongoDB connection and click Add Collection on the shortcut menu in the Catalog Browser. It helps you to add collections into a JReport catalog via the MongoDB connection you set up.



Schema

- If you open the dialog by right-clicking the Schemas node, the collections in all databases that you can access will be listed here.
- If you open the dialog by right-clicking a database node, the collections in this database will be listed here.

Added Schema

Lists the schemas with the collections that you have added.

Adds the selected collections or all the collections in the selected schema to the MongoDB connection.

Removes the selected collections or all the collections in the selected schema from the Added Schema box.

OK

Adds the selected collections to the MongoDB connection and closes the dialog.

Cancel

Does not retain changes and closes the dialog.

Help

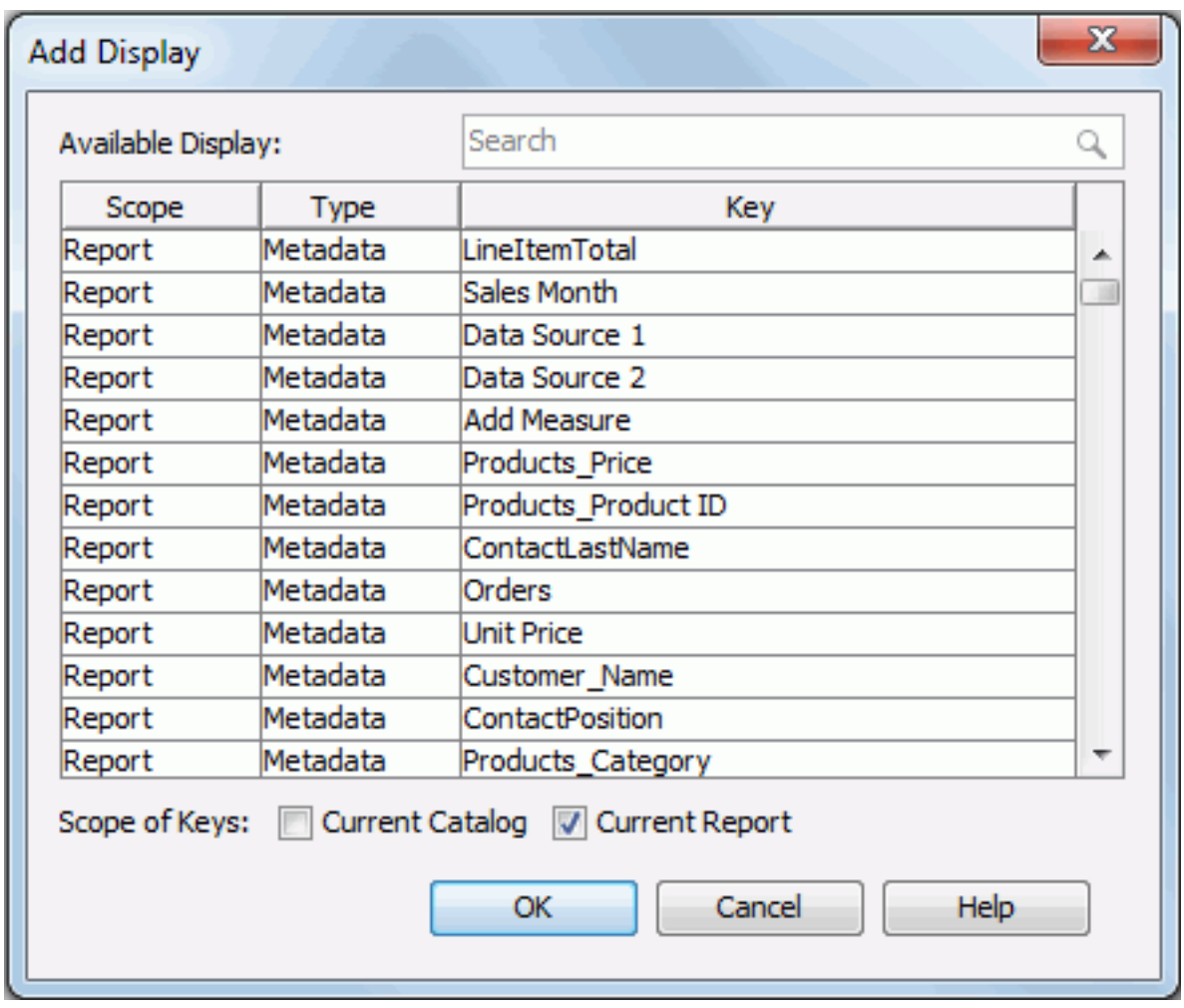
Displays the help document about this feature.

Related topics:

- Managing schemas in a MongoDB connection

Add Display dialog

This dialog appears when you click in the Display tab of the NLS Editor dialog. It helps you to add available display text keys in the current report and catalog for the target language to edit NLS.



The following are details about options in the dialog:

Available Display

Lists all available display text keys in the current report and catalog.

Quick search toolbar

Searches for the keys that contain the specified text. After typing the text, click the search button on the toolbar or press Enter on the keyboard to start searching.

Scope

Lists the scope of the keys. You can click the column header to sort the items by scope in an ascending or descending order.

- **Catalog**
  Indicates the keys are from catalog NLS resources.
- **Report**
  Indicates the keys are from report NLS resources.

Type

Lists types of display text for different objects. You can click the column header to sort the items by type in an ascending or descending order.

- **Column**
  This type is only for reports running in Page Report Studio. It is the type of display text of columns.
- **DisplayName**
  Type of display text of object display name.
- **Label**
  Type of display text of label, some web controls and UDOs.
- **Metadata**
  Type of display text of metadata. Metadata mainly refers to catalog resources, such as table/view columns, business views, formulas, summaries, parameters, and so on.
- **Prompt**
  Type of display text of parameter prompt value.
- **Title**
  Type of display text of filter control and library component.
- **TOC**
  Type of display text on the TOC tree.

Key

Lists keys to indicate the objects in the original language. You can click the column header to sort the items by key in an ascending or descending order.

Scope of Keys

Specifies the scope of the keys to be listed in the Available Display table.

- **Current Catalog**
  Specifies to list keys from the current catalog.
- **Current Report**
  Specifies to list keys from the current report.

OK

Adds the selected keys to the Display tab of the NLS Editor.

Cancel

Cancels the action and exits the dialog.

Help

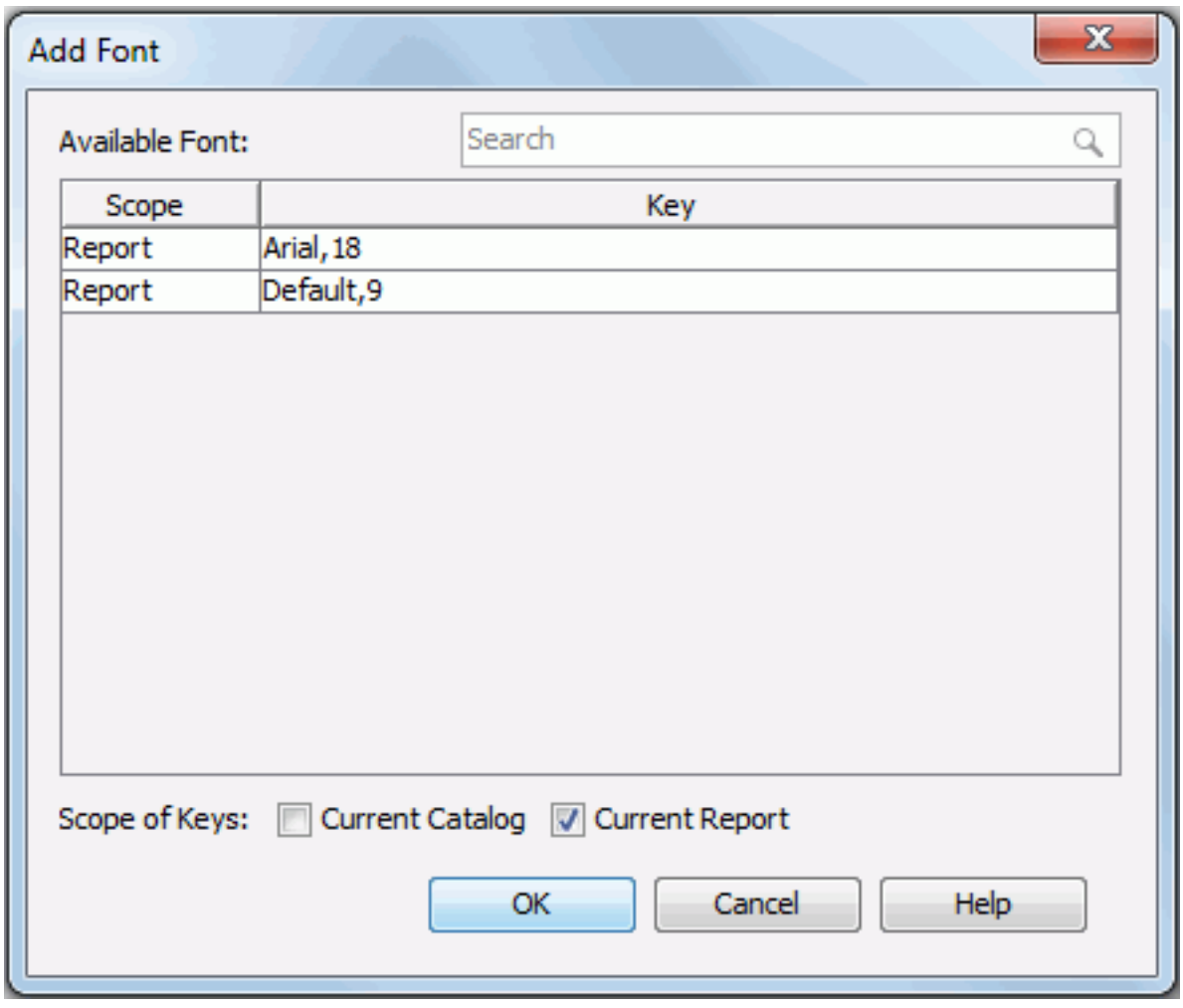
Displays the help document about this feature.

Related topics:

- Creating NLS property files

Add Font dialog

This dialog appears when you click in the Font tab of the NLS Editor. It helps you to add available font keys in the current report and catalog for the target language to edit NLS.



The following are details about options in the dialog:

Available Font

Lists all available font keys in the current report and catalog.

Quick search toolbar

Searches for the keys that contain the specified text. After typing the text, click the search button on the toolbar or press Enter on the keyboard to start searching.

Scope

Lists the scope of the keys. You can click the column header to sort the items by scope in an ascending or descending order.

- **Catalog**
  Indicates the keys are from catalog NLS resources.
- **Report**
  Indicates the keys are from report NLS resources.

Key

Lists keys to indicate the objects in the original language. You can click the column header to sort the items by key in an ascending or descending order.

Scope of Keys

Specifies the scope of the keys to be listed in the Available Font table.

- **Current Catalog**
  Specifies to list keys from the current catalog.
- **Current Report**
  Specifies to list keys from the current report.

OK

Adds the selected keys to the Font tab of the NLS Editor.

Cancel

Cancels the action and exits the dialog.

Help

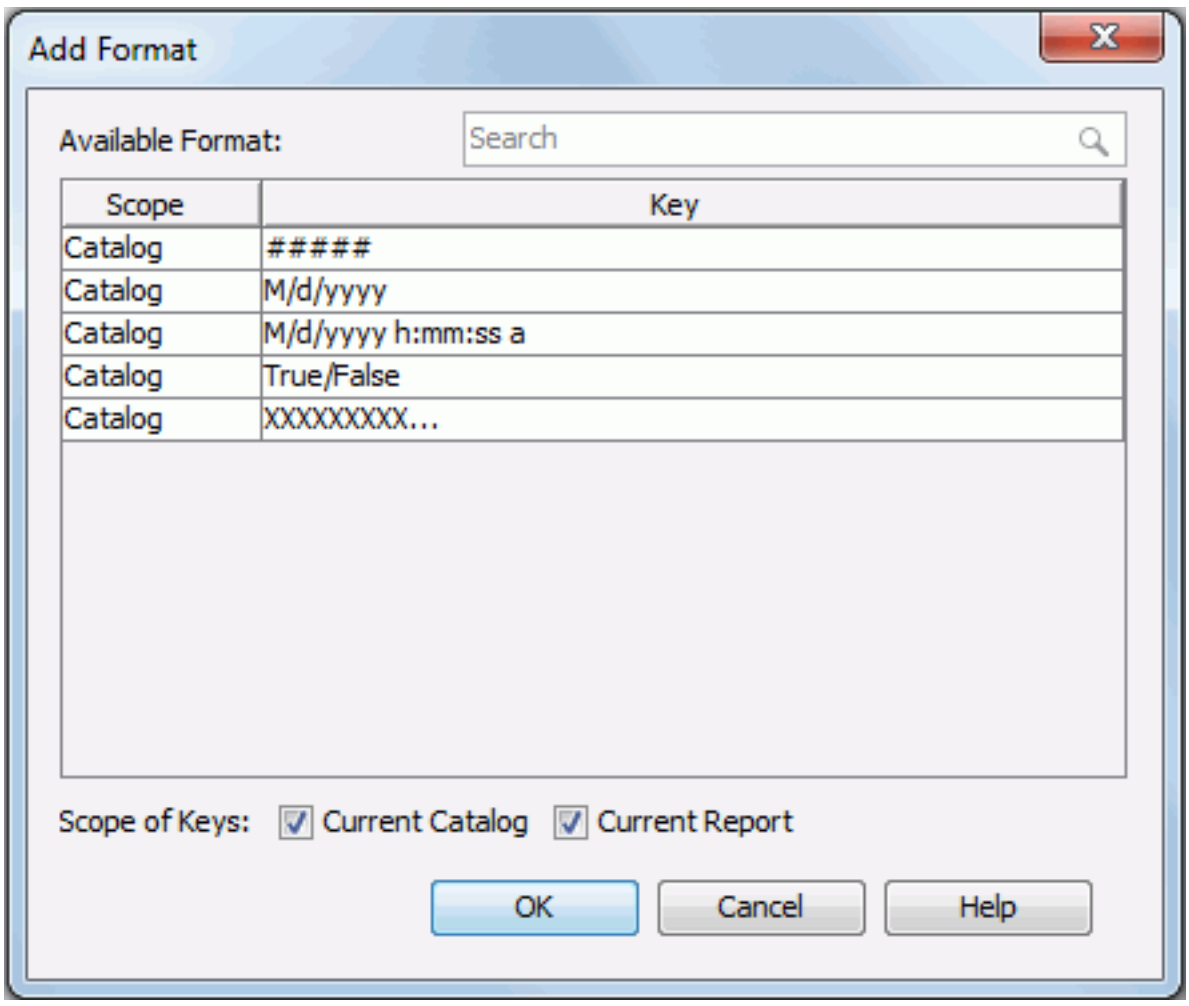
Displays the help document about this feature.

Related topics:

- Creating NLS property files

Add Format dialog

This dialog appears when you click in the Format tab of the NLS Editor dialog. It helps you to add available format keys in the current report and catalog for the target language to edit NLS.



The following are details about options in the dialog:

Available Format

Lists all available format keys in the current report and catalog.

Quick search toolbar

Searches for the keys that contain the specified text. After typing the text, click the search button on the toolbar or press Enter on the keyboard to start searching.

Scope

Lists the scope of the keys. You can click the column header to sort the items by scope in an ascending or descending order.

- **Catalog**
  Indicates the keys are from catalog NLS resources.
- **Report**
  Indicates the keys are from report NLS resources.

Key

Lists keys to indicate the objects in the original language. You can click the column header to sort the items by key in an ascending or descending order.

Scope of Keys

Specifies the scope of the keys to be listed in the Available Format table.

- **Current Catalog**
  Specifies to list keys from the current catalog.
- **Current Report**
  Specifies to list keys from the current report.

OK

Adds the selected keys to the Format tab of the NLS Editor.

Cancel

Cancels the action and exits the dialog.

Help

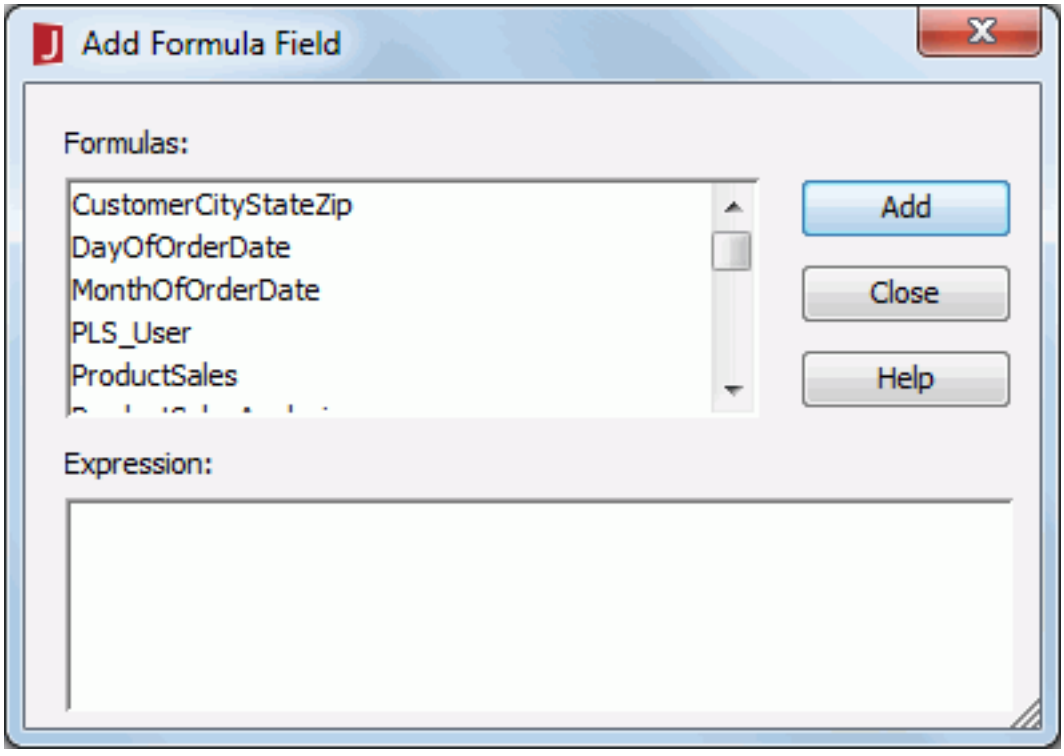
Displays the help document about this feature.

Related topics:

- Creating NLS property files

Add Formula Field dialog

This dialog appears when you click Column > Add Formula Field on the menu bar of the Query Editor window. It helps you to add formulas into a query to be columns of the query.



The following are details about options in this dialog:

Formulas

Lists all the formulas available to be added to the query.

Expression

Shows expressions of the selected formula.

Add

Adds the selected formula to the query.

Close

Exits the dialog.

Help

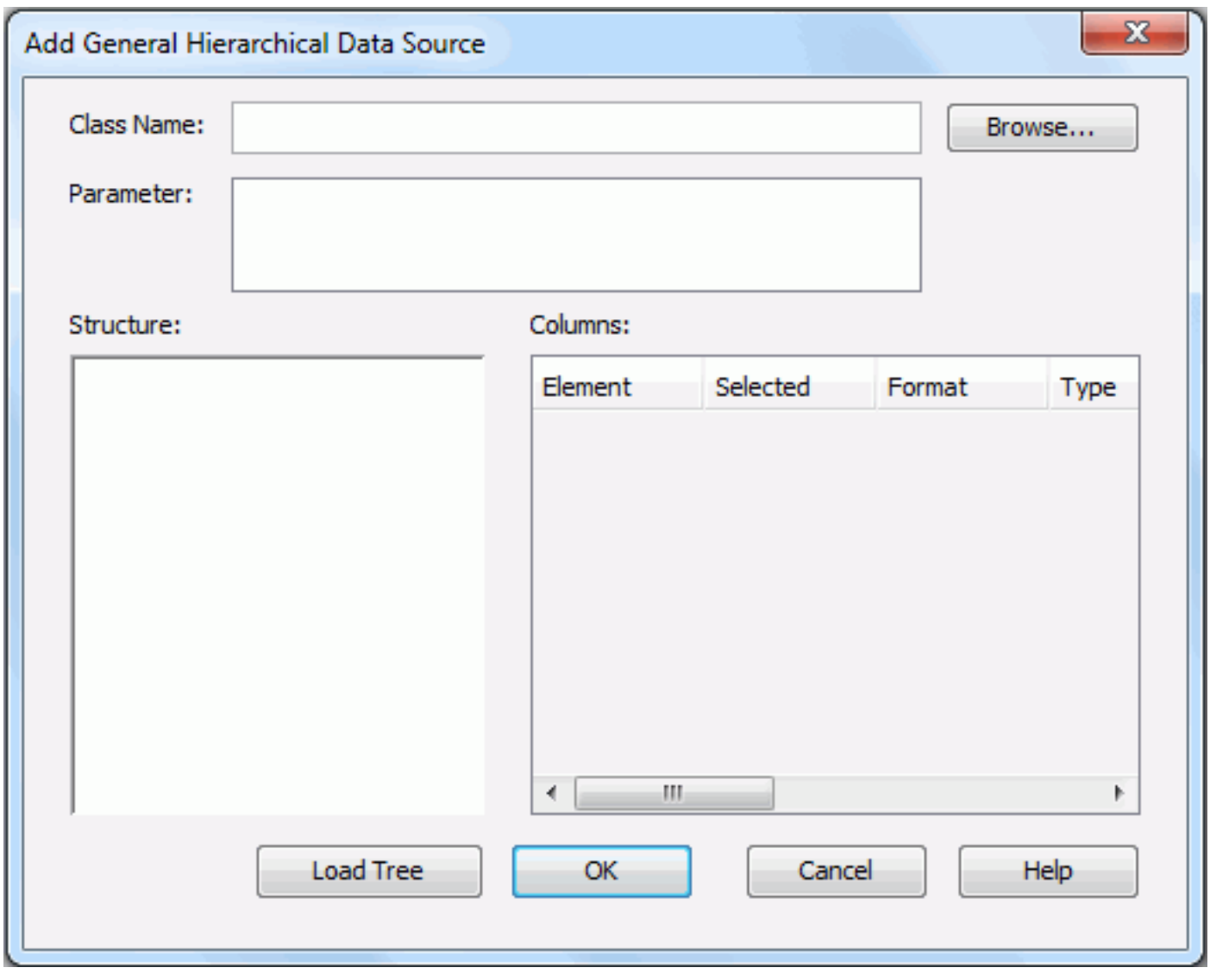
Displays the help document about this feature.

Related topics:

- Adding formula fields to a query

Add General Hierarchical Data Source dialog

This dialog appears when you right-click a data source node and select Add General Hierarchical Data Source from the shortcut menu in the Data tab of the Catalog Browser. It helps you to add a general HDS into a catalog.



The following are details about options in the dialog:

Class Name

Specifies the name of the entrance class that implements the HDS APIs. The class should be in the class path in the system environment.

Parameter

Specifies the parameter string that is required by the class, if any. The parameter string must match the format defined in the HDS class.

Structure

Shows the structure of the hierarchical data source.

Columns

Specifies the column settings.

- **Element**
  Displays the column name in the selected node of table.
- **Selected**
  Specifies whether to import the column into the catalog.
- **Format**
  Specifies the format in which the column values are to be displayed in the report result. Varies for different types of data, and can be manually defined.
- **Type**
  Specifies the data type of the column.
- **Precision**
  Displays the default value for each SQL type automatically. Click the cell and modify the value if necessary.
- **Scale**
  Specifies the decimal place.
- **Nullable**
  Specifies whether the field is nullable. You have three options to choose: Nullable, No Nulls and Nullable Unknown.
- **Currency**
  Specifies whether to use the Currency type for this field or not.
- **Array**
  Specifies whether to use the Array type for this field or not.

Load Tree

Analyzes and loads data tree from the external data source file. The structure is shown in the Structure box after you click this button.

OK

Applies all changes here and closes the dialog.

Cancel

Does not retain changes and closes the dialog.

Help

Displays the help document about this feature.

Related topics:

- Adding an HDS to a catalog

Add Group dialog

The dialog appears when you click and then select Add Group from the drop-down menu in one of the following dialogs or tabs:

- Security dialog
- Edit Business View Security dialog
- Category Property dialog > Security tab
- Add View Element dialog > Security tab
- Edit View Element dialog > Security tab

It helps you to add a group to the security policy so as to define its permission on the related resource, and consists of the following tabs:

- General tab
- Relation tab

OK

Accepts all changes and closes the dialog.

Cancel

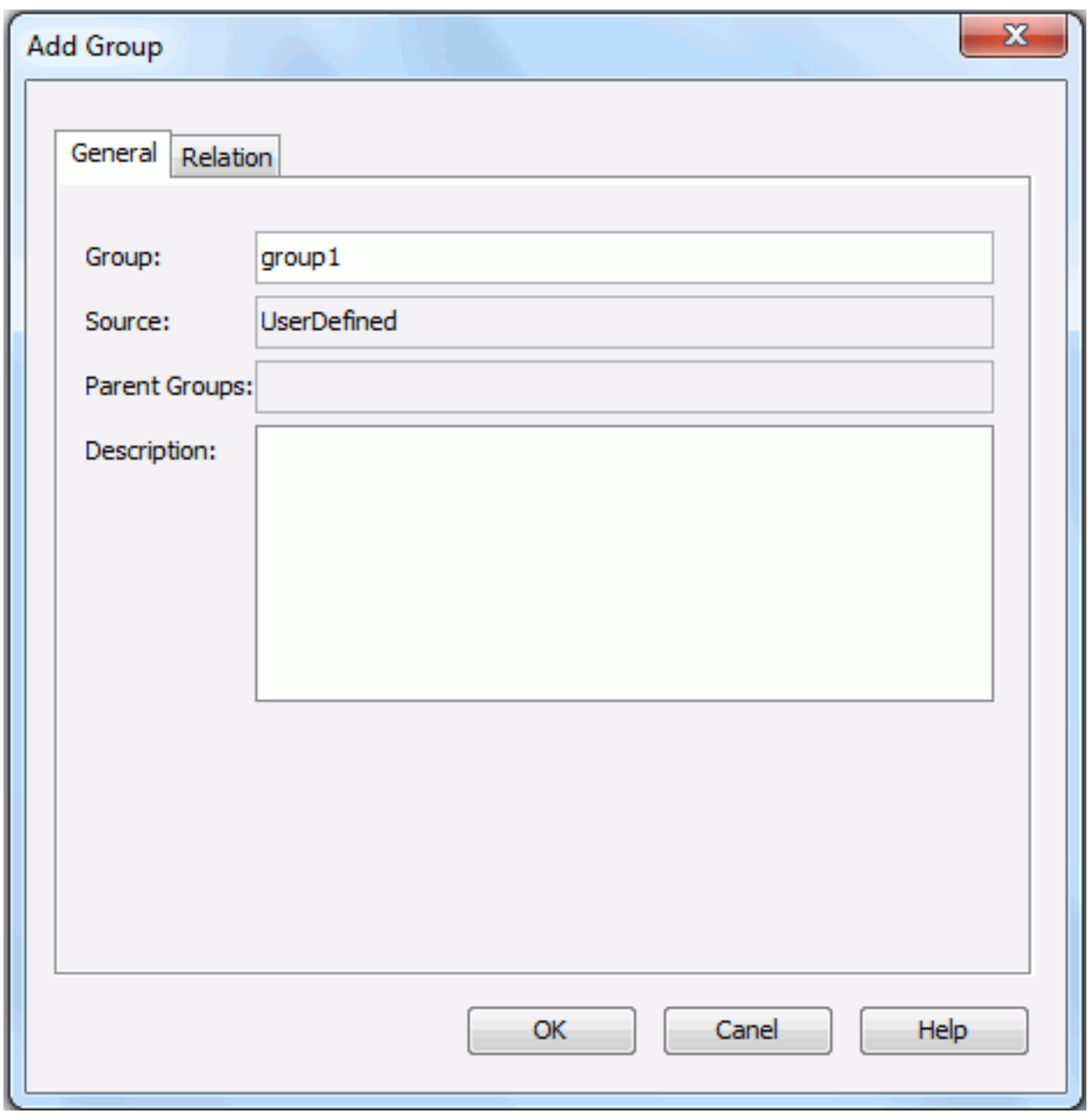
Does not retain any changes and closes the dialog.

Help

Displays the help document about this feature.

General tab

Specifies the group information.



Group

Specifies the name of the group.

Source

Displays the source of the group. It is read only.

Parent Groups

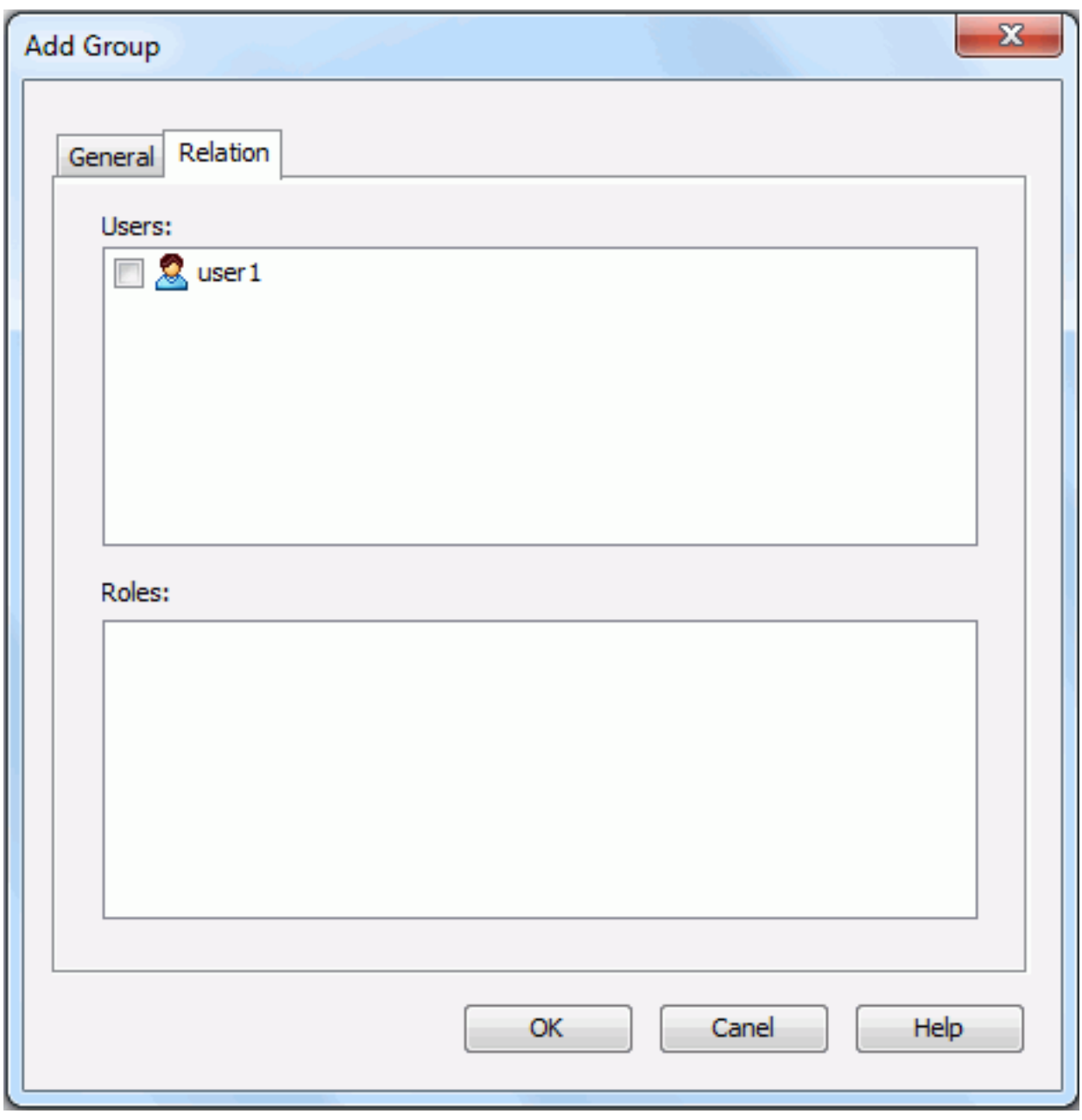
Displays the parent groups of the current group. It is read only.

Description

Specifies the description of the group.

Relation tab

Specifies the relationship of the group and the existing users and roles.



Users

All available users will be listed here. If a user is checked, the user will be added into the group.

Roles

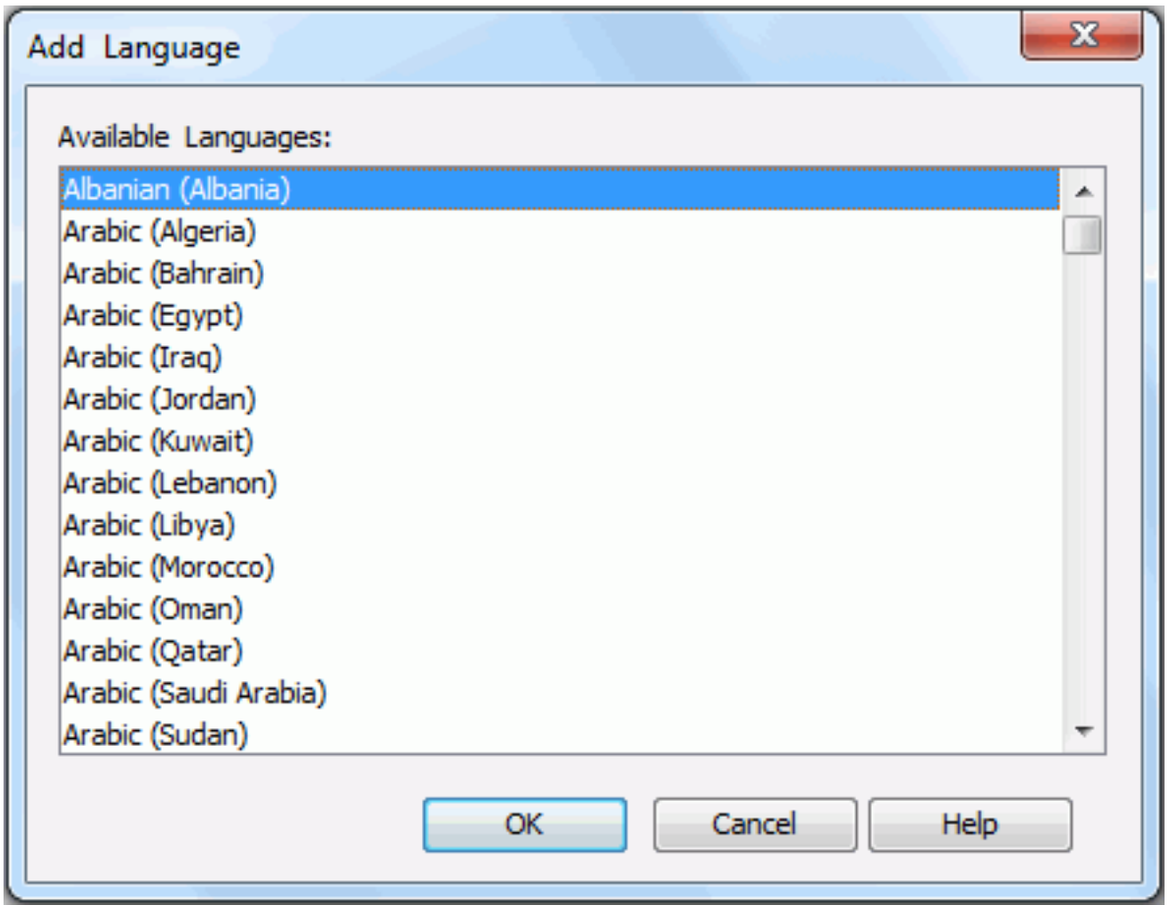
All available roles will be listed here. If a role is checked, the group will be added into the role.

Related topics:

- Setting up a connection-scope security policy
- Configuring member-level security

Add Language dialog

This dialog appears when you click above the Language box in the NLS Editor dialog. It helps you to add one or more languages, in which the report can run, to the NLS Editor for editing.



The following are details about options in the dialog:

Available Languages

Lists all the languages that are available for you to choose.

OK

- When Local NLS is checked in the NLS Editor, adds the selected languages to the Language box in the NLS Editor and closes the dialog.
- When Global NLS is checked in the NLS Editor, opens the Connect to JReport Server dialog to download global NLS resources that have been defined for the selected languages from the server.

Cancel

Cancels the action and exits the dialog.

Help

Displays the help document about this feature.

Related topics:

- Creating NLS property files

Add Role dialog

The dialog appears when you click and then select Add Role from the drop-down menu in one of the following dialogs or tabs:

- Security dialog
- Edit Business View Security dialog
- Category Property dialog > Security tab
- Add View Element dialog > Security tab
- Edit View Element dialog > Security tab

It helps you to add a role to the security policy so as to define its permission on the related resource, and consists of the following tabs:

- General tab
- Relation tab

OK

Accepts all changes and closes the dialog.

Cancel

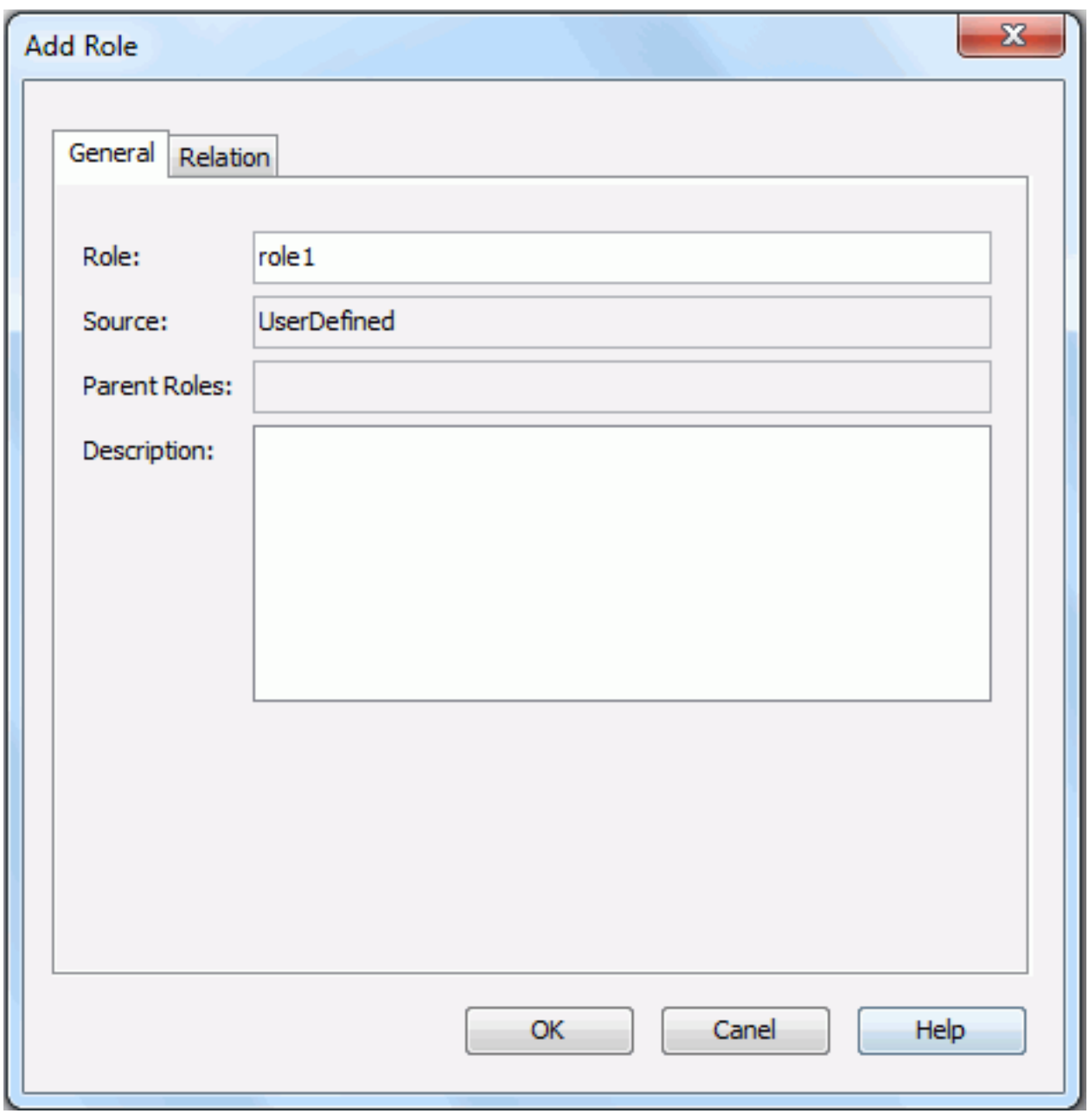
Does not retain any changes and closes the dialog.

Help

Displays the help document about this feature.

General tab

Specifies the role information.



Role

Specifies the name of the role.

Source

Displays the source of the role. It is read only.

Parent Roles

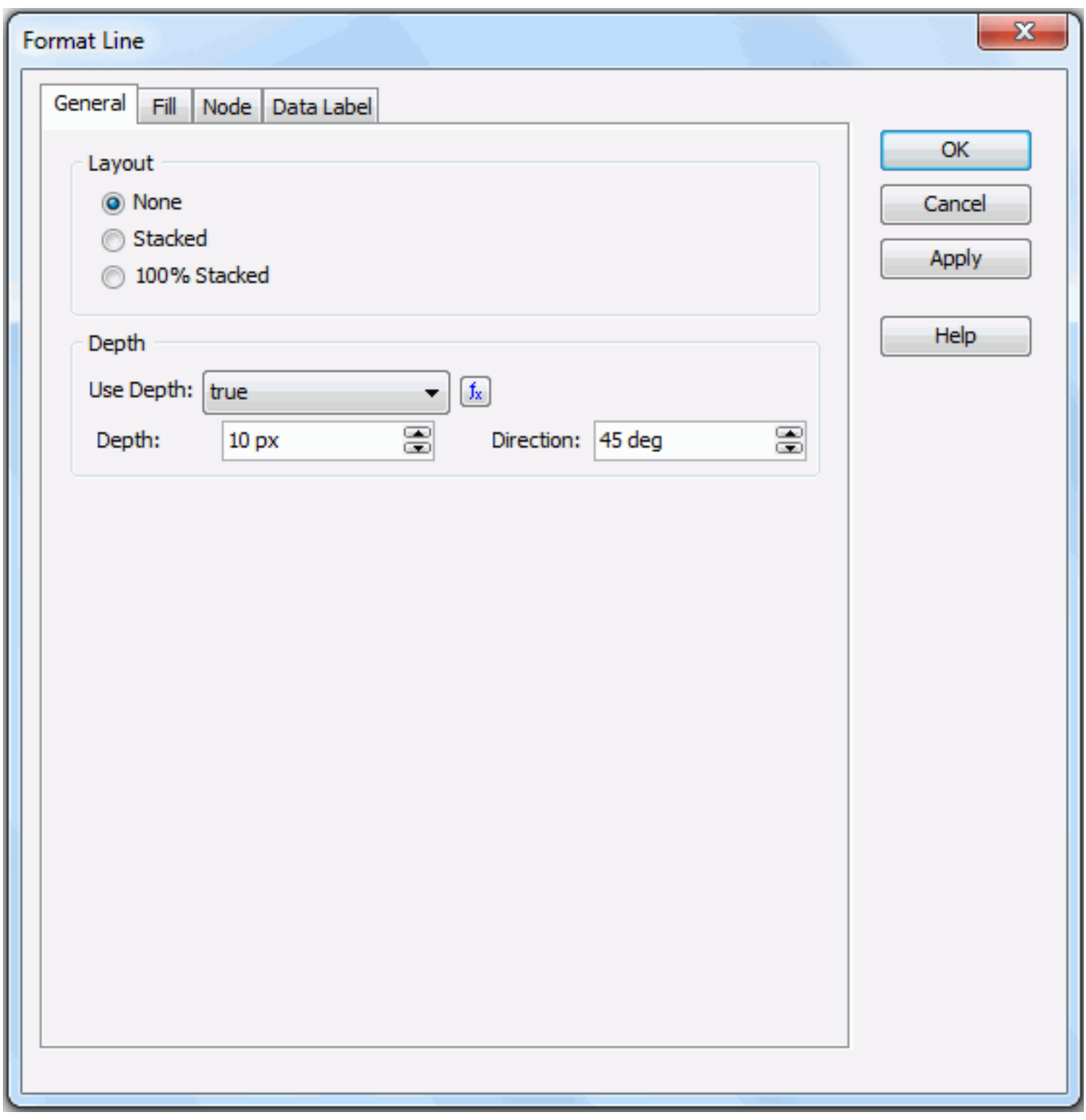
Displays the parent roles of the current role. It is read only.

Description

Specifies the description of the role.

Relation tab

Specifies the relationship of the role and the existing users and groups.



Users

All available users will be listed here. If a user is checked, the user will be added into the role.

Groups

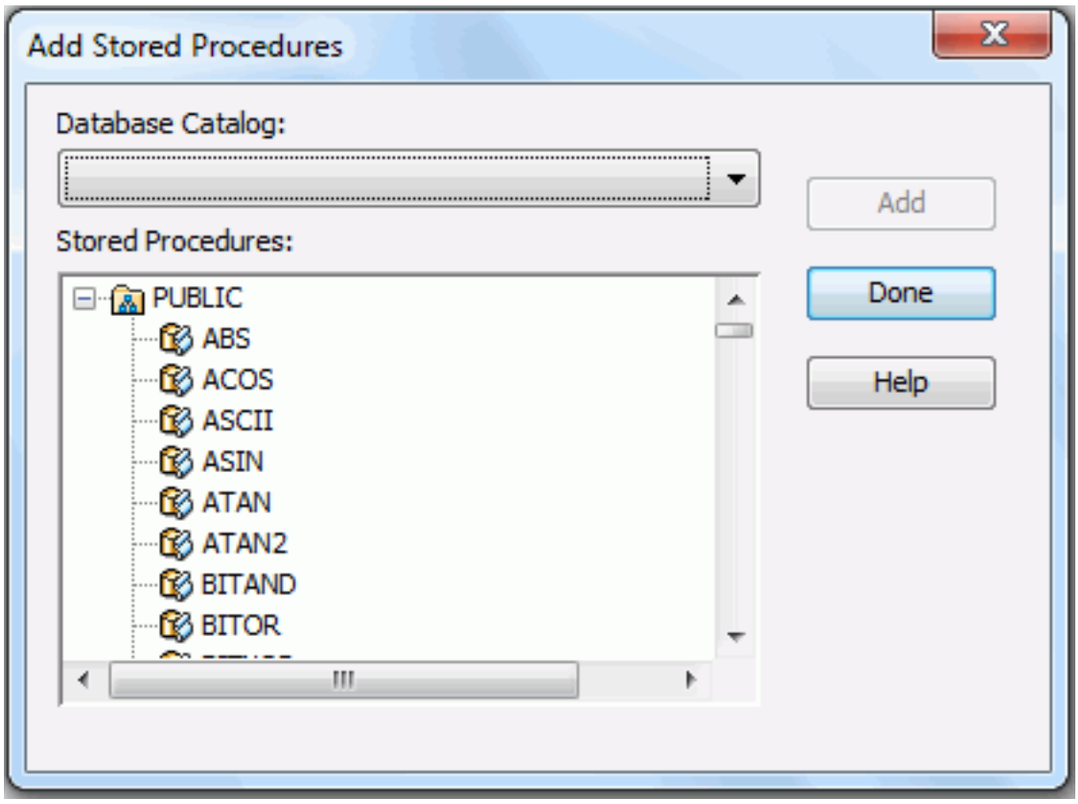
All available groups will be listed here. If a group is checked, the group will be added into the role.

Related topics:

- Setting up a connection-scope security policy
- Configuring member-level security

Add Stored Procedures dialog

This dialog appears when you right-click a JDBC connection node and then click Add Stored Procedure from the shortcut menu in the Catalog Browser. It displays all the stored procedures in the database, and allows you to select stored procedures for adding to a catalog.



The following are details about options in the dialog:

Database Catalog

Lists all the catalogs in the database.

Stored Procedures

Lists all the stored procedures in a three level tree. The top level is SQL-catalog, second is SQL-schema and the third are stored procedures. You can select one or more stored procedures to add to the catalog.

Note: Not all DBMSs support database stored procedures. In this case, you may see nothing in the box.

Not all stored procedures will return data usable by JReport. The stored procedure needs to return a result set.

Add

Adds the selected stored procedures to the catalog.

Done

Applies all changes and closes the dialog.

Help

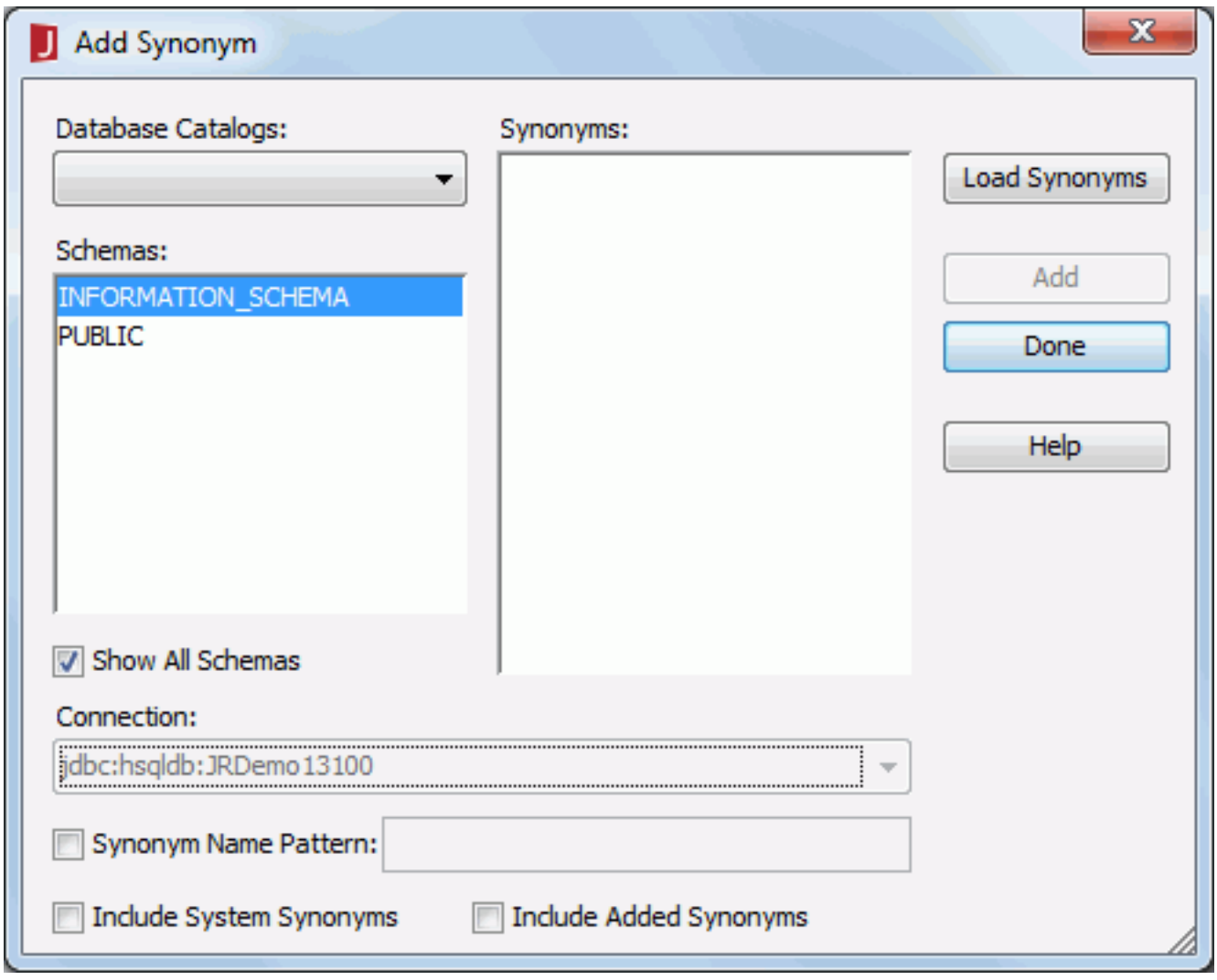
Displays the help document about this feature.

Related topics:

- Adding stored procedures to a catalog

Add Synonym dialog

This dialog appears when you right-click a JDBC connection node and then click Add Synonyms from the shortcut menu, or right-click the Synonyms node in the JDBC connection and click Add Synonym from the shortcut menu in the Catalog Browser. It helps you to add synonyms into a JReport catalog via the connection you set up.



The following are details about options in the dialog:

Database Catalogs

Lists all the catalogs in the database.

Schemas

Lists the schemas you have selected in the Schema tab of the Get JDBC Connection Information dialog.

Show All Schemas

If checked, all schemas in the selected catalog will be shown in the Schemas box.

Synonyms

Lists the synonyms contained in the selected schema after you click the Load Synonyms button. The synonyms include tables and views whose Type property is Synonym or Alias.

Connection

Shows the name of the current connection.

Synonym Name Pattern

Specifies whether to filter the tables according to the specified filtering condition.

Include System Synonyms

If checked, the system synonyms will become available in the Synonyms box.

Include Added Synonyms

If checked, the synonyms that have been added to the catalog will become available in the Synonyms box. You can then add them into the catalog again with a different name.

Load Synonyms

Loads the synonyms contained in the specified schema into the Synonyms box.

Add

Adds the selected synonyms to the catalog.

Done

Applies all changes and closes the dialog.

Help

Displays the help document about this feature.

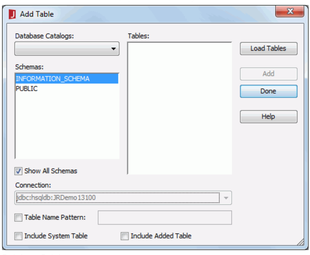
Related topics:

- Adding tables/views/synonyms to a catalog

Add Table dialog

This dialog appears when you right-click the Tables node in a connection and click Add Table on the shortcut menu in the Catalog Browser. It helps you to add tables into a JReport catalog via the connection you set up, and varies according to connection types.

If it is a JDBC or HIVE connection, options in the dialog are as follows.



Database Catalogs

Lists all the catalogs in the database.

Schemas

Lists the schemas you have selected in the Schema tab of the Get JDBC Connection Information dialog.

Show All Schemas

If checked, all schemas in the selected catalog will be shown in the Schemas box.

Tables

Lists the tables contained in the selected schema after you click the Load Tables button.

Connection

Shows the name of the current connection.

Table Name Pattern

Specifies whether to filter the tables according to the specified filtering condition.

Include System Table

If checked, the system tables will become available in the Tables box.

Include Added Table

If checked, the tables that have been added to the catalog will become available in the Tables box. You can then add them into the catalog again with a different name.

Load Tables

Loads the tables contained in the specified schema into the Tables box.

Add

Adds the selected tables to the JReport catalog.

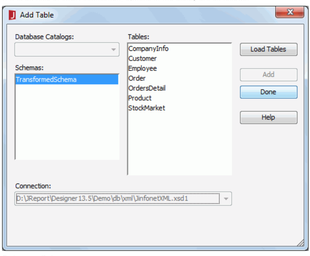
Done

Completes the adding tables process and closes the dialog.

Help

Displays the help document about this feature.

If it is an XML or JSON connection, options in the dialog are as follows.



Database Catalogs

Lists all the catalogs in the database. Not supported on XML connection.

Schemas

Lists the schema transformed from the XML source.

Tables

Lists the tables contained in the schema after you click the Load Tables button.

Connection

Shows the name of the current connection.

Load Tables

Loads the tables contained in the specified schema into the Tables box.

Add

Adds the selected tables to the JReport catalog.

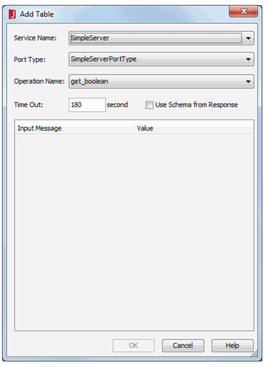
Done

Completes the adding tables process and closes the dialog.

Help

Displays the help document about this feature.

If it is a web service connection, options in the dialog are as follows.



Service Name

Contains all the service information defined in the WSDL file. Select one you want from the drop-down list.

Port Type

Specifies the port type for the selected service. A port type is a set of abstract operations and the abstract messages involved. Available only when the web service is defined by WSDL 1.1. Select one you want from the drop-down list.

Operation Name

Contains all the operation information for the selected service. Select one you want from the drop-down list. An operation is an abstract description of an action supported by the service. Each operation refers to an input message and output messages.

Time Out

Specifies how long to wait to get the service and operation information defined in the WSDL file.

Use Schema from Response

This option is available only when the web service is defined by WSDL 1.1.

When the XML schema in the output message does not match the specific XML instance in the SOAP responded from web service, JReport will not be able to read data properly from that XML instance. In this condition, you can use this option to control whether to ignore the XML schema in the output message and parse the XML schema directly from the specific XML instance in the SOAP for the web service.

Input Message

Lists the input message information for the selected operation.

Value

Specifies the value for the input message. Type in the value, or define a constant level formula or a parameter to be the value.

OK

Accepts the adding table result and closes the dialog.

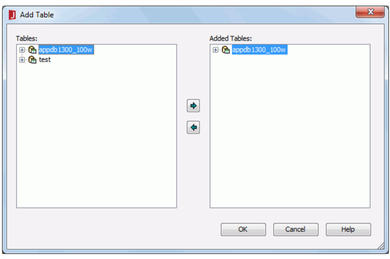
Cancel

Cancels the adding table process and exists the dialog.

Help

Displays the help document about this feature.

If it is a MongoDB connection, options in the dialog are as follows.



Tables

Lists the tables transformed from the MongoDB schemas.

Added Tables

Lists the tables that you have added.

Adds the selected tables.

Removes the selected tables from the Added Tables box.

OK

Accepts the adding table result and closes the dialog.

Cancel

Cancels the adding table process and exists the dialog.

Help

Displays the help document about this feature.

Related topics:

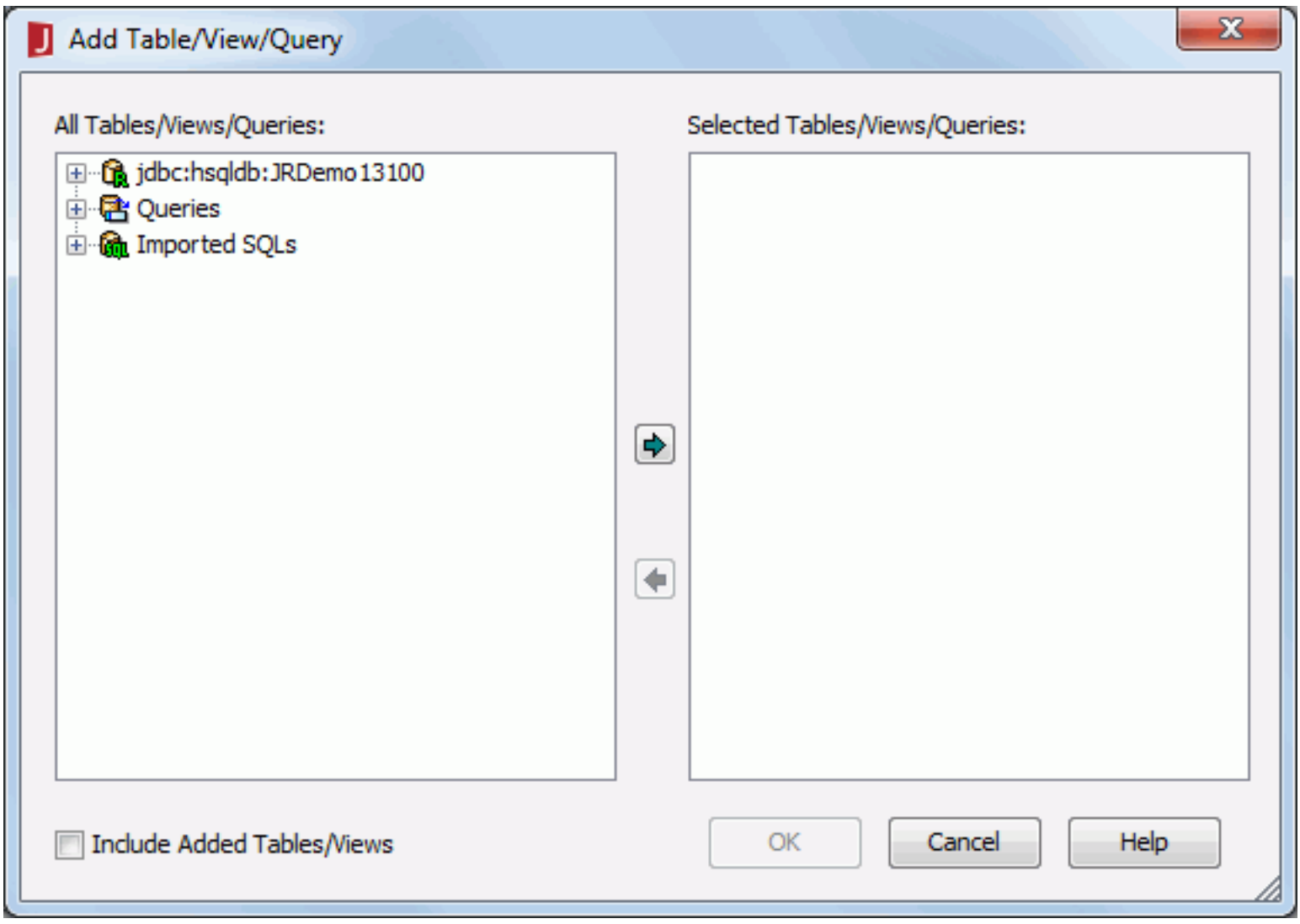
- Tables/Views/Synonyms
- Managing tables in a JSON connection
- Managing tables in an XML connection
- Adding tables to a web service connection
- Managing tables in a MongoDB connection

Add Table/View/Query dialog

This dialog appears when you do either of the following:

- In the Query Editor, click the Add Table/View/Query button on the toolbar or select Add Table/View/Query from the Query menu.
- In the Catalog Browser, right-click the Business View node and select Add Business View from the shortcut menu. Then in the Input Business View Name dialog provide a name and click OK.

It helps you to add resources to use in a query or a business view.



All Tables/Views/Queries

List all the tables, views, synonyms, queries, imported SQLs, stored procedures, and user defined data sources in the current catalog data source that can be added to the query or business view.

When two resources (for example, a table and a view) use the same name, they cannot be added at the same time, and when a table is already contained in a query, you cannot add the table and the query at the same time.

Selected Tables/Views/Queries

Lists all the added tables, views, synonyms, queries, imported SQLs, stored procedures, and user defined data sources.

Adds the selected resources from the left box to the right box. If the selected resources contain a table, view or synonym that is already added, the Input Table Name dialog is displayed for you to specify another name for the resource.

Removes the selected resources from the right box.

Include Added Tables/Views

Specifies whether to list the tables, views and synonyms in JDBC connections in the All Tables/Views/Queries box that have already been added to the right box. This option is available only when there is at least a JDBC connection in the current catalog data source.

OK

Adds the selected resources to the current query or business view. When a query, imported SQL, stored procedure or user defined data source is added, it will be added as a single table with all of its columns.

Cancel

Exits the dialog and discards the setting.

Help

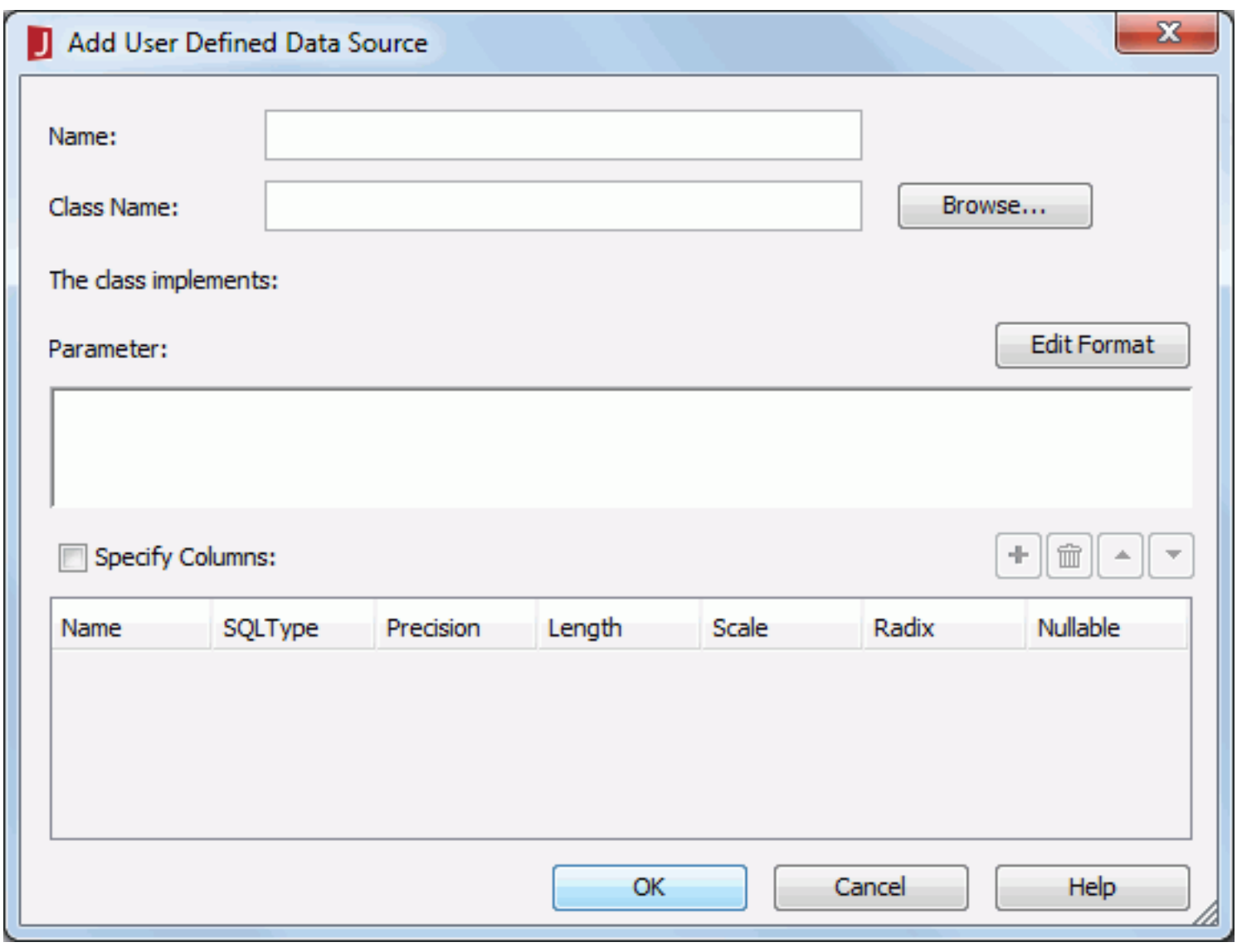
Displays the help document of this feature.

Related topics:

- Adding and deleting tables in a query

Add User Defined Data Source dialog

This dialog appears when you right-click a data source node and select Add User Defined Data Source from its shortcut menu in the Data tab of the Catalog Browser. It is used to add user data source objects into a JReport catalog.



The following are details about options in the dialog:

Name

Specifies the name of the UDS. The name cannot be the same as any of the existing procedures, SQL files, HDSs, business views or queries. It also cannot be empty.

Class Name

Specifies the full name (including package name) of the class represented by the UDS. The class should be appended to the class path in the system environment.

The class implements

After filling in the Class Name field, the classname of the interface that the class implements will be displayed here.

Parameter

Specifies the parameter string for the UDS. The parameter string must match the format defined in the UDS class. You can use parameters or constant level formulas predefined in the current catalog data source or the special field User Name in the parameter string, and the format of the predefined parameters used in the string is "@" + parameter name. For example, if the parameter string of a UDS is SQL=select * from employee, and you want to use the parameter sql in a catalog to replace the part after "=" in the string, then the parameter string of the UDS will be SQL=@sql.

Edit Format

Opens the Edit Parameter Format dialog, in which you can edit the parameter format:

- **Name**
  Displays names of the parameters used in the UDS.
- **Format**
  Specifies formats for the parameters as required.
- **OK**
  Accepts changes and closes the dialog.
- **Cancel**
  Does not retain changes and closes the dialog.

Specify Columns

Specifies the column definitions. If you don't specify the column definitions, JReport will retrieve them from result set.

- **Name**
  Specifies the valid name of the column. The name of the column should have the same validation with a common table column. The default names for column definitions are ‘column1', ‘column2', and so on.
- **SQLType**
  Specifies the data type of the column. The following types are supported:
  - **String**
    The corresponding JDBC type is VARCHAR or LONGVARCHAR.
  - **java.math.BigDecimal**
    The corresponding JDBC type is NUMERIC.
  - **boolean**
    The corresponding JDBC type is BIT.
  - **byte**
    The corresponding JDBC type is TINYINT.
  - **short**
    The corresponding JDBC type is SMALLINT.
  - **int**
    The corresponding JDBC type is INTEGER.
  - **long**
    The corresponding JDBC type is BIGINT.
  - **float**
    The corresponding JDBC type is REAL.
  - **double**
    The corresponding JDBC type is DOUBLE.
  - **java.io.InputStream**
    The corresponding JDBC type is VARBINARY or LONGVARBINARY.
  - **java.sql.Date**
    The corresponding JDBC type is DATE.
  - **java.sql.Time**
    The corresponding JDBC type is TIME.
  - **java.sql.Timestamp**
    The corresponding JDBC type is TIMESTAMP.
- **Precision, Length, Scale, Radix**
  Specifies the default value for each SQL type. Click the cell to modify the value if necessary.
- **Nullable**
  Specifies whether the value of the column can be null. X stands for No Nulls, •stands for Nullable and ? stands for Nullable Unknown.

Adds a default row in the column definitions. If there is no row selected, an empty row will be added to the last line.

Removes the selected row.

Moves the selected column one step up.

Moves the selected column one step down.

OK

Adds the UDS and closes the dialog.

Cancel

Does not retain any changes and closes the dialog.

Help

Displays the help document about this feature.

Related topics:

- Adding a UDS to a catalog

Add User dialog

The dialog appears when you click and then select Add User from the drop-down menu in one of the following dialogs or tabs:

- Security dialog
- Edit Business View Security dialog
- Category Property dialog > Security tab
- Add View Element dialog > Security tab
- Edit View Element dialog > Security tab

It helps you to add a user to the security policy so as to define his permission on the related resource, and consists of the following tabs:

- General tab
- Relation tab

OK

Accepts all changes and closes the dialog.

Cancel

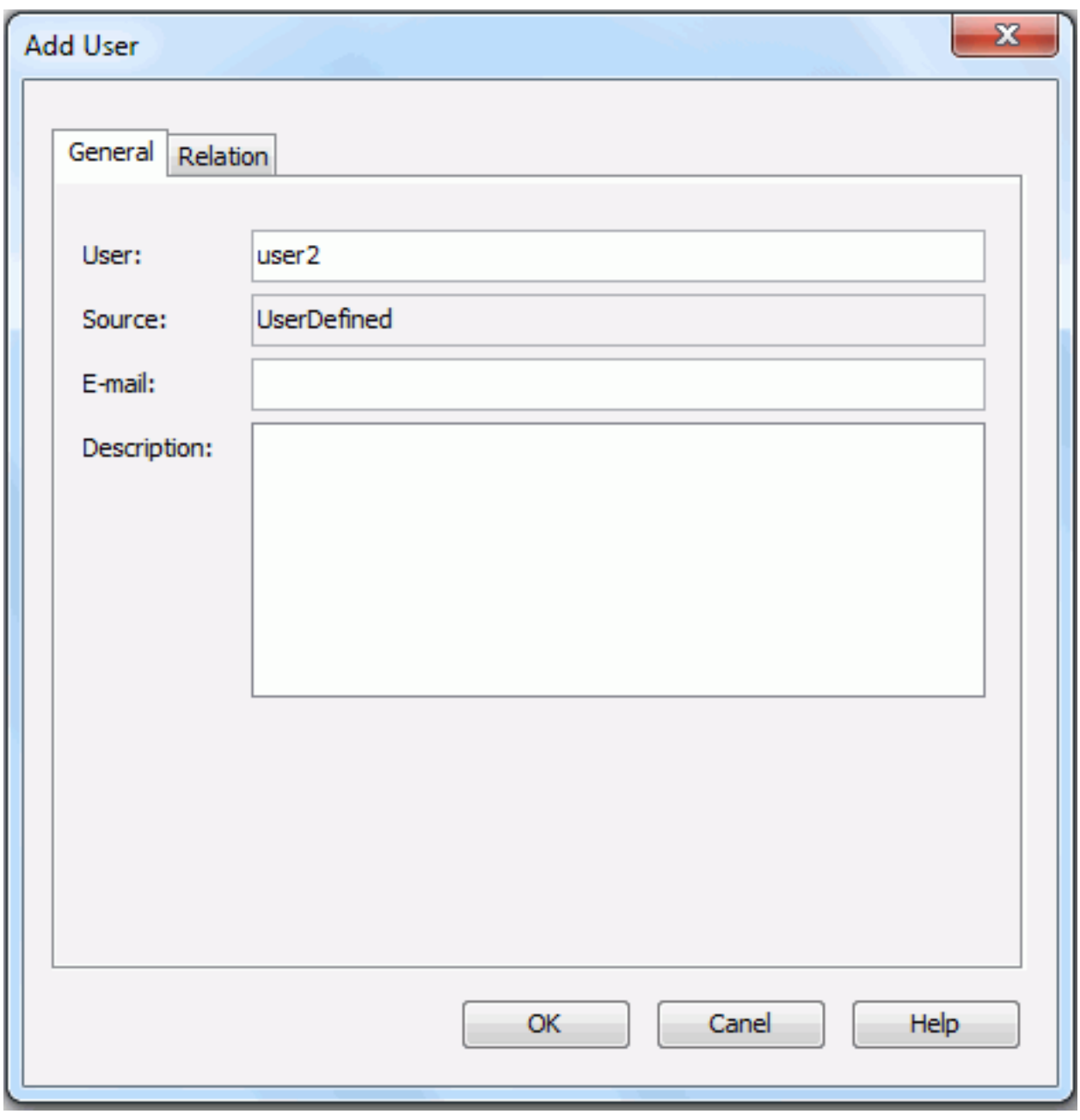
Does not retain any changes and closes the dialog.

Help

Displays the help document about this feature.

General tab

Specifies the user information.



User

Specifies the name of the user.

Source

Displays the source of the user. It is read only.

E-mail

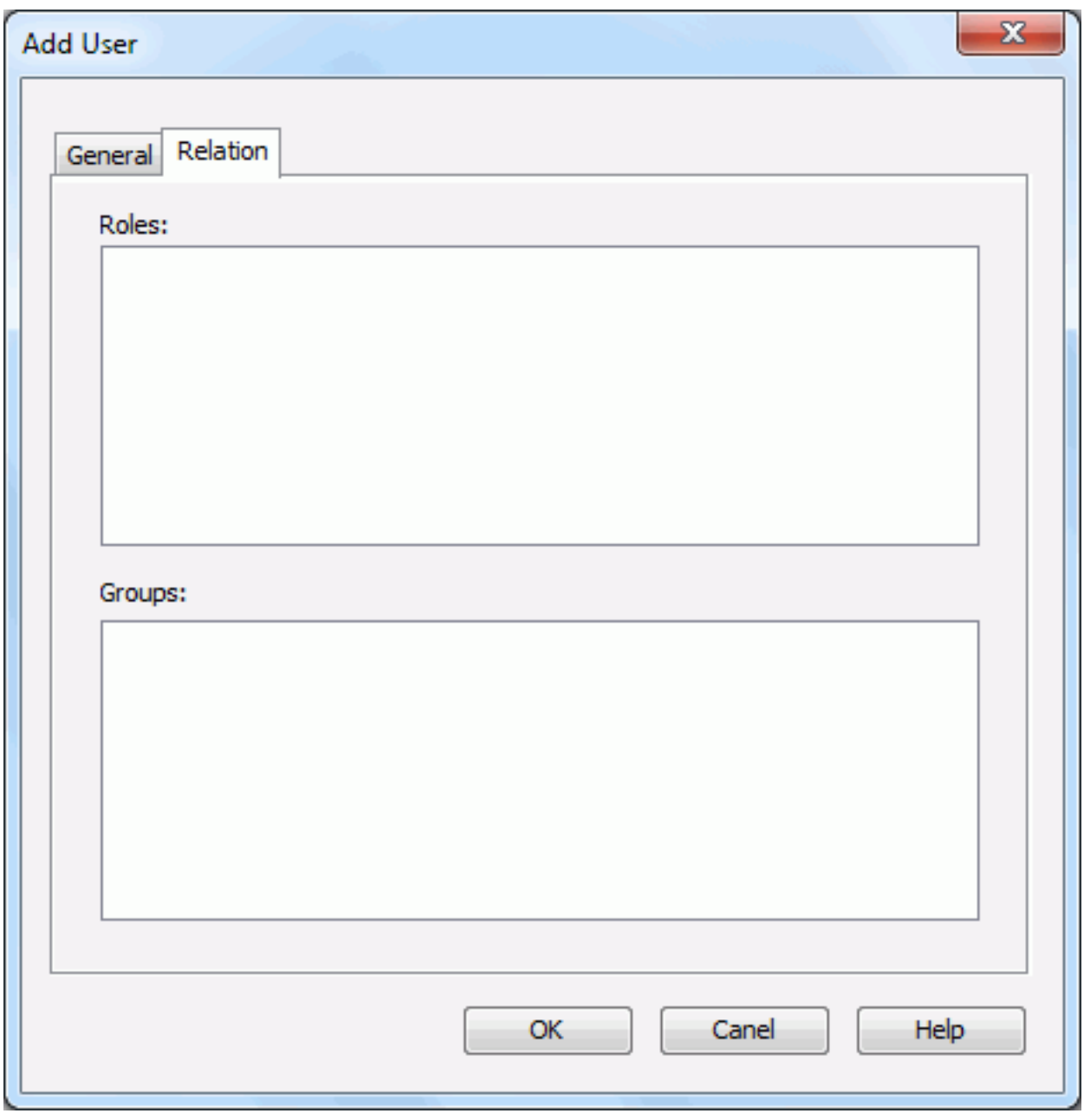
Specifies the e-mail address of the user that is to be added.

Description

Specifies the description of the user.

Relation tab

Specifies the relationship of the user and the existing roles and groups.



Roles

All available roles will be listed here. If a role is checked, the user will belong to the role.

Groups

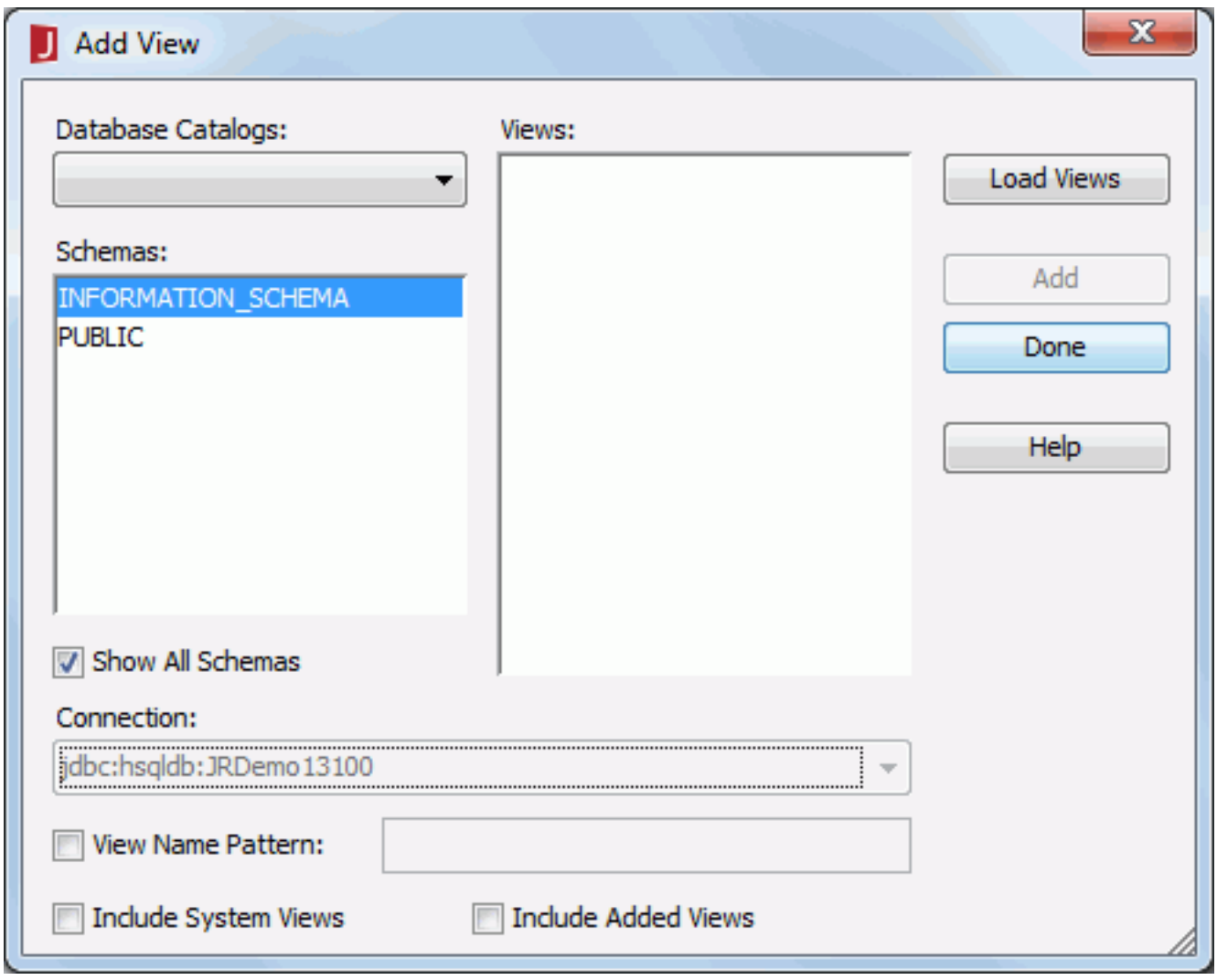
All available groups will be listed here. If a group is checked, the user will belong to the group.

Related topics:

- Setting up a connection-scope security policy
- Configuring member-level security

Add View dialog

This dialog appears when you right-click the Views node and select Add View from the shortcut menu, or right-click a view and click New View on the shortcut menu in the Catalog Browser. It helps you to add views into a JReport catalog.



The following are details about options in the dialog:

Database Catalogs

Lists all the catalogs in the database.

Schemas

Lists the schemas you have selected in the Schema tab of the Get JDBC Connection Information dialog.

Show all Schemas

If checked, all schemas in the current catalog will be shown in the Schema box.

Views

Lists the views contained in the selected schema after you click the Load Views button.

Connection

Shows the name of the current connection.

View Name Pattern

Specifies whether to filter the views according to the specified filtering condition.

Include System View

If checked, system views will become available in the Views box.

Include Added View

If checked, views that have been added to the catalog will become available in the Views box. You can then add them into the catalog again with a different name.

Load Views

Loads the views contained in the specified schema to the Views box.

Add

Adds the selected views to the JReport catalog.

Done

Completes the adding views process and closes the dialog.

Help

Displays the help document about this feature.

Related topics:

- Adding tables/views/synonyms to a catalog

Add View Element dialog

This dialog appears when you,

- Do any of the following in the Business View Editor:
  - Click Insert > Group/Aggregation/Detail.
  - Click the Insert Group button , Insert Aggregation button or Insert Detail button .
  - Right-click the business view or a category and select Add View Element from the shortcut menu.
- Right-click a business view or a category and click Add View Element on the shortcut menu in the Catalog Browser resource tree.

It helps you to add a new view element into a business view, and consists of the following tabs:

- General tab
- Security tab

OK

Applies all changes and closes the dialog.

Cancel

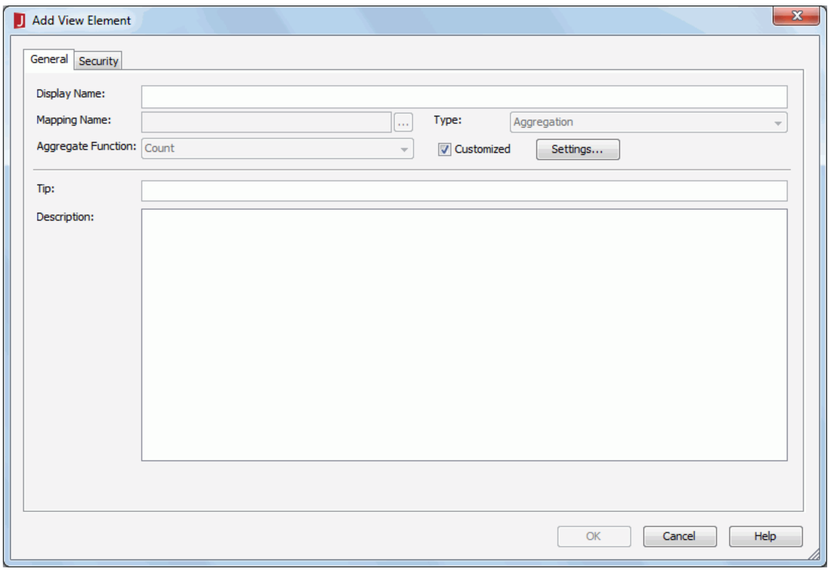
Does not retain changes and closes the dialog.

Help

Displays the help document about this feature.

General tab

Specifies properties of the view element.



Display Name

Specifies the display name of the view element which is intuitive to the end user.

Mapping Name

Specifies the mapping name of the field to which the view element is mapped. Click ... to select the field.

Type

Specifies the type of the view element.

- **Group**
  Group objects are the view elements that will become the basis for analyzing data in a report. They characteristically return text data or dates. For example they may return items like country, state and year to use for grouping data in reports.
- **Aggregation**
  Aggregation objects are numeric view elements that are calculated at the time the report runs. The values they return are dependent on the group objects which they are used with. For example you may want a sales total by year.
- **Detail**
  Detail objects provide additional information. They can be bound to group, aggregation, or category elements. Detail objects are not used for analysis but are useful in tables to show detail information such as addresses and phone numbers.

Aggregate Function

Available only when the type of the view element is defined as Aggregation. It specifies the function for the aggregation object. For details about each function, see Math functions.

Customized

Available only when the type of the view element is defined as Aggregation. If checked, you can create a custom aggregation by writing a formula.

- **Settings**
  Opens the Formula Editor dialog to compose a formula using resources in the current business view, which can return an aggregation.

Tip

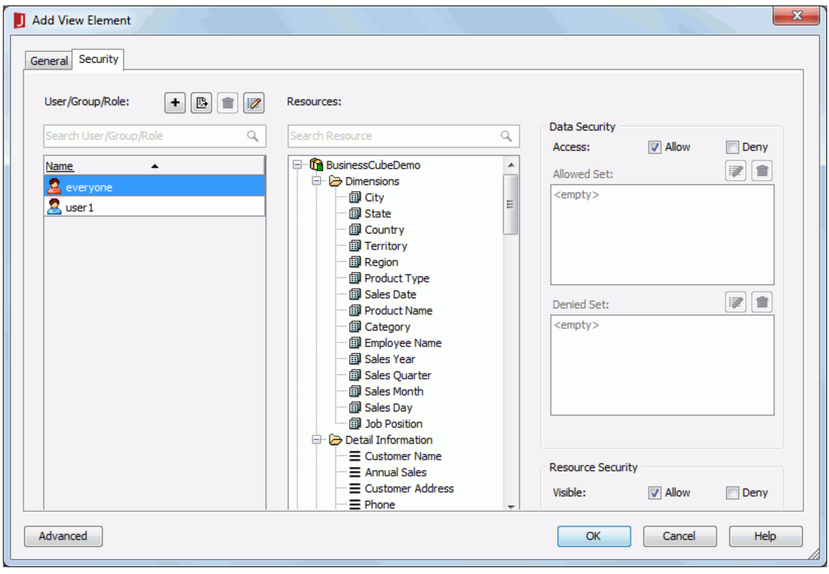
Specifies the tooltip of the view element which will be shown when you hover the mouse pointer over the element in the Resources panel of Web Report Studio or the Resource View panel of Page Report Studio.

Description

Specifies the description of the view element.

Security tab

Specifies user accessibility to members of the group objects in all the business views in current data source.



For details about options in the tab, refer to Edit Business View Security dialog.

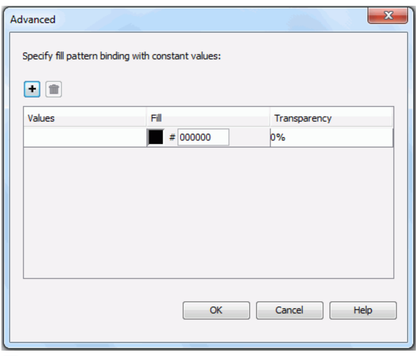
Related topics:

- Business view elements
- Adding/Editing elements

Advanced dialog

This dialog varies according to different source it is opened from.

If it is opened by clicking the Advanced button in the Color List dialog, it helps you to define certain pattern to some specific values in a chart to make the same value the same effect, and the options in the dialog are as follows. Note that this feature does not apply to surface 3-D, gauge bubble 2-D and all scatter chart types, and the settings in the dialog have higher priority than the ones in the Color List dialog.



Adds a new value line.

Removes the selected value.

Values

- If the chart has series field, specify a series value to which the fill pattern will be applied.
- If the chart has no series field, specify a category value to which the fill pattern will be applied.

Fill

Specifies the color schema for the pattern.

To specify the color, click the color image button and select a color swatch or type the hexadecimal value of the desired color (for example, 0xff0000) in the text box directly. If the color swatches cannot meet your requirement, you can click More Colors on the color palette to open the Pick a Color dialog, with which you can set a full range of color, or click More Fill Effects to specify a fill effect in the Fill Effects dialog.

Transparency

Specifies the transparency for the color.

OK

Applies the changes and closes the dialog.

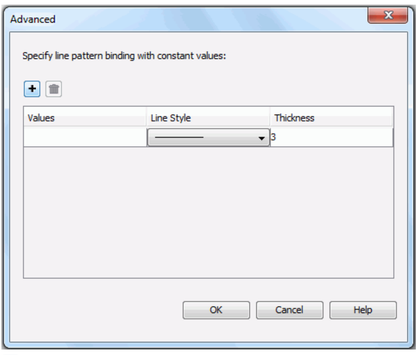
Cancel

Does not retain any changes and closes the dialog.

Help

Displays the help document about this feature.

If it is opened by clicking the Advanced button in the Line Pattern List dialog, it helps you to define certain pattern to some specific values in a chart to make the same value the same effect, and the options in the dialog are as follows. Note that this feature does not apply to surface 3-D, gauge bubble 2-D and all scatter chart types, and the settings in the dialog have higher priority than the ones in the Line Pattern List dialog.



Adds a new value line.

Removes the selected value.

Values

Specifies a series value to which the fill pattern will be applied.

Line Style

Specifies the line style for the pattern.

Thickness

Specifies the thickness for the line.

OK

Applies the changes and closes the dialog.

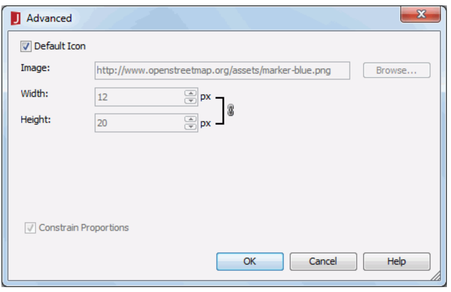
Cancel

Does not retain any changes and closes the dialog.

Help

Displays the help document about this feature.

If it is opened by clicking the button next to the Shape By text box when no field is added to the box and then the Advanced option in the Display screen of the map wizard, it helps you to specify an image as the shape of the markers for the selected group in a geographic map, and the options in the dialog are as follows.



Default Icon

Specifies to use the default icon as the shape of the markers. If you want to use your own icon, uncheck the option.

Image

Specifies the image of the markers. Input the URL for a web image or click the Browse button to specify a local image.

- **Width**
  Specifies the width of the image in inches or centimeters.
- **Height**
  Specifies the height of the image in inches or centimeters.

Shadow

Specifies the shadow image of the markers. Input the URL for a web image or click Browse button to specify a local image. Available only to Google Maps types.

- **Width**
  Specifies the width of the shadow image in inches or centimeters.
- **Height**
  Specifies the height of the shadow image in inches or centimeters.

Constrain Proportions

Specifies whether to change the width and height of the images at the same time in a certain proportion. If checked, when the width/height is changed, the height/width will also be changed in a certain proportion.

OK

Applies the changes and closes the dialog.

Cancel

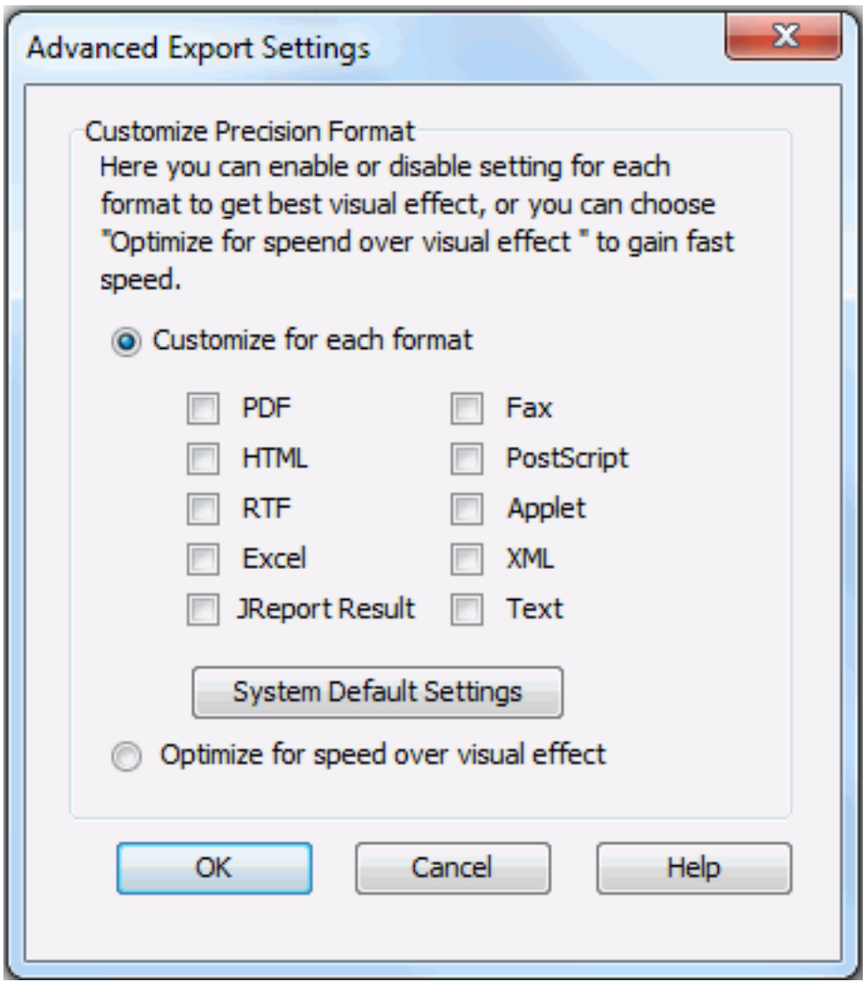
Does not retain any changes and closes the dialog.

Help

Displays the help document about this feature.

Advanced Export Settings dialog

This dialog appears when you click the Layout Precision button in the Export to category of the Options dialog. It helps you to configure the precision settings used for exporting in JReport Designer. The settings in this dialog work in the reports whose Precision Sensitive property is set to true in the Report Inspector.



The following are details about options in the dialog:

Customize for each format

For formats that are checked, the precision settings predefined for them via the System Default Settings button will take effect in the report results.

For formats that are unchecked, the precision settings predefined for them via the System Default Settings button are ignored, and the contextual precision level is applied.

- **System Default Settings**
  Opens the Precision Settings dialog to predefine precision for each format.

Optimize for speed over visual effect

If the option is checked, the contextual precision level is applied.

OK

Applies the settings and closes the dialog.

Cancel

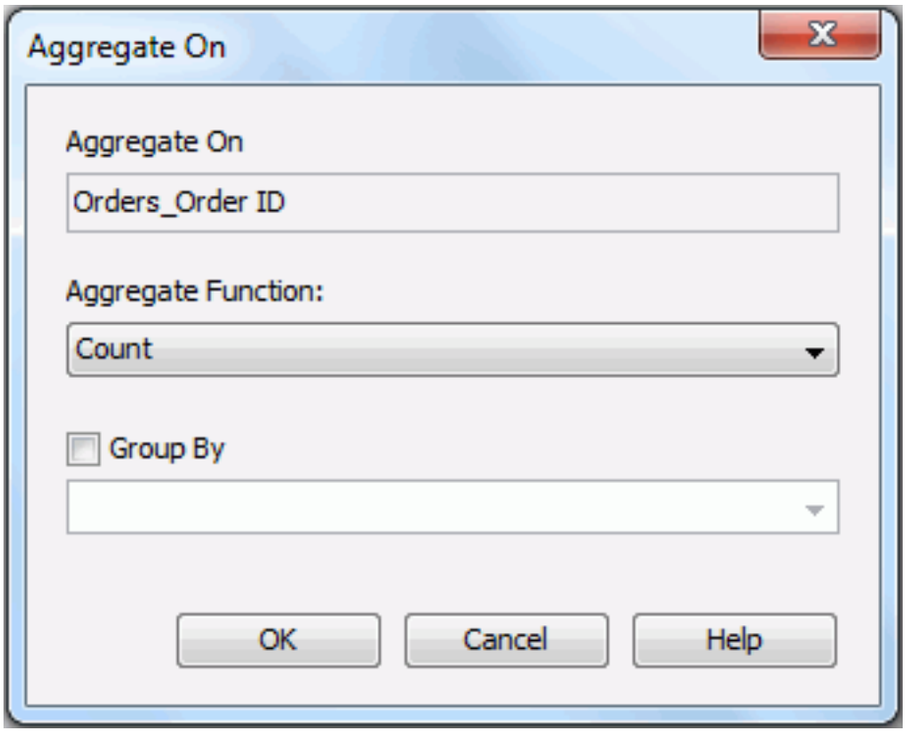
Do not retain any changes and closes the dialog.

Help

Displays the help document about this feature.

Aggregate On dialog

The dialog appears when you right-click a detail column in a table and select Aggregate On from the shortcut menu. It helps you to summarize data of the detail column.



Aggregate On

Displays the field bound with the detail column. The field is also the one on which the summary is based.

Aggregate Function

Specifies the function to compute the field in the detail column. For details about each function, see Math functions.

Group By

- If checked, you can select a group by field from the drop-down list below, to which the summary will be applied. If no field is selected, the summary will be applied to the whole dataset.
- If unchecked, a dynamic summary will be created.

OK

Summarizes the detail column data with the specified function and closes the dialog.

Cancel

Does not retain any changes and closes the dialog.

Help

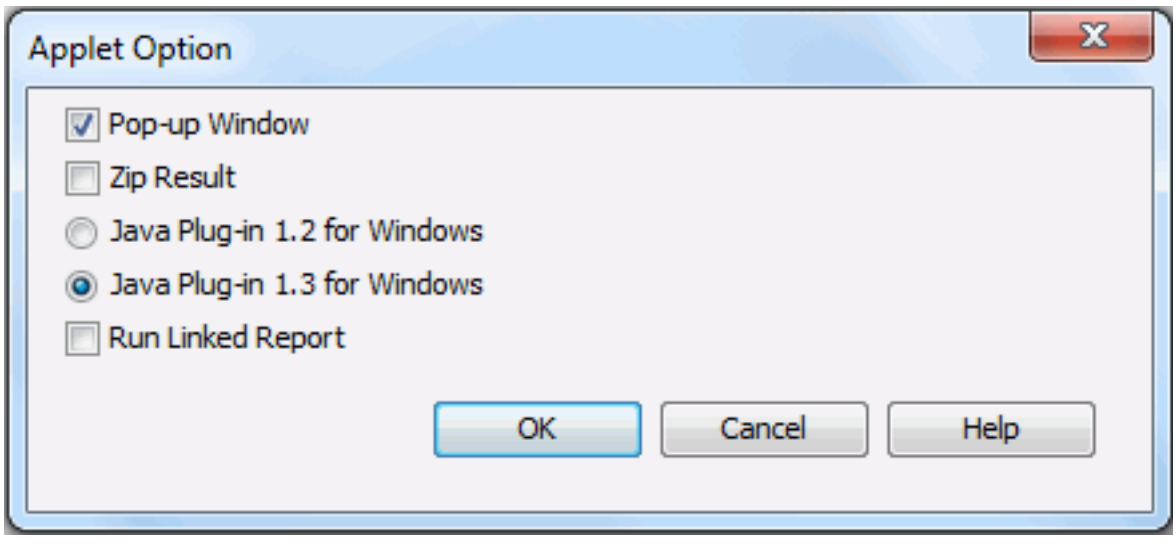
Displays the help document about this feature.

Related topics:

- Aggregating on a detail column

Applet Option dialog

The dialog appears when you set the Default Format for Viewing Report property of a report tab to Applet and then click ... in the value cell in the Report Inspector. It helps you to specify the parameters for running a report tab in Applet format.



The following are details about options in the dialog:

Pop-up Window

Specifies to show an Applet window when viewing a report.

Zip Result

Specifies to compress the result and its size would be smaller.

Java Plug-in 1.2 for Windows

Directs applets to run using Sun's 1.2 Java Runtime Environment (JRE).

Java Plug-in 1.3 for Windows

Directs applets to run using Sun's 1.3 Java Runtime Environment (JRE).

Run Linked Report

If the report is linked with other reports, you can check whether or not to generate the linked reports in the Applet result. If you are only interested in the primary report, leave this option unchecked. Generating linked reports at the same time, especially when the linked reports contain a large amount of data, will cause performance issue.

Note: You'd better specify a file destination when exporting a report via Applet, such as `D:\folder\filename.pdf` in case you may not find the result. By default when exporting a report via Applet on Internet Explorer, the exported result will be located on the Desktop, and when on Firefox, it will be in its installation root.

If you do not have the Read or Write permission to the default location or the location you specified, you will get an access denied error. To handle this, add the required permission in jdk\jre\lib\security\java.policy that the applet uses:

For the default location "Desktop":

```
permission java.util.PropertyPermission "user.dir", "read";
permission java.io.FilePermission "${user.home}${/}Desktop${/}*", "read,write";
```

For user defined location:

```
permission java.io.FilePermission "D:${/}temp${/}*", "read,write";
```

OK

Applies the changes and closes the dialog.

Cancel

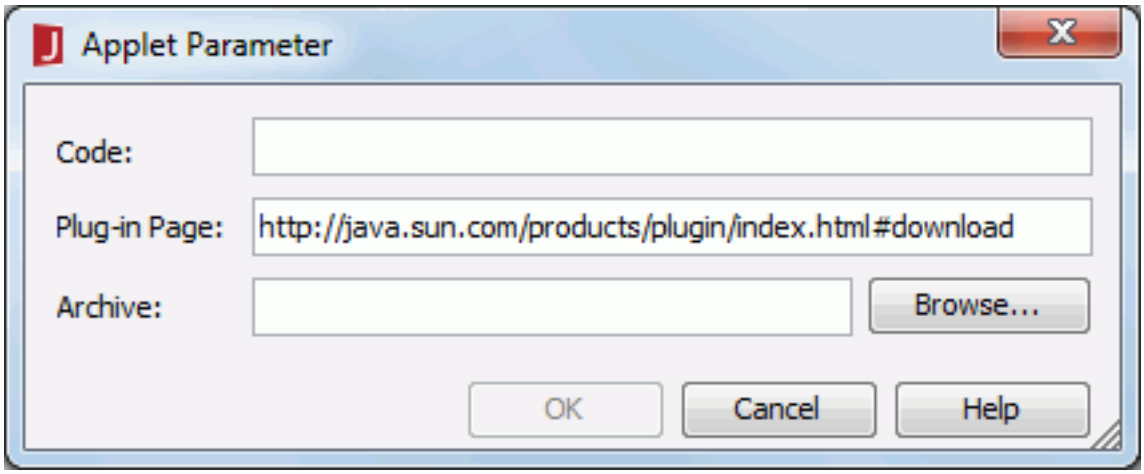
Cancels the changes and exits the dialog.

Help

Displays the help document about this feature.

Applet Parameter dialog

This dialog appears when you click Insert > Multimedia Objects > Applet on the menu bar, or drag Applet from the Toolbox panel to a report. It enables you to insert an applet into your report.



The following are details about options in the dialog:

Code

Specifies the name for the class file which is contained in this archive file.

Plug-in Page

Specifies the URL from which you can download JDK to run this applet.

Archive

Specifies the position, where locates the resources that can make the applet run in this report properly.

OK

Accepts all changes and closes the dialog.

Cancel

Does not retain any changes and closes the dialog.

Help

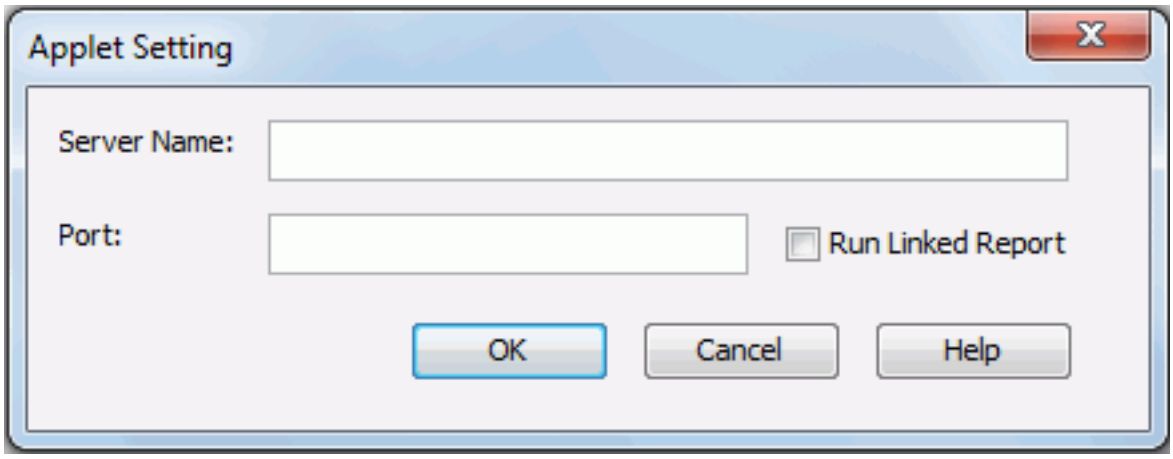
Displays the help document about this feature.

Related topics:

- Inserting an applet object

Applet Setting dialog

This dialog appears when you click File > Export To > Applet on the menu bar. It helps you to specify the server information for exporting the results of a report as an applet to the server.



The following are details about the options in the dialog:

Server Name

Specifies the name of the server, for example, 222.222.222.109.

Port

Specifies the port number of the server, for example 8888.

Run Linked Report

If the report is linked with other reports, you can check whether or not to generate the linked reports in the exported Applet file. If you are only interested in the primary report, leave this option unchecked. Generating linked reports at the same time, especially when the linked reports contain a large amount of data, will cause performance issue.

OK

Applies all changes and opens the Export to applet dialog to specify the directory that you want to save this Applet file.

Cancel

Cancels all changes and closes the dialog.

Help

Displays the help document about this feature.

Related topics:

- Exporting report results - To Applet

Banded Wizard

This wizard appears when you select the Banded layout in the New Page Report Tab dialog or New Page Report dialog and then click OK, or right-click a banded object and click Banded Wizard on the shortcut menu. It helps you to create or modify a banded object.

The wizard consists of the following screens:

- Data screen
- Display screen
- Group screen
- Summary screen
- Chart screen
- Filter screen
- Style screen

Back

Goes back to the previous screen.

Next

Goes to the next screen.

Finish

Finishes creating or modifying the banded object and closes this wizard.

Cancel

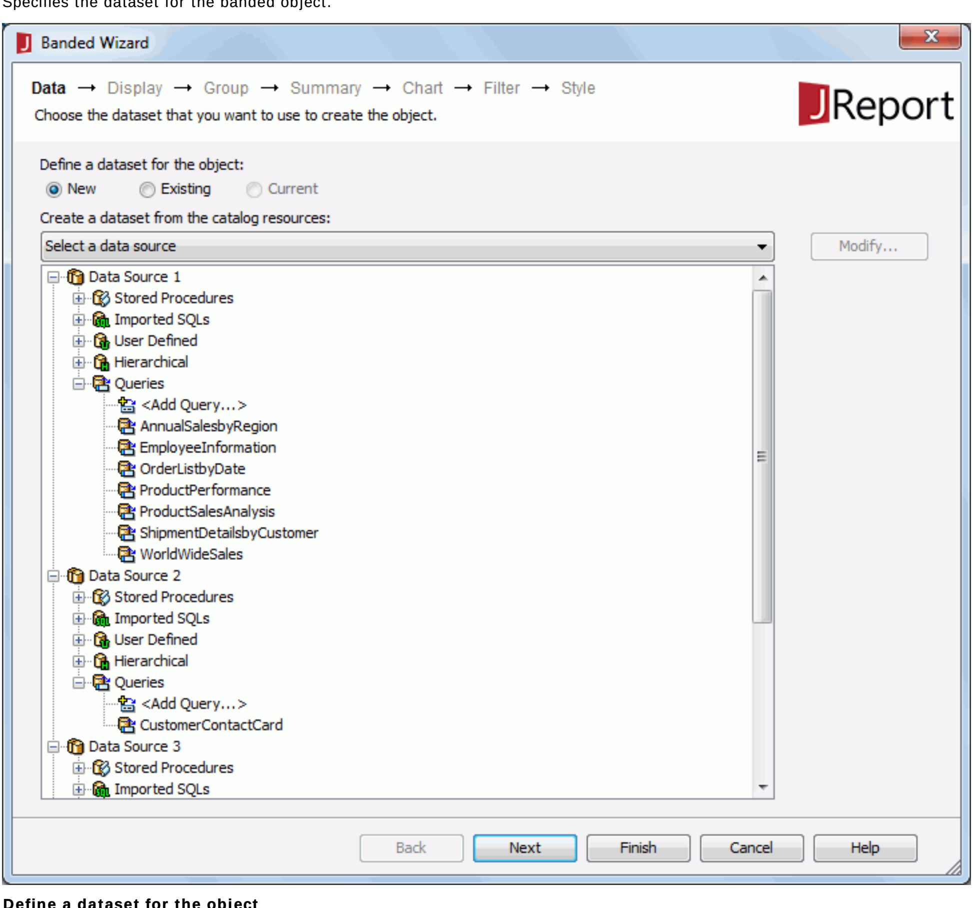
Does not retain changes and closes this wizard.

Help

Displays the help document about this feature.

Data screen

Specifies the dataset for the banded object.



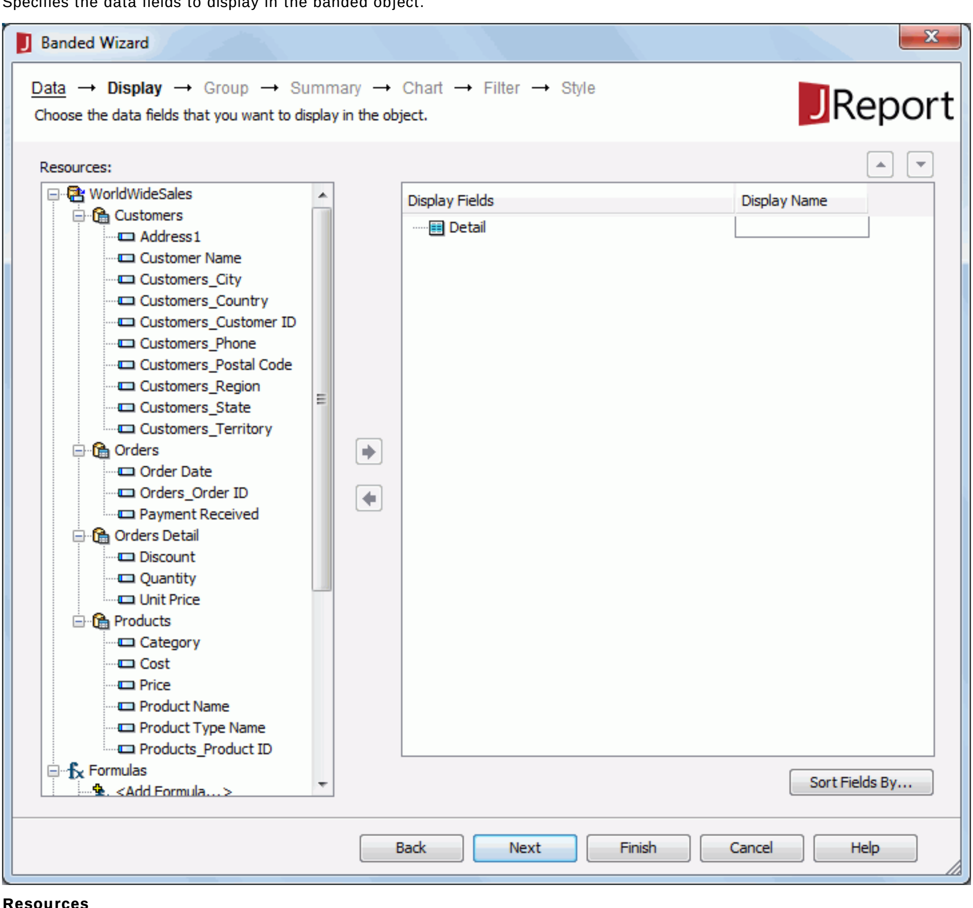
Define a dataset for the object

Specifies the dataset you want to use to create or modify the banded object.

- **New**
  If checked, select a data source from the catalog resources to create a dataset that will be used to build the banded object. When you choose to create the dataset from a query, you can click the Modify button to edit the query in the Query Editor if required.
- **Existing**
  If checked, select a dataset from the ones existing in the open report to create the banded object. Click the Modify button to edit the dataset in the Dataset Editor if required.
- **Current**
  If checked, the current dataset used by the parent object will be applied to the banded object.

Display screen

Specifies the data fields to display in the banded object.



Resources

Lists all the available data resources.

Adds the specified field to use in the banded object.

Removes the specified field that is not required in the banded object.

Replaces the selected field in the banded object with the specified field in the Resources box.

Moves the specified field one step up.

Moves the specified field one step down.

Show Banded Group Structure

Specifies whether to show the group structure of the banded object.

Display Fields

Lists the fields that you have selected to display in the banded object.

Display Name

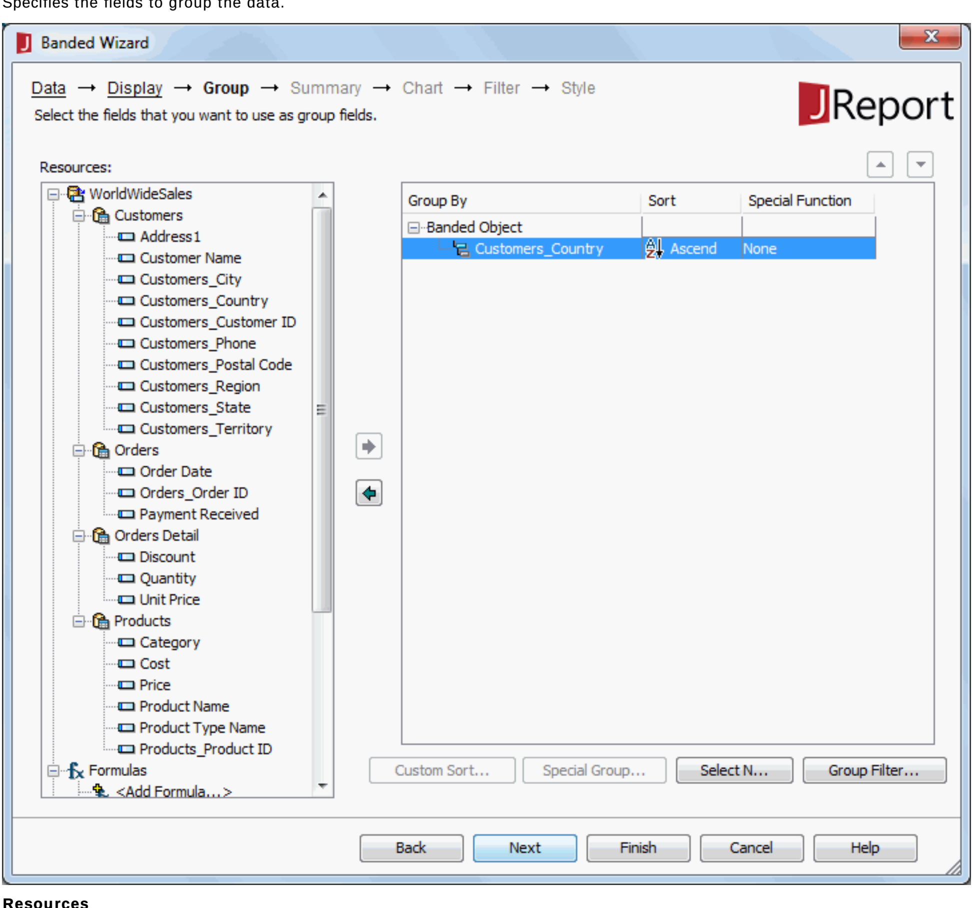
Shows the display names of the selected fields. You can double-click the name cells to edit them if required.

Sort Fields By

Opens the Sort Fields By dialog to specify how to sort data in the banded object.

Group screen

Specifies the fields to group the data.



Resources

Lists all the available data resources.

Adds the selected field as the group by field in the banded object.

Removes the selected group by field that is not required.

Replaces the selected group by field in the banded object with the specified field in the Resources box.

Moves the specified group one step up.

Moves the specified group one step down.

Group By

Lists the fields that are used to group data in the banded object.

Sort

Specifies how groups at the specific group level will be sorted.

- **Ascend**
  Groups will be sorted in an ascending order (A, B, C).
- **Descend**
  Groups will be sorted in a descending order (C, B, A).
- **No Sort**
  Groups will be sorted in the original order in database.
- **Special Group**
  Opens the User Defined Group dialog to define grouping information.

  For example, if you placed a field named Region for grouping, and this field contains all 50 states of the United States; and if you want to see the data between Maryland (MD) and New York (NY), you can define the criteria by selecting the between operator to further define your grouping information.
- **Custom Sort**
  Opens the Custom Sort dialog to set how groups will be sorted.

Special Function

If the group by field is of Numeric/String/Date/Time type, you can select a special function for the field in the Special Function column to further specify to which level the data will be grouped by.

If Customize is selected, the Customized Function dialog will be displayed, in which you can set the function by your own.

Custom Sort

Specifies how to sort the groups. Activated only when you have selected Custom Sort from the Sort column to define the sorting manner of groups for the selected group level.

Special Group

Specifies how to group your information. Activated only when you have clicked Special Group from the Sort column to define a special group.

Select N

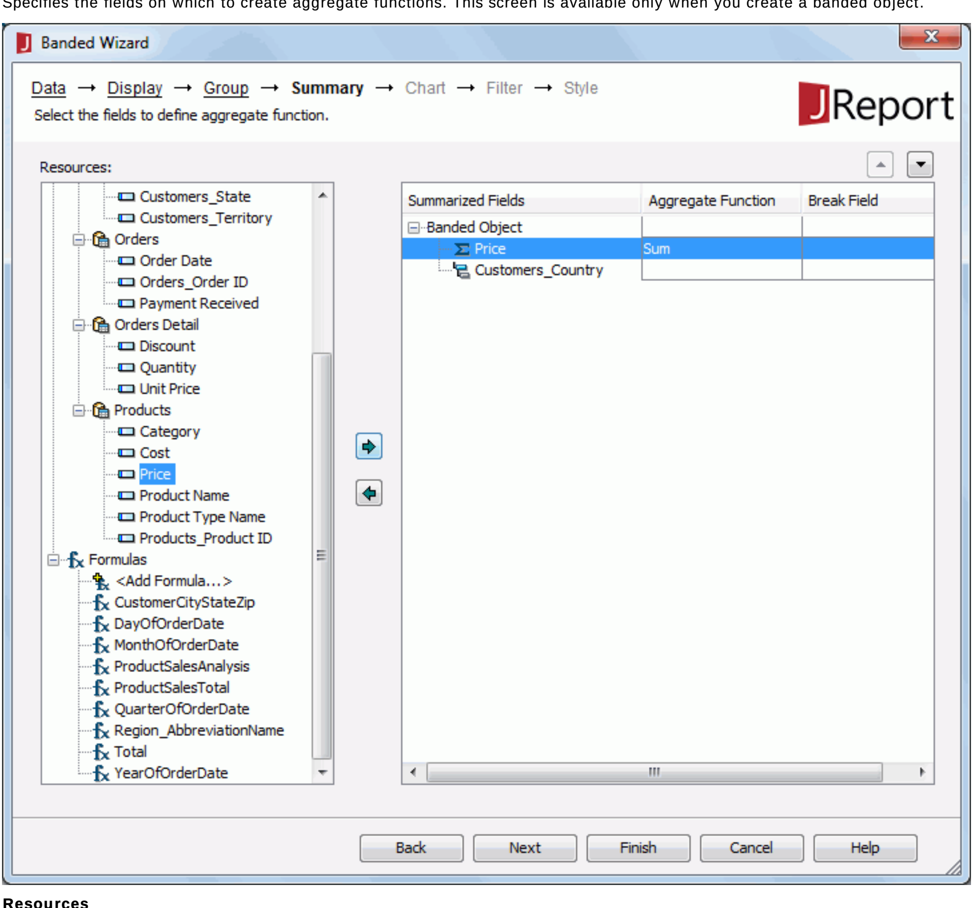
Opens the Select N dialog to specify the Select N condition.

Group Filter

Opens the Group Filter dialog to specify the group filter condition.

Summary screen

Specifies the fields on which to create aggregate functions. This screen is available only when you create a banded object.



Resources

Lists all the available data resources.

Adds the selected field as the summary field to the banded object.

Removes the selected summary field that is not required.

Moves the specified summary field one step up.

Moves the specified summary field one step down.

Summarized Fields

Lists all the fields that you want to display and to be summarized in the banded object.

Aggregate Function

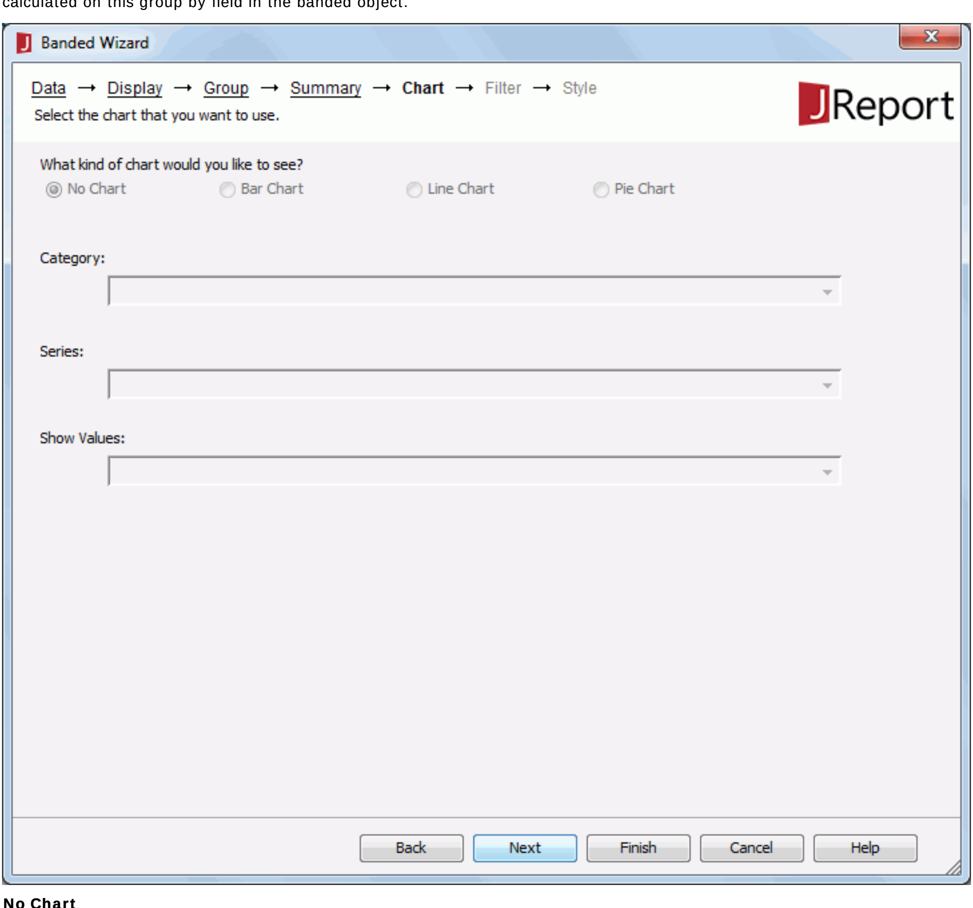
Specifies the function to summarize the selected field.

Break Field

Displays a defined field on which the summary will be calculated. If a summary field is added below the Banded Object node, the break field is null and the summary will be calculated on the whole dataset.

Chart screen

Specifies to create a chart together with the banded object, which will be placed in the banded header panel. This screen is available only when you create a banded object, and when there is at least one group by field and one summary field which is calculated on this group by field in the banded object.



No Chart

Specifies not to create a chart.

Bar Chart

Specifies to create a Clustered Bar 2-D chart together with the banded object.

Line Chart

Specifies to create a Line 2-D chart together with the banded object.

Pie Chart

Specifies to create a Clustered Pie chart together with the banded object.

Category

Lists the group by fields of the banded object on which the summary fields are calculated. Choose the field you want to display on the category (X) axis of the chart from the drop-down list.

Series

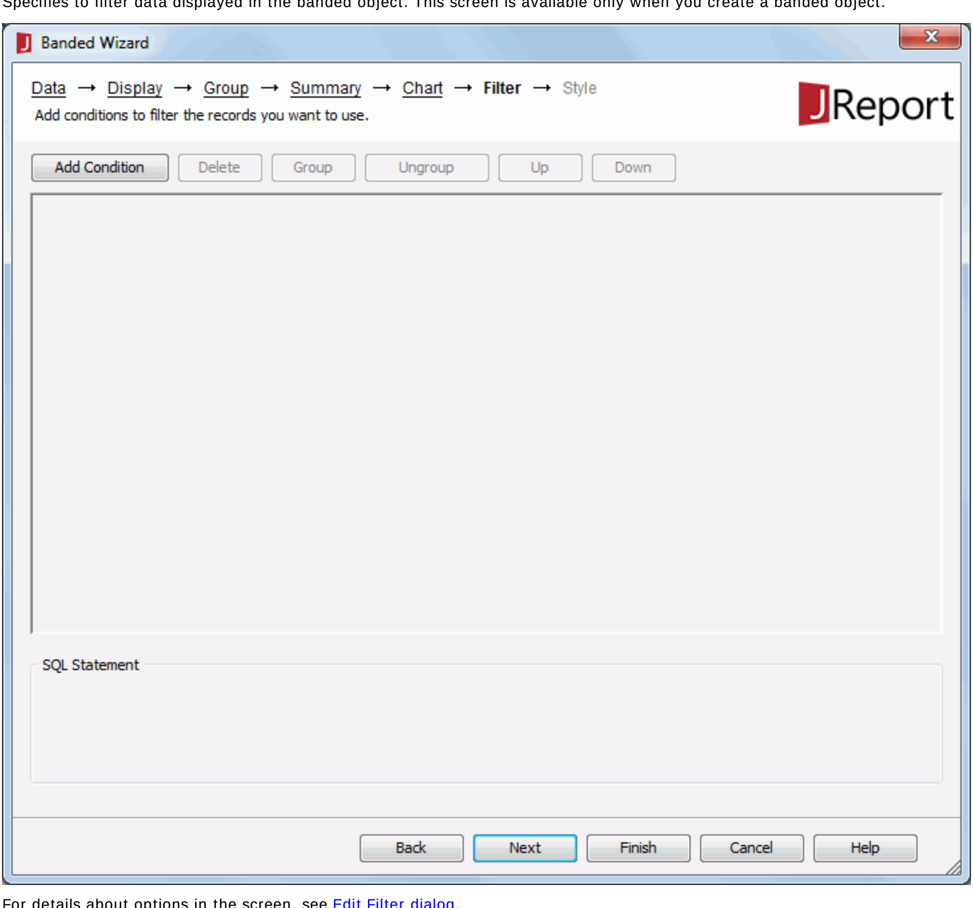
Lists the fields that have been added as the group by fields of the banded object. Choose the field you want to display on the series (Z) axis of the chart from the drop-down list.

Show Values

Lists the summary fields which are calculated based on the field you choose to display on the category axis of the chart. Choose the value you want to display in the chart from the drop-down list.

Filter screen

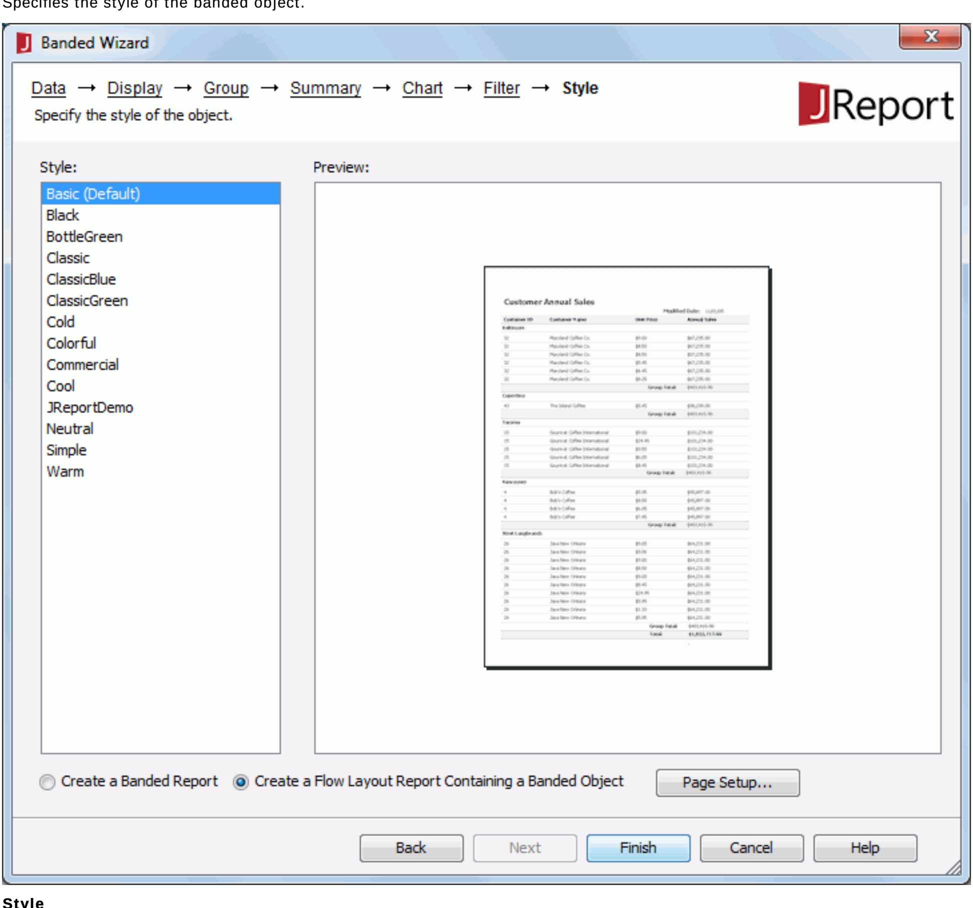
Specifies to filter data displayed in the banded object. This screen is available only when you create a banded object.



For details about options in the screen, see Edit Filter dialog.

Style screen

Specifies the style of the banded object.



Style

Lists all the available styles that can be applied to the banded object.

- **<Custom>**
  There is no any style information on it and it is only used to cover previous-version report which did not bind any style or the bound style cannot be found in the style list.

Preview

Displays a diagram illustrating the effect of the selected style on the banded object.

Inherit Style

Specifies whether to make the banded object take the style of its parent. This option is available only when you modify a banded object and the banded object is inserted into another banded object.

Create a Banded Report

Creates a restricted version of a flow layout report, one that can contain a single banded object only. This layout option is intended to provide compatibility with JReport Version 7 report templates.

Create a Flow Layout Report Containing a Banded Object

Creates a report that contains a banded object mapped to the specified dataset. This layout option is specified by default.

Page Setup

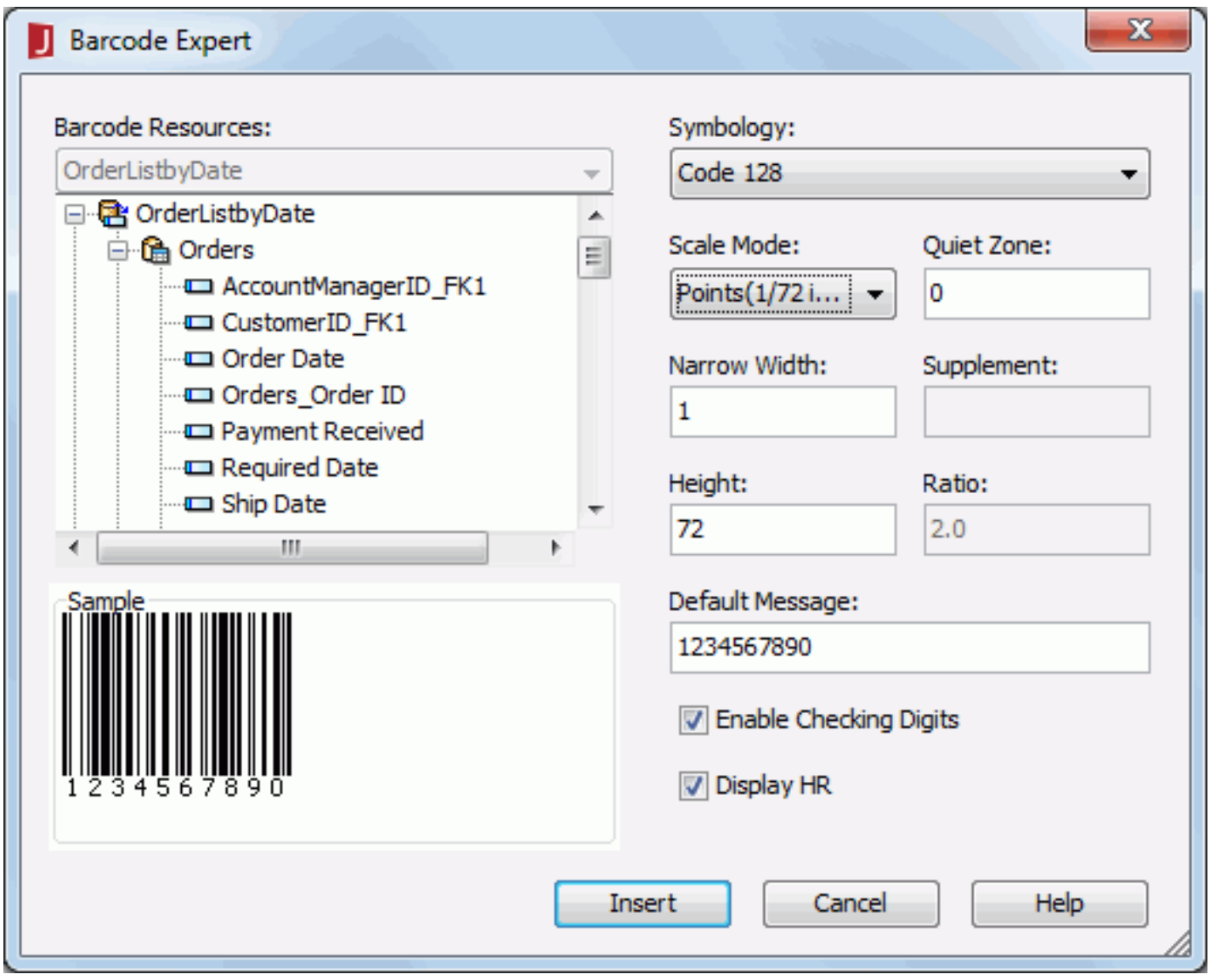
Opens the Page Setup dialog to specify page properties. Available only when creating a page report.

Related topics:

- Creating a banded report
- Modifying a banded object

Barcode Expert dialog

This dialog appears when you click Insert > Barcode on the menu bar, or the Insert Barcode button on the Insert toolbar. It helps you to create and edit a barcode object and then insert it into your report.



The following are details about options in the dialog:

Barcode Resources

Lists the resource that can be used as the data source for the barcode. The data source of a barcode can be database data, formulas, parameters, or a string typed by a user.

Symbology

Specifies the barcode type. You can choose from the following: UPC-A, UPC-E, EAN-13, EAN-8, Code39, Code128 and Codabar.

Scale Mode

Specifies the unit for values of Quiet Zone, Narrow Width, Supplement, Height, and Ratio. Can be Mils (0.001 inch), Metric (0.01 millimeter), and Points (1/72 inch).

Quiet Zone

Specifies the space around the barcode where there are no dark marks. Most of symbology requires a quiet zone which precedes the following barcode.

Narrow Width

Specifies the barcode bar width.

Supplement

Specifies the supplement (if necessary). Some barcode types, such as CODE39, CODE128 and Codabar, do not have supplements.

Height

Specifies the height of the barcode.

Ratio

Specifies the width ratio of the thick bar to the thin bar in the CODE39/ Codabar barcode. The ratio box can have only 2 effective values, 2.0 and 3.0. Any ratio values that are not equal to 2.0 or 3.0 will be regarded as 2.0. Enabled only for CODE39 and Codabar.

Default Message

Specifies the default value for the barcode.

Enable Checking Digits

If checked, the checking digits will be enabled.

Display HR

Specifies whether to display the barcode numbers together with the barcode. Enabled only for CODE39, CODE128, and Codabar.

Insert

Accepts the changes and inserts the barcode into your report.

Cancel

Does not retain the changes and closes the dialog.

Help

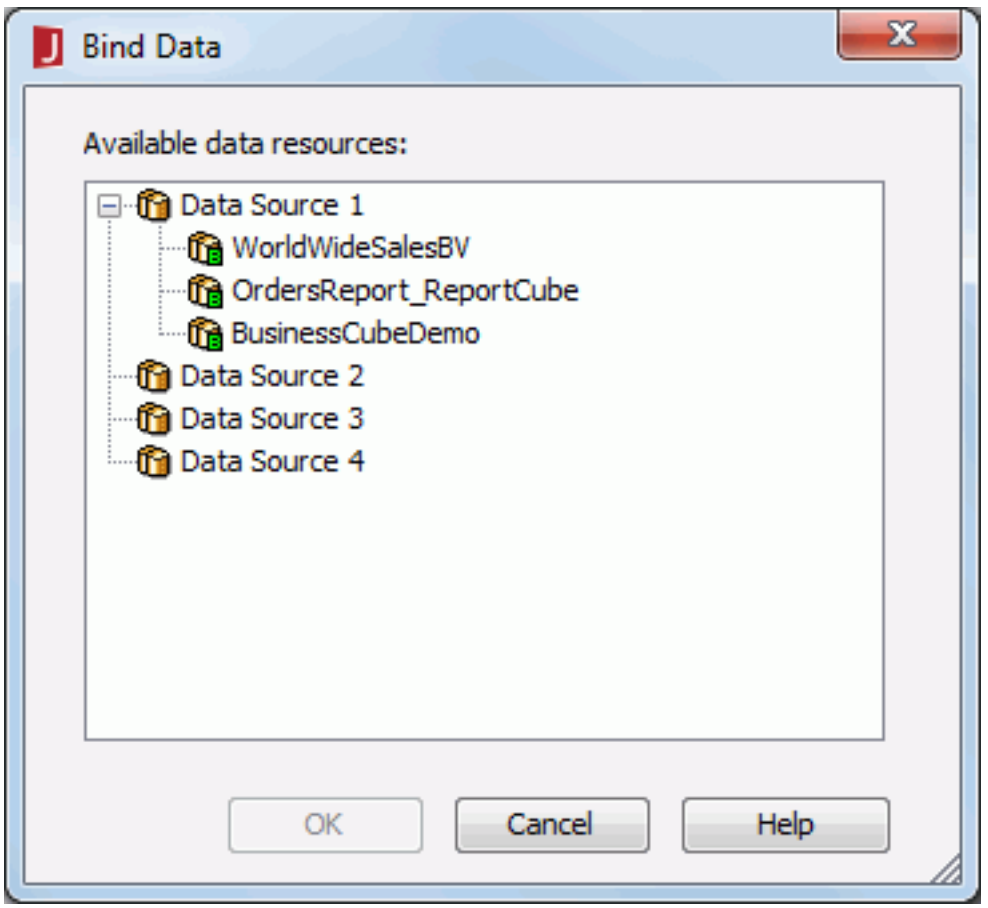
Displays the help document about this feature.

Related topics:

- Component - Barcodes

Bind Data dialog

This dialog appears when you click Edit > Bind Data on the menu bar. It helps you to bind a data source to a web report or library component and then create formulas based on this data source in order to control the Invisible property of objects in the web report or library component.



The following are details about options in the dialog:

Available data resources

Lists all the business views in the current catalog that can be used as data source of the web report or library component. Select the one you want to use.

OK

Binds the business view to the web report or library component.

Cancel

Does not retain the changes and closes the dialog.

Help

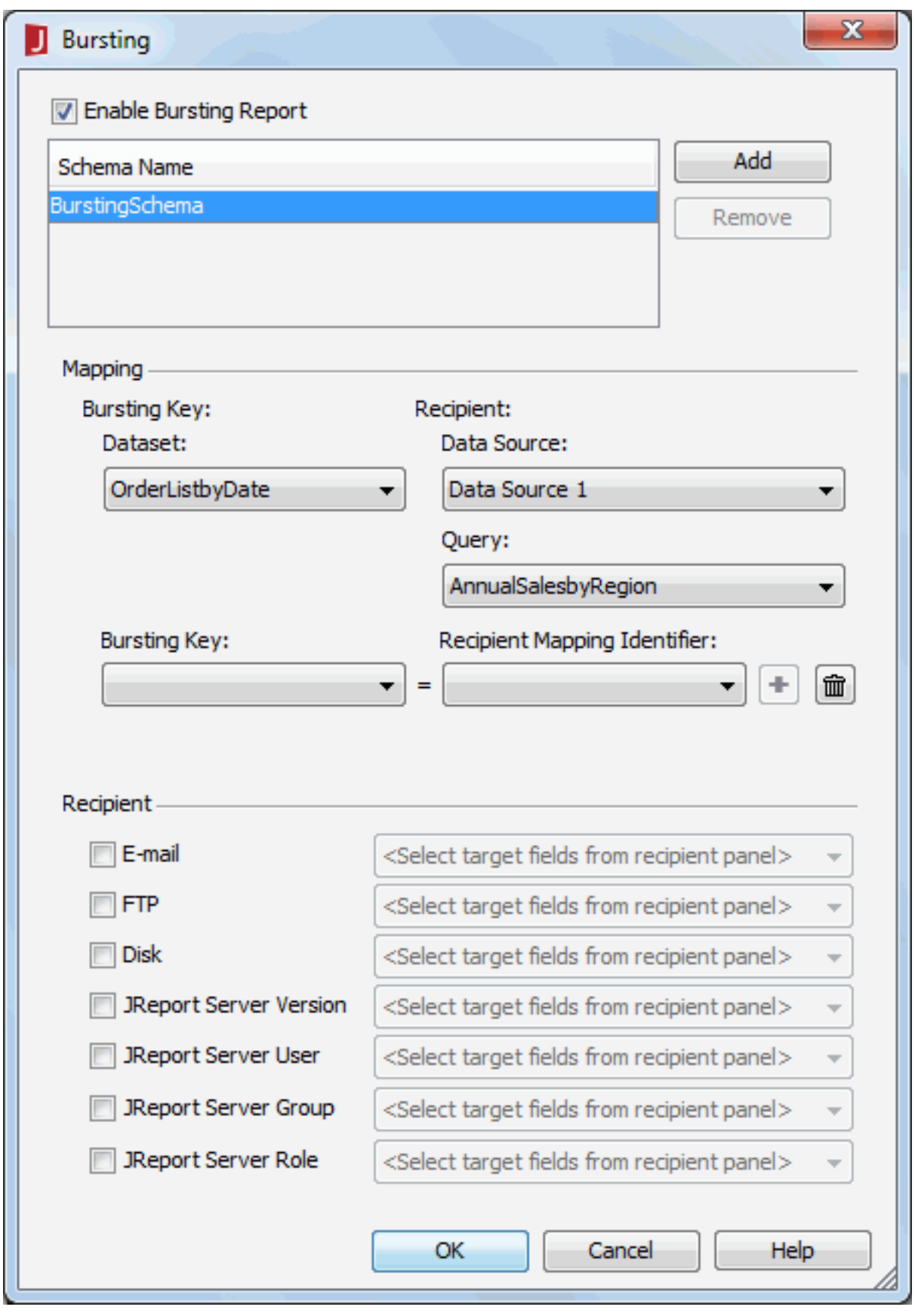
Displays the help document about this feature.

Related topics:

- Using formulas to control showing or hiding components

Bursting dialog

The dialog appears when you click Report > Bursting on the menu bar. It helps you to make a normal report become a bursting report.



Enable Bursting Report

Specifies whether to enables the bursting options.

Schema Name

Lists the names of the bursting schemas for the report.

When a schema name is selected, the schema's bursting information is displayed in the Mapping and Recipient sections below, and by double-clicking a schema name you can rename it.

Add

Creates a bursting schema for the report. You need define the schema's Mapping and Recipient information.

Remove

Removes a selected schema in the Schema Name box.

Mapping

Specifies the relationship between the bursting key and recipient for the selected schema.

- **Bursting Key**
  - **Dataset**
    Specifies the dataset used in the report based on which to set the bursting key. Multiple bursting schemas for a report can be based on different datasets in the report.
  - **Bursting Key**
    Specifies DB columns according to which to split report data. The available values are columns in the specified dataset.
- **Recipient**
  - **Data Source**
    Specifies the data source that contains the recipient query.
  - **Query**
    Specifies the query from the selected data source which retrieves recipient addresses. Multiple bursting schemas for a report can be based on different recipient queries in the report.
  - **Recipient Mapping Identifier**
    Specifies a unique recipient column to map to each bursting key column. The available values are columns in the specified query of the same data type as the bursting key.

    The recipient columns should contain all desired information of addresses which will be used in recipient.
-
  Adds a row to map another pair of bursting key column and recipient column.
-
  Removes the row of mapping.

Recipient

Defines the formats of recipient addresses for the selected schema.

- **E-mail**
  Specifies the column that defines E-mail addresses.
- **FTP**
  Specifies the column that defines FTP addresses.
- **Disk**
  Specifies the column that defines file system directories.
- **JReport Server Version**
  Specifies the column that defines resource directories in the JReport Server resource system.
- **JReport Server User**
  Specifies the column that defines user addresses in the JReport Server security system.
  - **User E-mail**
    If checked, the report results will be sent to e-mail addresses.
  - **User Private Folder**
    If checked, the report results will be sent to the users' My Reports folders in the server resource tree and you are allowed to give a sub path.
    - **Sub Path**
      If checked, specify a sub path in the My Reports folder and the report result will be sent to the sub path.
- **JReport Server Group**
  Specifies the column that defines group addresses in the JReport Server security system.
  - **User E-mail**
    If checked, the report results will be sent to e-mail addresses.
  - **User Private Folder**
    If checked, the report results will be sent to the group members' My Reports folders in the server resource tree and you are allowed to give a sub path.
    - **Sub Path**
      If checked, specify a sub path in the My Reports folder and the report result will be sent to the sub path.
- **JReport Server Role**
  Specifies the column that defines role addresses in the JReport Server security system.
  - **User E-mail**
    If checked, the report results will be sent to e-mail addresses.
  - **User Private Folder**
    If checked, the report results will be sent to the role members' My Reports folders in the server resource tree and you are allowed to give a sub path.
    - **Sub Path**
      If checked, specify a sub path in the My Reports folder and the report result will be sent to the sub path.

OK

Accepts all changes and closes the dialog.

Cancel

Does not retain any changes and closes the dialog.

Help

Displays the help document about this feature.

Related topics:

- Report Bursting

Business View Editor window

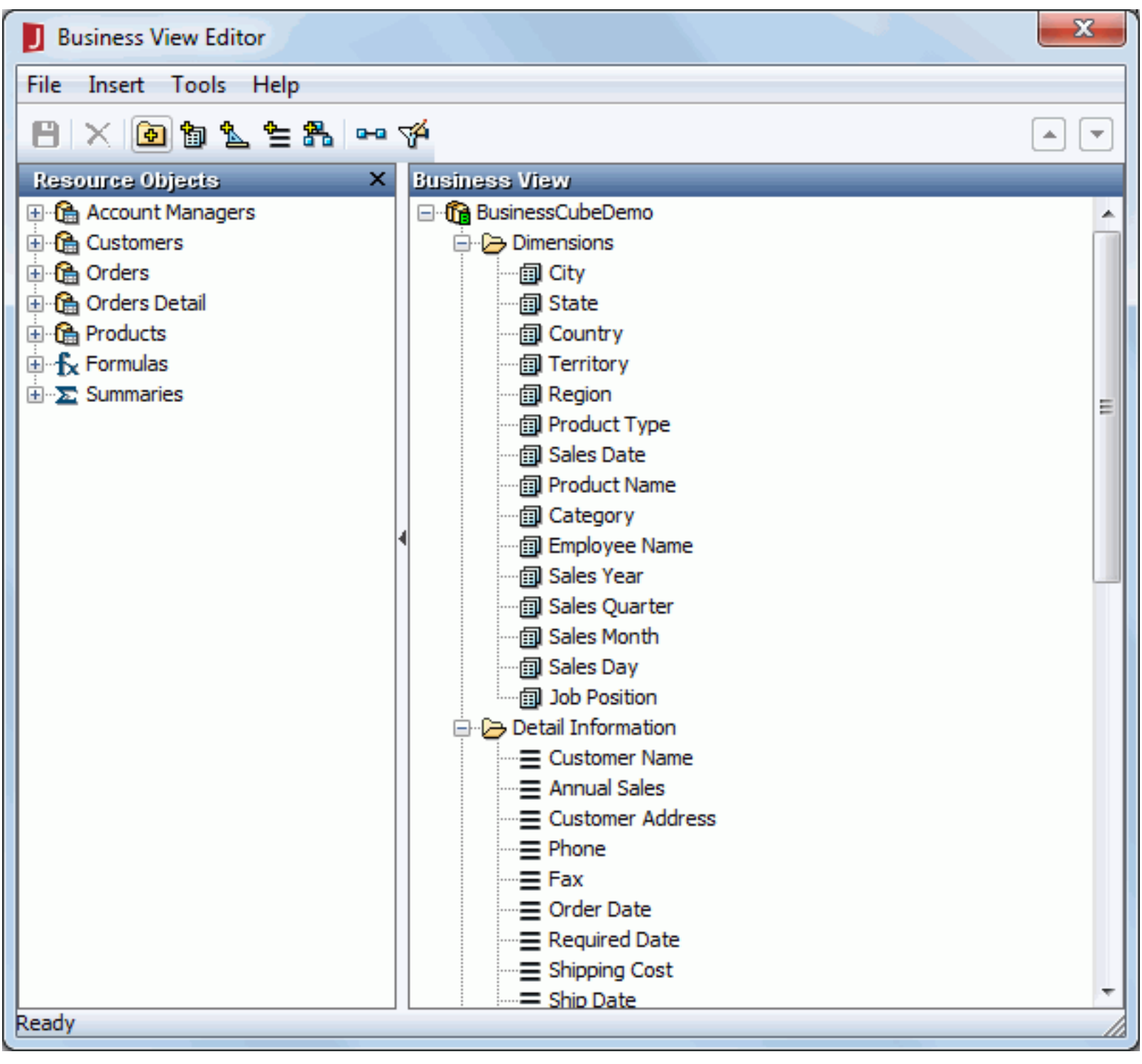
This window helps you to create or edit a business view. It appears when you do one of the following:

- Right-click the blank Business View node in the Catalog Browser, select Add Business View from the shortcut menu, input a name in the displayed dialog and click OK. Then in the Add Table/View/Query dialog, select one resource and then click OK.

  Or if you select more than one resource in the Add Table/View/Query dialog and click OK, the Query Editor is displayed. Select the required table columns and join the tables and click OK.

- In the Catalog Browser right-click a business view and select Edit from its shortcut menu.

- Right-click a business view or an element in it and select Edit from the shortcut menu in the Resources box of the Display screen in the map wizard.



The following are details about options in the window:

File

- **Save**
  Saves the business view and applies all the changes made in this window.

- **Save As**
  Saves a copy of the business view with a different name.

- **Close**
  Exits this window.

Insert

- **Category**
  Opens the Category Property dialog to insert a new folder into the business view.

- **Group**
  Opens the Add View Element dialog to insert a group object into the business view.

- **Aggregation**
  Opens the Add View Element dialog to insert an aggregation object into the business view.

- **Detail**
  Opens the Add View Element dialog to insert a detail object into the business view.

- **Hierarchy**
  Prompts you to provide a name for a new hierarchy and then adds a hierarchy into the business view resource tree. The name should be unique among the hierarchies in a business view.

Tools

- **Security Configuration**
  Opens the Edit Business View Security dialog to configure security for all the business views in the data source.

- **Join Editor**
  Opens the Query Editor window to edit the self contained query for the business view.

- **Predefined Filter**
  Opens the Predefined Filter dialog to add predefined filters to the business view.

Help

- **Business View Help**
  Displays the help document about this feature.

Toolbar

- **Save**
  Saves the business view and applies all the changes made in this window.

- **Delete**
  Deletes the selected categories, groups, aggregations or details from the business view.

- **Insert Category**
  Inserts a new folder into the business view.

- **Insert Group**
  Inserts a group object into the business view.

- **Insert Aggregation**
  Inserts an aggregation object into the business view.

- **Insert Detail**
  Inserts a detail object into the business view.

- **Insert Hierarchy**
  Prompts you to provide a name for a new hierarchy and then adds a hierarchy into the business view resource tree. The name should be unique among the hierarchies in a business view.

- **Join Editor**
  Opens the Query Editor window to edit the self contained query for the business view.

- **Predefined Filter**
  Opens the Predefined Filter dialog to add predefined filters to the business view.

- **Move Up**
  Moves the selected object one step up.

- **Move Down**
  Moves the selected object one step down.

Resource Objects

Lists all the resources that can be inserted into the business view.

Business View

You can add or edit the resources of a business view in this panel. The resources are arranged in a tree structure. Under the root category, categories, view elements and hierarchies can be added. Categories are used as virtual folders to hold sub categories and view elements. View elements and hierarchies can be added in any categories including the root category.

For how to add and edit elements in a business view, see Adding/Editing elements.

For how to add hierarchies, see Defining hierarchies.

To change the display name of an element or the name of a hierarchy, use the Rename option on its shortcut menu.

To adjust the order of an element or a hierarchy, select it and click or on the toolbar. For an element, you can also make use of the Move Up and Move Down options on the shortcut menu.

To delete one or more elements or hierarchies, first select them, then click on the toolbar or use the Delete option on the shortcut menu. For an element, you can also drag it to the left panel to remove it. The root category cannot be removed.

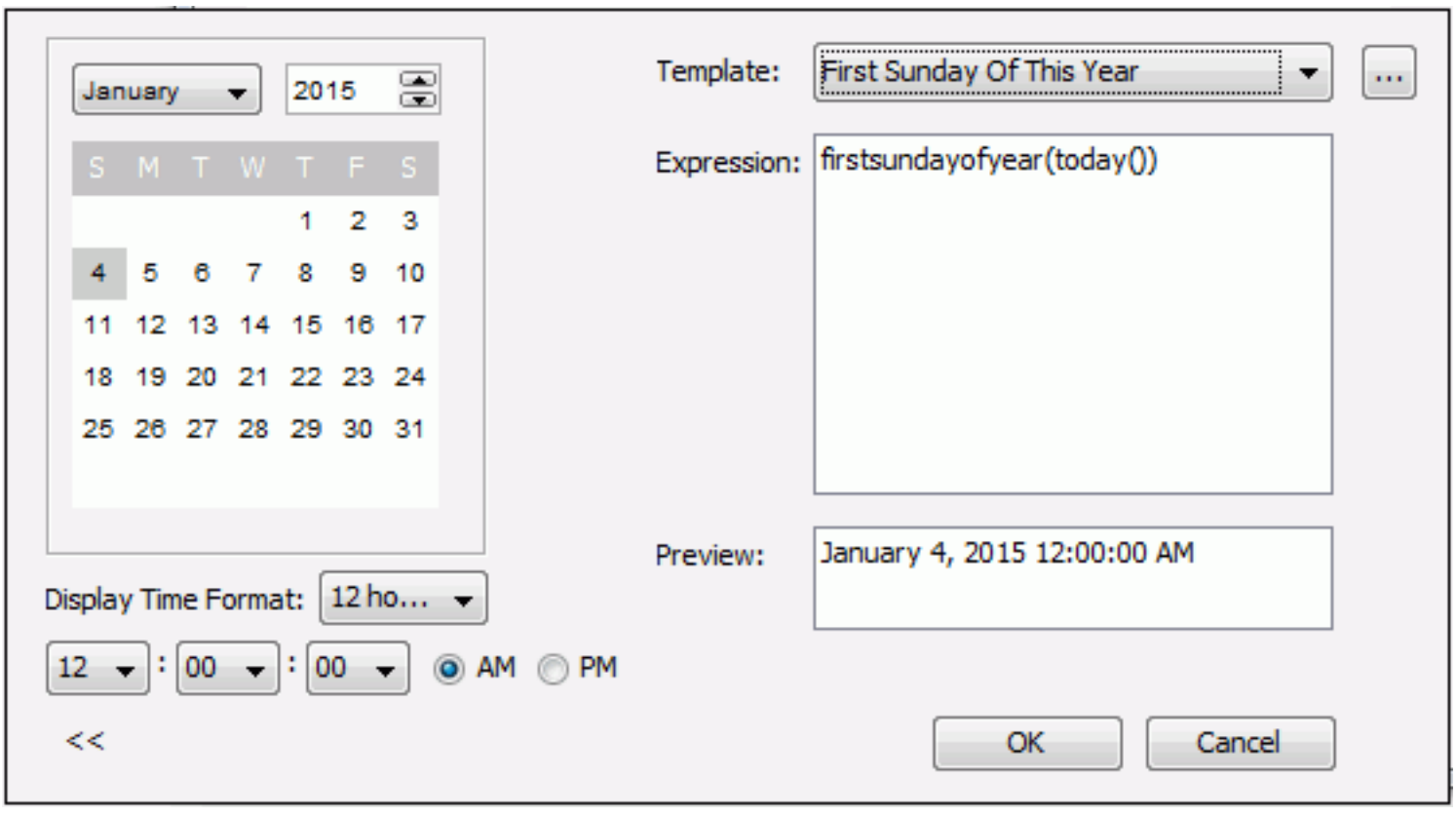
Related topics:

- Editing a business view

Calendar dialog

The dialog appears when you click the calendar button while specifying value to a Date, DateTime, or Time parameter which allows for type-in values. It helps you to create a date and time value using either the calendar or an expression.

The dialog is divided into two sections: the left section is a calendar for selecting date and time, the right section is for editing an expression by using built-in formula functions. The expression has higher priority than the value specified in the calendar. If you cannot see the expression settings, click **>>** at the bottom left corner.



Calendar

Specifies the month, year, day and time from the calendar. The time setting is available to DateTime or Time type parameters. The date/time in the calendar can be synchronized with the date/time calculated by a valid expression.

Display Time Format

Specfies in which format to display the time, 12-hour or 24-hour. It is available to DateTime type parameters only. Each time when the calendar is accessed, the default selected format is always 12-hour format.

>>/<<

Displays or hides the expression settings.

Template

You can choose an existing expression template and then either use it directly or edit your expression based on it. The Template name field does not support typing directly into it. If you would like to create an expression all by yourself, select the first blank row from the drop-down list.

Expression

Displays the expression. You can edit the expression directly in the text box.

Opens the Expression dialog to insert built-in Date/Time formula functions into the expression.

Preview

Displays the value result based on the specified expression, or gives error information if the expression is not correct.

OK

Sets either the value selected in the calendar or the expression as the parameter value. Expression has higher priority.

Cancel

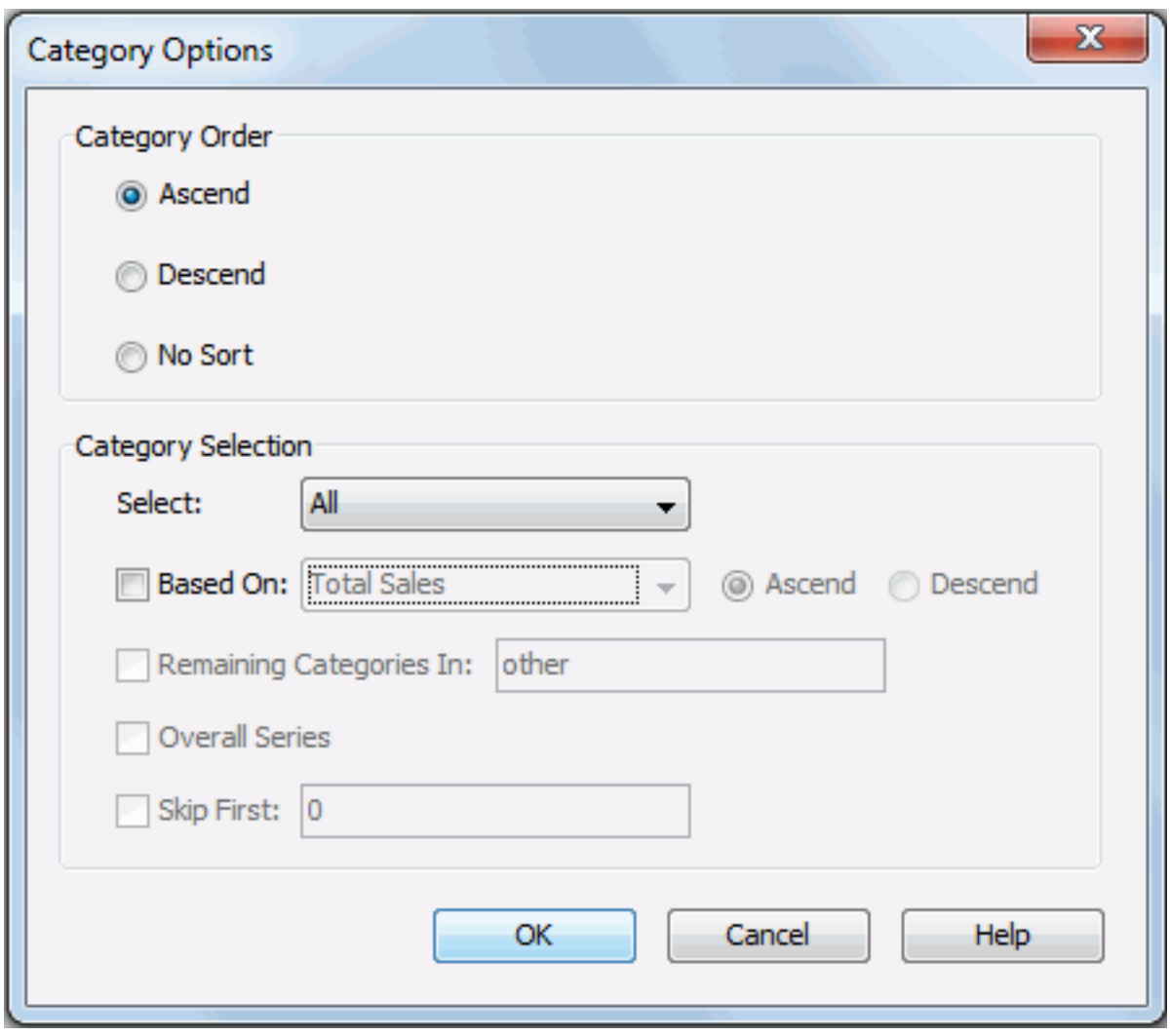
Does not retain any changes and closes the dialog.

Related topics:

- Using built-in functions to set date and time parameter values

Category Options dialog

The dialog appears when you click the Order/Select N button below the Category box in the Display screen of the chart wizard. It helps you to specify the sort order of the category values and define the number of the category values that will be displayed in a chart.



The following are details about options in the dialog:

Category Order

Specifies in which manner to sort the category values.

- **Ascend**
  Values will be sorted in an ascending order (A, B, C).
- **Descend**
  Values will be sorted in a descending order (C, B, A).
- **No Sort**
  Values will be sorted in the original order in database.

Category Selection

Specifies the number of the category values that will be displayed in the chart.

- **Select**
  Specifies the Select N condition to define the number of the category values that will be displayed.
  - **All**
    If selected, all category values in the chart will be displayed.
  - **Top N**
    If selected, specify a number in the field to the right and the first N category values in the chart will be displayed. You can also select a parameter from the drop-down list, which returns an integer, and then specify a value for the parameter when viewing the report to dynamically define the Top N condition.
  - **Bottom N**
    If selected, specify a number in the field to the right and the last N category values in the chart will be displayed. You can also select a parameter from the drop-down list which returns an integer, and then specify a value for the parameter when viewing the report to dynamically define the Bottom N condition.
- **Based On**
  If checked, the category values will be sorted by values of the summary selected with the direction specified. If unchecked, the category values will be sorted by the order specified in the Category Order box of the dialog.
- **Remaining Categories In**
  Enabled only when Top N or Bottom N is selected from the Select drop-down list. Check this option and then type a character string in the text field to group all the category values beyond the top/bottom N range.
- **Overall Series**
  Specifies whether to calculate the top or bottom N category values based on the series values. Enabled when Top N or Bottom N is selected from the Select drop-down list.
- **Skip First**
  If you check the Skip First option and input a number M in the text field to the right, then the first M category values in the chart will be skipped and the Select N condition will take effect beginning with M+1. The skipped values will be included in the Remaining Categories group together with all the category values beyond the top/bottom N range. Enabled when Top N or Bottom N is selected from the Select drop-down list.

OK

Accepts the changes and closes the dialog.

Cancel

Discards the changes and closes the dialog.

Help

Displays the help document about this feature.

Related topics:

- Setting Order/Select N condition for category/series/group field

Category Property dialog

This dialog appears when you:

- Do any of the following in the Business View Editor:
  - Click Insert > Category or the Insert Category button .
  - Right-click the business view or a category and select Add Category from the shortcut menu.
  - Right-click a category and select Edit Category from the shortcut menu.
- In the Catalog Browser resource tree, right-click a category and click Add Category or Edit Category on the shortcut menu.

It helps you to add or edit a category, and consists of the following tabs:

- General tab
- Security tab

OK

Applies all changes and closes the dialog.

Cancel

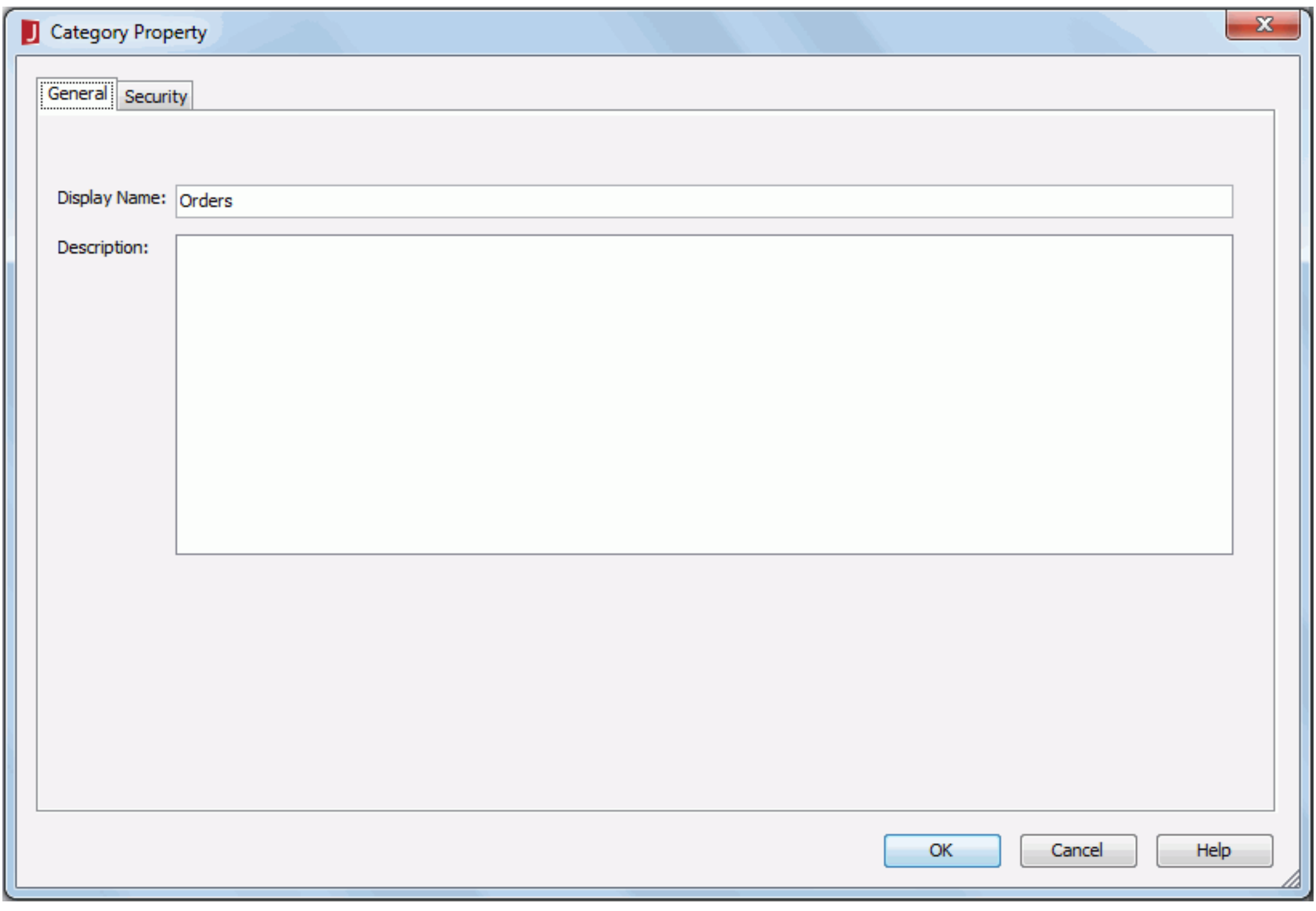
Does not retain any changes and closes the dialog.

Help

Displays the help document about this feature.

General tab

Specifies properties for the category.



Display Name

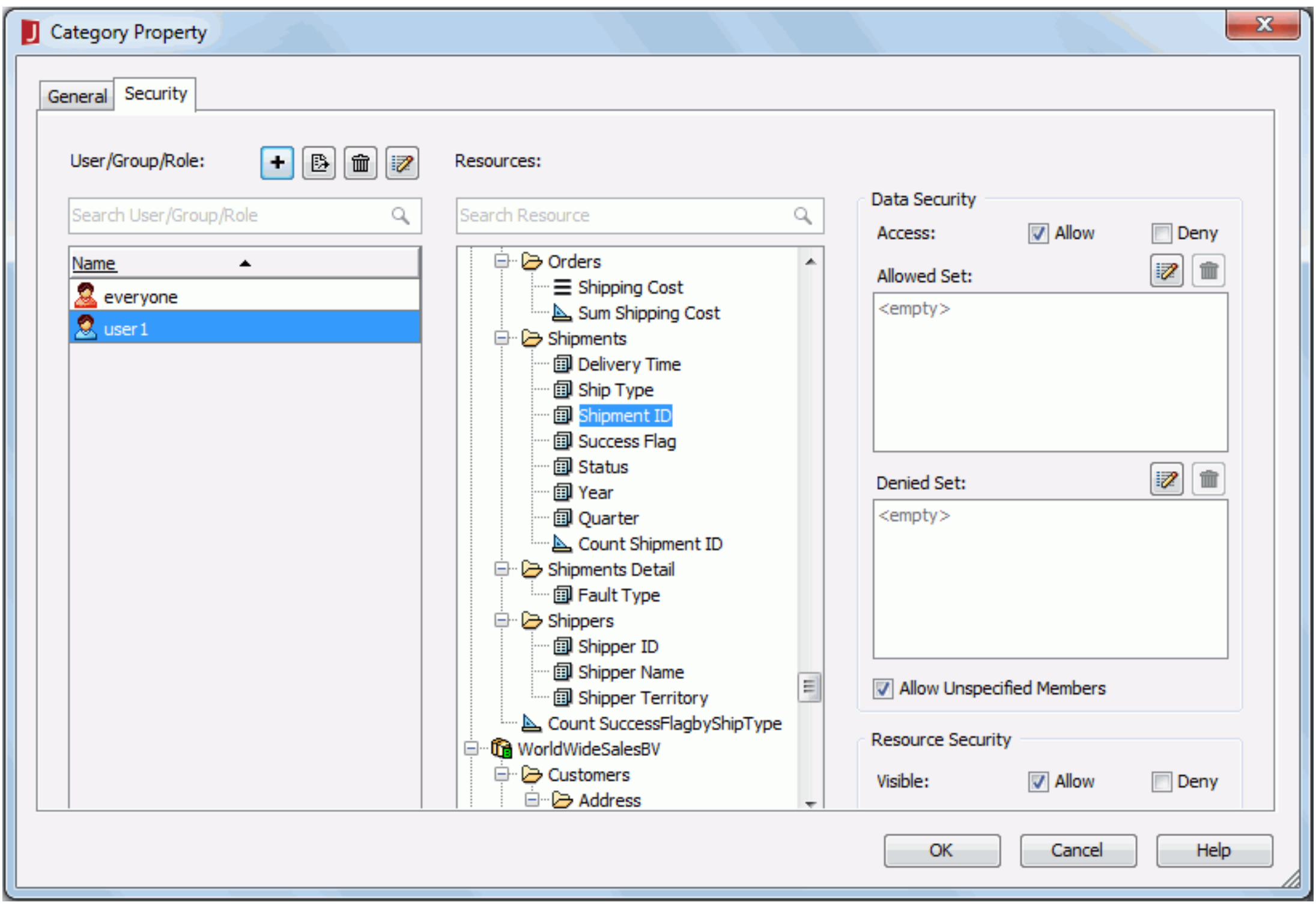
Specifies the display name of the category.

Description

Specifies the description of the category which will be shown when you hover the mouse pointer over the business view category in the resources panel of Page Report Studio or Web Report Studio.

Security tab

Specifies user accessibility to the members of the group objects in the data source.



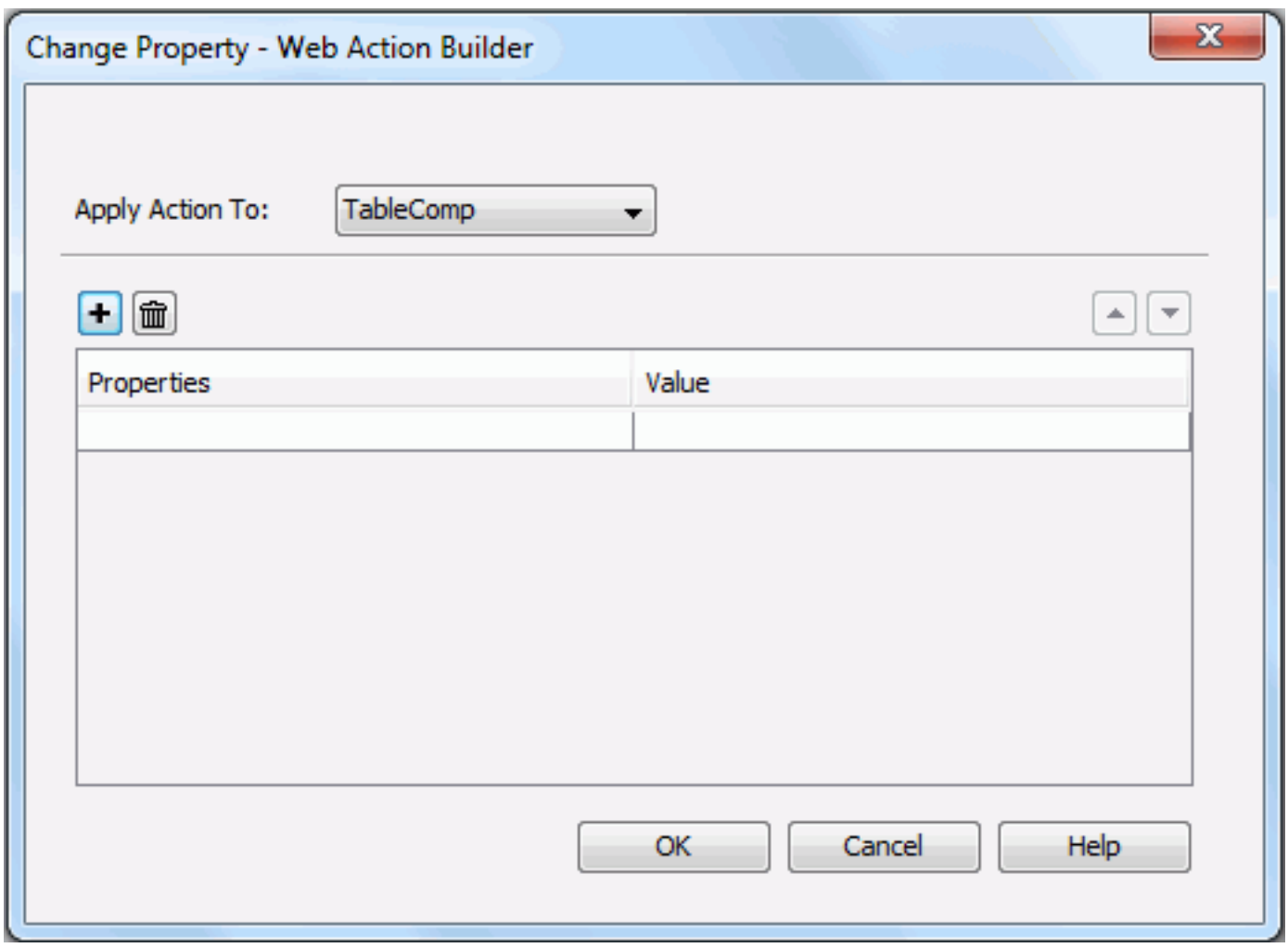
For details about options in the tab, refer to Edit Business View Security dialog.

Related topics:

- Editing a business view - Adding/Editing elements

Change Property - Web Action Builder dialog

This dialog appears when you select *Property and then click OK in the Web Action List dialog. It helps you to build a Change Property web action on an object so as to change the properties of a specified object when the event bound with the action is triggered at runtime.



The following are details about options in the dialog:

Apply Action To

Specifies to which object to apply the change property action. Select the desired object from the drop-down list.

Adds a new property line.

Removes the selected property line.

Moves the selected property line one step up.

Moves the selected property line one step down.

Properties

Specifies which property you want to change. Which properties are available here depends on the object you select from the Apply Action To drop-down list. Select the property you want to change from the drop-down list.

Value

Specifies the value of the property.

OK

Accepts the changes and closes the dialog.

Cancel

Does not retain any changes and closes the dialog.

Help

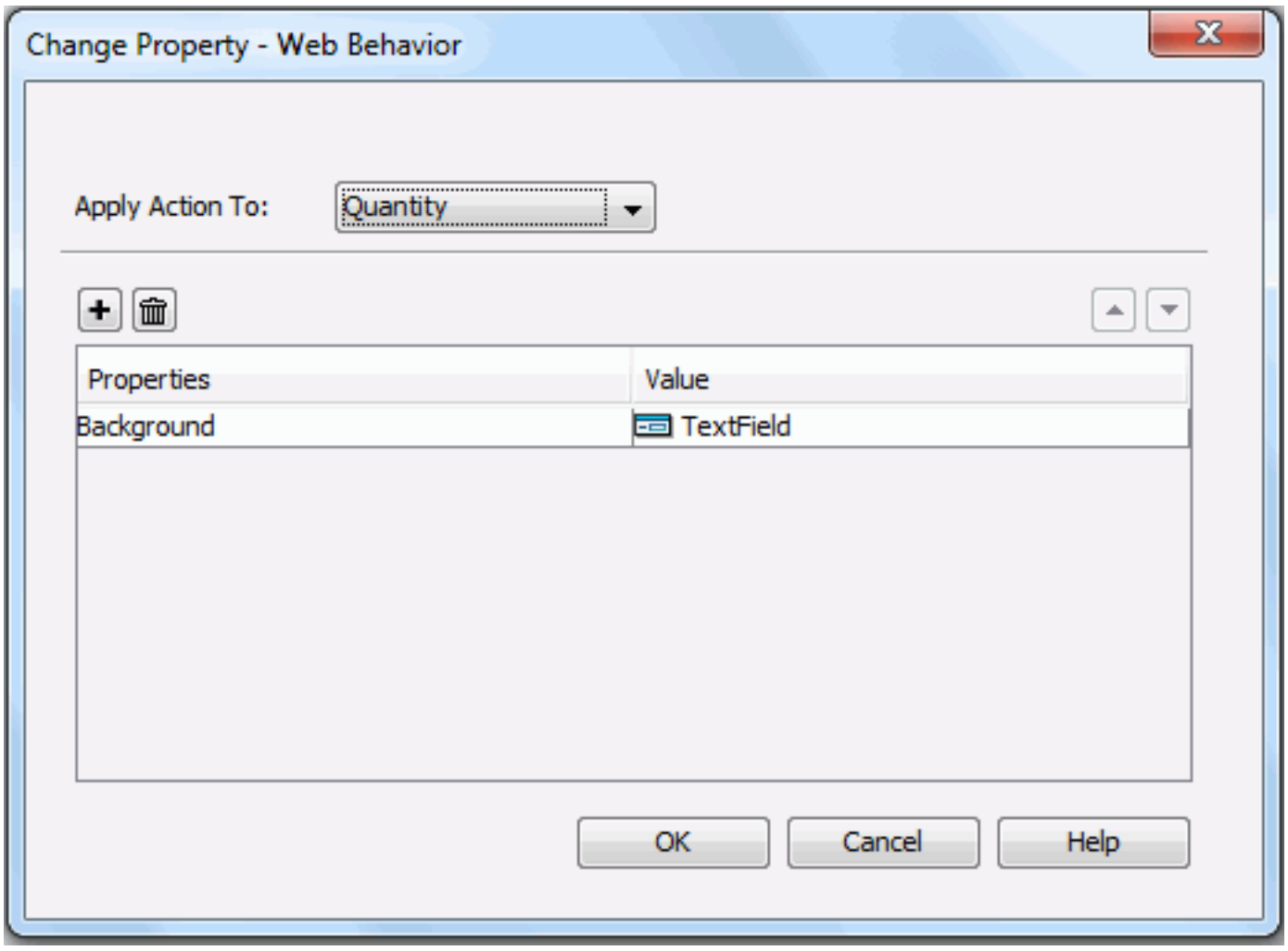
Displays the help document about this feature.

Related topics:

- Applying web actions to a label - Property

Change Property - Web Behavior dialog

This dialog appears when you right-click a web control in the configuration panel of a library component and select Web Behaviors > Change Property from the shortcut menu. It helps you to build a Change Property web behavior to the web control so as to change properties of a specified object in the library component at runtime.



The following are details about options in the dialog:

Apply Action To

Specifies to which object to apply the change property action. Select the desired object from the drop-down list.

Adds a new property line.

Removes the selected property line.

Moves the selected property line one step up.

Moves the selected property line one step down.

Properties

Specifies which property you want to change. Which properties are available here depends on the object you select from the Apply Action To drop-down list. Select the property you want to change from the drop-down list.

Value

Specifies the value of the property. The value may be input by yourself, or be specified by the value of a web control.

OK

Accepts the changes and closes the dialog.

Cancel

Does not retain any changes and closes the dialog.

Help

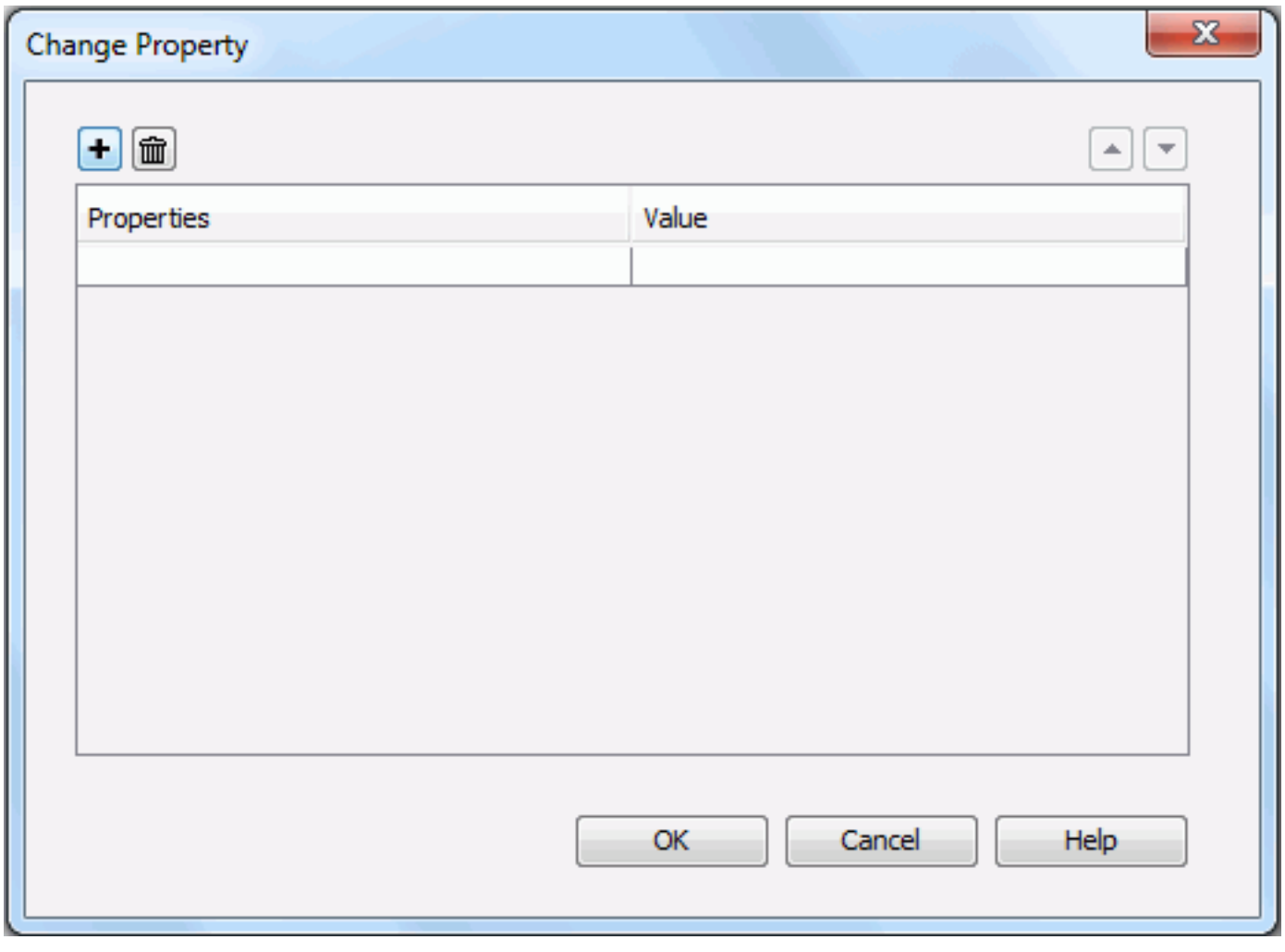
Displays the help document about this feature.

Related topics:

- Applying web behaviors to web controls in the panel

Change Property dialog

This dialog appears when you select *Property and then click OK in the Web Action List dialog. It helps you to define to change properties of the current data component as a response to the message the data component receives at runtime.



The following are details about options in the dialog:

Adds a new property line.

Removes the selected property line.

Moves the selected property line one step up.

Moves the selected property line one step down.

Properties

Specifies the property the value of which you want to change. Select the desired property from the drop-down list or input the property name by yourself. Only properties of the data component itself can be changed at runtime.

Value

Specifies the value of the property.

OK

Accepts the changes and closes the dialog.

Cancel

Does not retain any changes and closes the dialog.

Help

Displays the help document about this feature.

Related topics:

- Receiving messages

⇦⇨

Chart Wizard for page report

This wizard appears when you select the Chart layout in the New Page Report Tab dialog or New Page Report dialog and click OK, or right-click a chart in a page report and click Chart Wizard from the shortcut menu. It helps you create or modify a chart.

The wizard consists of the following screens:

- Data screen
- Type screen
- Display screen
- Filter screen
- Layout screen
- Style screen

Back

Goes back to the previous screen.

Next

Goes to the next screen.

Finish

Finishes creating or modifying the chart and closes this wizard.

Cancel

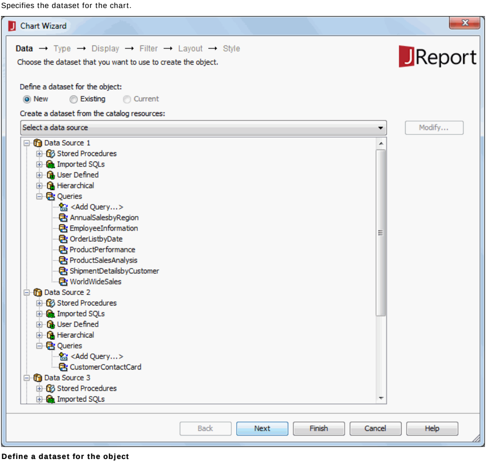
Does not retain changes and closes this wizard.

Help

Displays the help document about this feature.

Data screen

Specifies the dataset for the chart.



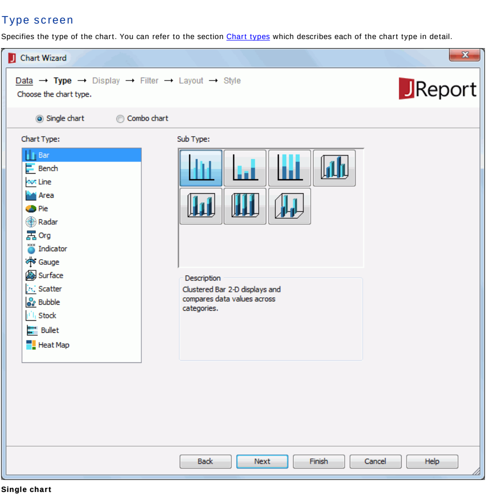
Define a dataset for the object

Specifies the dataset you want to use to create or modify the chart.

- **New**
  If checked, select a data source from the catalog resources to create a dataset that will be used to build the chart. When you choose to create the dataset from a query, you can click the Modify button to edit the query in the Query Editor if required.
- **Existing**
  If checked, select a dataset from the ones existing in the open report to create the chart. Click the Modify button to edit the dataset in the Dataset Editor if required.
- **Current**
  If checked, the current dataset used by the parent object will be applied to the chart.

Type screen

Specifies the type of the chart. You can refer to the section Chart types which describes each of the chart type in detail.



Single chart

Specifies to create a single chart.

- **Chart Type**
  Lists all the chart types that can be used to create a single chart.
- **Sub Type**
  Lists all the sub types of the selected chart type.
- **Description**
  Displays the description of the selected chart type.

Combo chart

Specifies to make more than one type use the primary or secondary axis. The types should be featured as Combination charts. Two types of data markers are used to represent different data values.

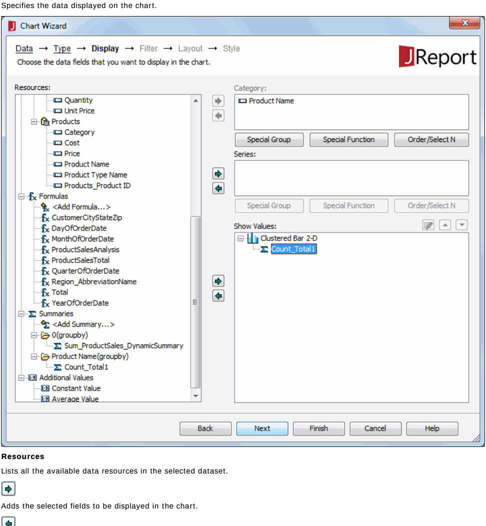
- **Chart Type**
  Lists all the chart types that can be added as a combo chart.
- **Sub-Type**
  Lists all the sub-types of the selected chart type.
- **Description**
  Displays the description of the selected chart type.
- **Chart Type Groups**
  Lists all the chart types that are to be used in the combo chart.
  - **Primary Axis**
    - **<Add Combo Type>**
      Adds a chart type to the primary axis.
  - **Secondary Axis**
    - **<Add Combo Type>**
      Adds a chart type to the secondary axis.
- Removes a specified chart type from the Chart Type Groups box.

Display screen

The Display screen differs with chart types: common chart types, org chart, or heat map.

Display screen for common chart types

Specifies the data displayed on the chart.



Resources

Lists all the available data resources in the selected dataset.

Adds the selected fields to be displayed in the chart.

Removes the selected fields from the chart if they are not required.

Replaces the selected field in the chart with the specified field in the Resources box.

Category/Series

Lists the field you want to display on the category/series axis of the chart.

For a real time chart, if no field is specified on the category axis, Use System Refresh Time will be automatically displayed in the Category box, namely, the time at which the chart refreshes itself will be used as the category value; the series axis will be disabled.

- **Special Group**
  Opens the User Defined Group dialog to define how to group information.
- **Special Function**
  Opens the Special Function dialog to define some special function to the field.
- **Order/Select N**
  Opens the Category Options dialog or Series Options dialog to specify the sort order of the category or series values and define the number of the category or series values that will be displayed in the chart.

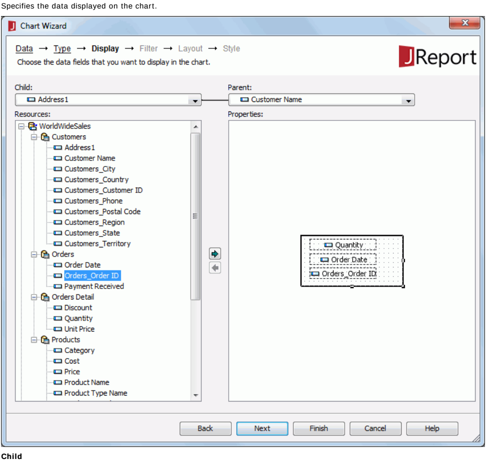
Show Values

Lists the values you want to show in the chart. For a real time chart, the values you add must be of Numeric type and cannot be summary fields.

- Opens the Edit Additional Value dialog to edit an additional value. Available only when a constant value or an average value is selected in the Show Values box.
- Moves the selected value one step up.
- Moves the selected value one step down.
- Adds a new pair of Y Axis and Radius for the bubble chart.
- Removes the selected field from the bubble chart if they are not required.
- **X Axis**
  Lists the value you want to show on the X axis of the bubble chart.
- **Y Axis**
  Lists the value you want to show on the Y axis of the bubble chart.
- **Radius**
  Lists the value you want to show as the bubble radius.

Display screen for org chart

Specifies the data displayed on the chart.



Child

Specify a field from the drop-down list which identifies the entity.

Parent

Specify a field from the drop-down list which shows the "reporting to" relationship among the entity members, that is, which child field member reports to or belongs to which child field member.

Resources

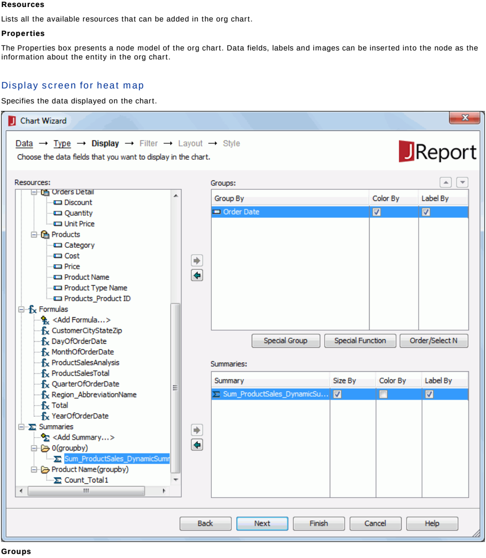
Lists all the available resources that can be added in the org chart.

Properties

The Properties box presents a node model of the org chart. Data fields, labels and images can be inserted into the node as the information about the entity in the org chart.

Display screen for heat map

Specifies the data displayed on the chart.



Groups

Lists the fields used to group the data. There should be at least one group.

- **Color By**
  Specifies whether to color by a group. 0-n groups can be used as the color-by fields.
- **Label By**
  Specifies whether to show a group field in the innermost rectangle. 0-n groups can be used as the label-by fields.
- **Special Group**
  Opens the User Defined Group dialog to define how to group information.
- **Special Function**
  Opens the Special Function dialog to define some special function to the field.
- **Order/Select N**
  Opens the Group Options dialog to specify the sort order of the group values and define the number of the group values that will be displayed in the chart.

Summaries

Lists the summaries used as size-by/color-by or displayed in the innermost rectangle.

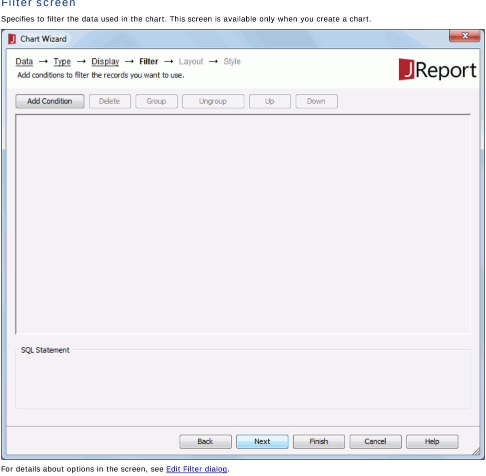
The summaries should match the groups. For example, if the groups level is A > B > C, the static summaries grouped by C can be inserted into the Summaries box, but the static summaries grouped by A, B or other fields cannot.

If no group is specified in the Groups box, you can insert any static summary. And its group-by field will be inserted into the Groups box automatically.

- **Size By**
  Specifies to size by one summary or none.
- **Color By**
  Specifies to color by one summary or none.
- **Label By**
  Specifies whether to show a summary in the innermost rectangle. 0-n summaries can be used as the label-by fields.

Filter screen

Specifies to filter the data used in the chart. This screen is available only when you create a chart.



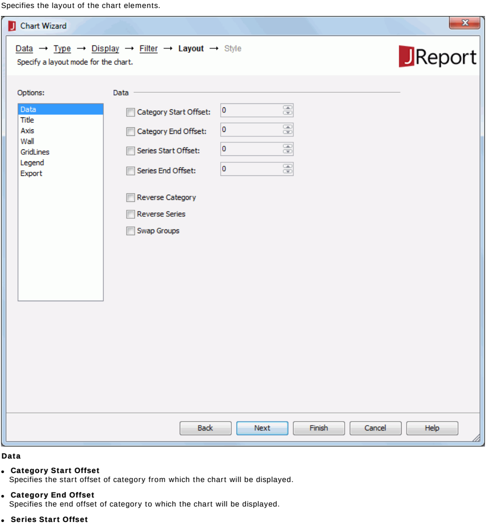
For details about options in the screen, see Edit Filter dialog.

Layout screen

The Layout screen differs with chart type: common chart types, org chart, or heat map.

Layout screen for common chart types

Specifies the layout of the chart elements.



Data

- **Category Start Offset**
  Specifies the start offset of category from which the chart will be displayed.
- **Category End Offset**
  Specifies the end offset of category to which the chart will be displayed.
- **Series Start Offset**
  Specifies the start offset of series from which the chart will be displayed.
- **Series End Offset**
  Specifies the end offset of series to which the chart will be displayed.
- **Reverse Category**
  Specifies whether to reverse the display order of the categories.
- **Reverse Series**
  Specifies whether to reverse the display order of the series.
- **Swap Groups**
  Specifies to display values from different data fields by switching data between the category and series axes, the category and values axes.

Notes:

- If values are set to be less than 0, it means the categories or series will be displayed without any offset.
- If values are set to be greater than 0 and the value of Category/Series Start Offset is greater than that of Category/Series End Offset, all of the categories or series will be displayed.

Title

- **Chart Title**
  Specifies the title of the chart.
- **Category (X) Axis Title**
  Specifies the title of the X-axis.
- **Value (Y) Axis Title**
  Specifies the title of the Y-axis.
- **Value (Y2) Axis Title**
  Specifies the title of the Y2-axis.

Axis

- **Show Category (X) Axis**
  Specifies whether to show the category (X) axis in this chart.
- **Show Value (Y) Axis**
  Specifies whether to show the value (Y) axis in this chart.
- **Show Value (Y2) Axis**
  Specifies whether to show the value (Y2) axis in this chart.
- **Show Series (Z) Axis**
  Specifies whether to show the series (Z) axis in this chart.

Wall

- **Show Wall**
  Specifies whether to show the wall in this chart.
- **Show Floor**
  Specifies whether to show the floor in this chart.

Gridlines

- **Show Category (X) Axis Gridlines**
  Specifies whether to show the gridlines of category (X) axis.
- **Show Value (Y) Axis Gridlines**
  Specifies whether to show the gridlines of value (Y) axis.
- **Show Value (Y2) Axis Gridlines**
  Specifies whether to show the gridlines of value (Y2) axis.
- **Show Series (Z) Axis Gridlines**
  Specifies whether to show the gridlines of series (Z) axis.

Legend

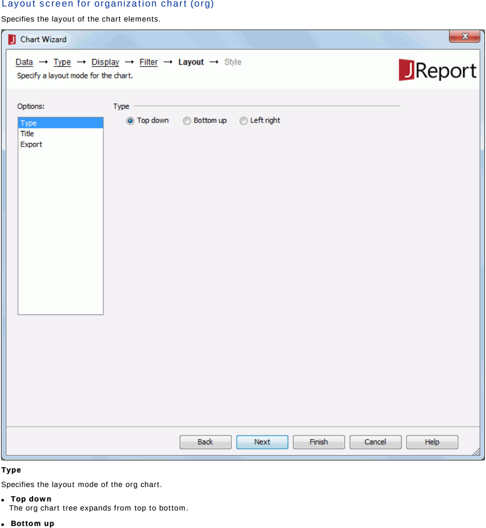
- **Show Legend**
  Specifies whether to show the legend in this chart.

Export

- **Mapping Component**
  Specifies the component to which the chart will be mapped. For details, see Creating dynamic chart in Excel.

Layout screen for organization chart (org)

Specifies the layout of the chart elements.



Type

Specifies the layout mode of the org chart.

- **Top down**
  The org chart tree expands from top to bottom.
- **Bottom up**
  The org chart tree expands from bottom to top. It is the reverse of the top down type.
- **Left right**
  The org chart tree expands from left to right.

Title

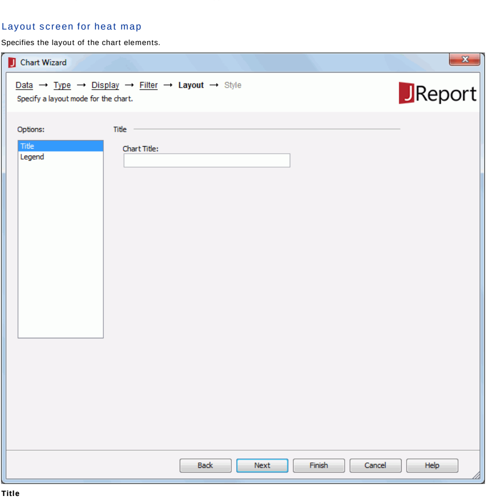
- **Chart Title**
  Specifies the title of the chart.

Export

- **Mapping Component**
  Specifies the component to which the chart will be mapped. For details, see Creating dynamic chart in Excel.

Layout screen for heat map

Specifies the layout of the chart elements.



Title

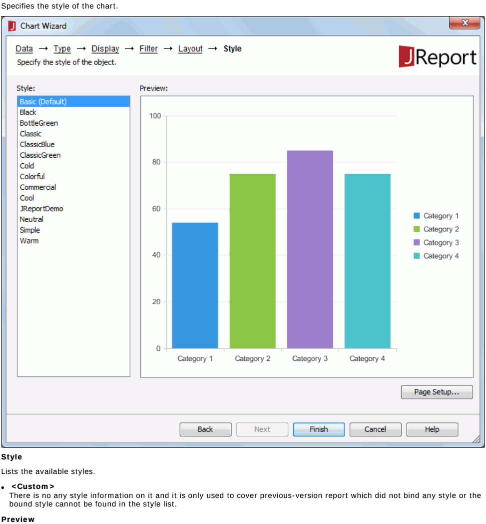
- **Chart Title**
  Specifies the title of the chart.

Legend

- **Show Legend**
  Specifies whether to show the legend in this chart.

Style screen

Specifies the style of the chart.



Style

Lists the available styles.

- **<Custom>**
  There is no any style information on it and it is only used to cover previous-version report which did not bind any style or the bound style cannot be found in the style list.

Preview

Displays a diagram illustrating the effect of the selected style on the chart.

Inherit Style

Specifies whether to make the chart take the style of its parent. This options is available only when you modify a chart and the chart is inserted into a banded object or table in a page report.

Page Setup

Opens the Page Setup dialog to specify page properties. Available only when creating a page report.

Related topics:

- Creating a chart report
- Modifying a chart

⇦⇨

Chart Wizard for web report/library component

This wizard appears when you select Chart in the component type box and click OK in the New Library Component dialog, or right-click a chart in a web report or library component and click Chart Wizard from the shortcut menu. It helps you create or modify a chart.

The wizard consists of the following screens:

- Data screen
- Type screen
- Display screen
- Filter screen
- Layout screen
- Style screen

Back

Goes back to the previous screen.

Next

Goes to the next screen.

Finish

Finishes creating or modifying the chart and closes this wizard.

Cancel

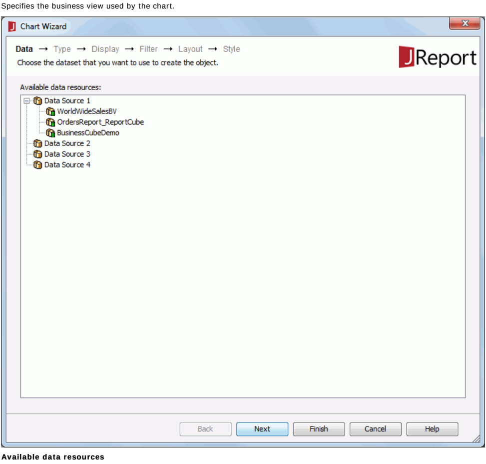
Does not retain changes and closes this wizard.

Help

Displays the help document about this feature.

Data screen

Specifies the business view used by the chart.

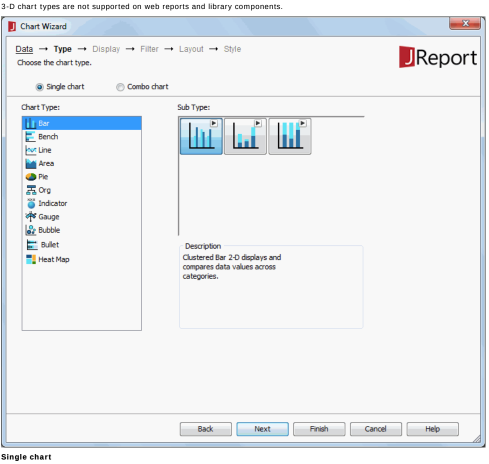


Available data resources

Lists all the business views in the current catalog. Specify the one you want to use.

Type screen

Specifies the type of the chart. You can refer to the section Chart types which describes each of the chart type in detail. All the 3-D chart types are not supported on web reports and library components.



Single chart

Specifies to create a single chart.

- **Chart Type**
  Lists all the chart types that can be used to create a single chart.
- **Sub Type**
  Lists all the sub types of the selected chart type.
- **Description**
  Displays the description of the selected chart type.

Combo chart

Specifies to make more than one type use the primary or secondary axis. The types should be featured as Combination charts. Two types of data markers are used to represent different data values.

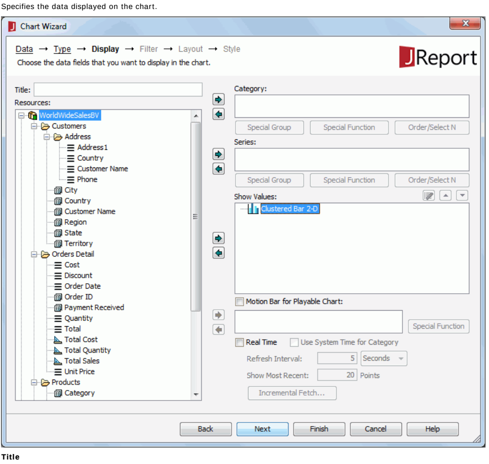
- **Chart Type**
  Lists all the chart types that can be added as a combo chart.
- **Sub-Type**
  Lists all the sub-types of the selected chart type.
- **Description**
  Displays the description of the selected chart type.
- **Chart Type Groups**
  Lists all the chart types that are to be used in the combo chart.
  - **Primary Axis**
    - **<Add Combo Type>**
      Adds a chart type to the primary axis.
  - **Secondary Axis**
    - **<Add Combo Type>**
      Adds a chart type to the secondary axis.
- Removes a specified chart type from the Chart Type Groups box.

Display screen

The Display screen differs with chart types: common chart types, org chart, or heat map.

Display screen for common chart types

Specifies the data displayed on the chart.



Title

Specifies the title of the chart.

Resources

Lists all the available data resources in the selected dataset. You can also create dynamic formulas and aggregations to use in the chart.

Adds the selected fields to be displayed in the chart.

Removes the selected fields from the chart if they are not required.

Category/Series

Lists the field you want to display on the category/series axis of the chart.

For a real time chart, if no field is specified on the category axis, Use System Refresh Time will be automatically displayed in the Category box, namely, the time at which the chart refreshes itself will be used as the category value; the series axis will be disabled.

- **Special Group**
  Opens the User Defined Group dialog to define how to group information.
- **Special Function**
  Opens the Special Function dialog to define some special function to the field.
- **Order/Select N**
  Opens the Category Options dialog or Series Options dialog to specify the sort order of the category or series values and define the number of the category or series values that will be displayed in the chart.

Show Values

Lists the values you want to show in the chart. For a real time chart, the values you add must be of Numeric type and cannot be summary fields.

- Opens the Edit Additional Value dialog to edit an additional value. Available only when a constant value or an average value is selected in the Show Values box.
- Moves the selected value one step up.
- Moves the selected value one step down.
- Adds a new pair of Y Axis and Radius for the bubble chart.
- Removes the selected field from the bubble chart if they are not required.
- **X Axis**
  Lists the value you want to show on the X axis of the bubble chart.
- **Y Axis**
  Lists the value you want to show on the Y axis of the bubble chart.
- **Radius**
  Lists the value you want to show as the bubble radius.

Motion Bar for Playable Chart

Lists the field you want to use as the motion field. A motion field can only be of Integer, Date or Time data type. Available to single bar, bench and bubble charts in web reports and library components only.

- **Special Function**
  Opens the Special Function dialog to define some special function to the motion field. Available only when the motion field is of Date data type.

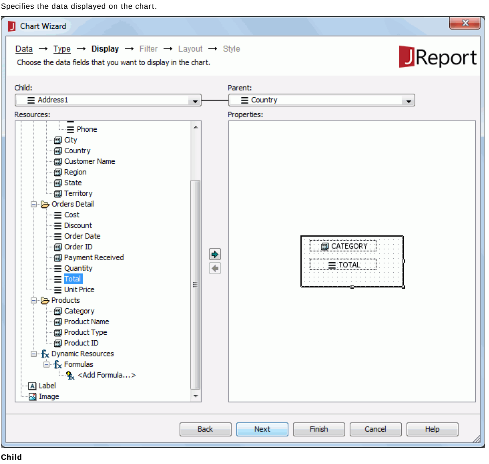
Real Time

Specifies to run the chart in real time mode, which means it will be updated automatically by using real time data after it is published to JReport Server. Available to single bar, bench, line, and area charts in web reports and library components only.

- **Use System Time for Category**
  Specifies to use the time at which the chart refreshes itself as the category value.
- **Refresh Interval**
  Specifies the time interval at which the chart will get data and refresh itself automatically.
- **Show Most Recent N Points**
  Specifies the number of points that will be kept for the real time data on the chart.
- **Incremental Fetch**
  Opens the Unique Key dialog to configure a unique key for the real time chart.

Display screen for org chart

Specifies the data displayed on the chart.



Child

Specify a field from the drop-down list which identifies the entity.

Parent

Specify a field from the drop-down list which shows the "reporting to" relationship among the entity members, that is, which child field member reports to or belongs to which child field member.

Resources

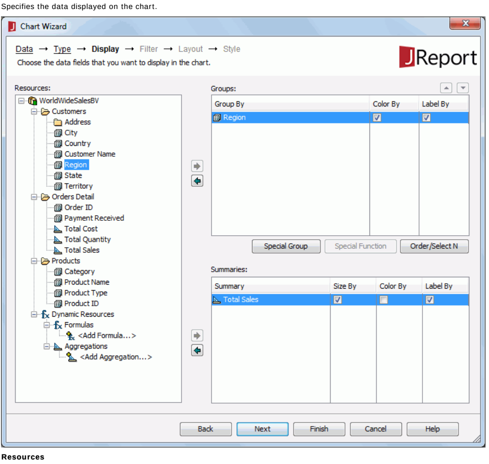
Lists all the available resources that can be added to the org chart. You can also create dynamic formulas and aggregations to use in the chart.

Properties

The Properties box presents a node model of the org chart. Data fields, labels and images can be inserted into the node as the information about the entity in the org chart.

Display screen for heat map

Specifies the data displayed on the chart.



Resources

Lists all the available data resources in the selected dataset. You can also create dynamic formulas and aggregations to use in the chart.

Groups

Lists the fields used to group the data. There should be at least one group.

- **Color By**
  Specifies whether to color by a group. 0-n groups can be used as the color-by fields.
- **Label By**
  Specifies whether to show a group field in the innermost rectangle. 0-n groups can be used as the label-by fields.
- **Special Group**
  Opens the User Defined Group dialog to define how to group information.
- **Special Function**
  Opens the Special Function dialog to define some special function to the field.
- **Order/Select N**
  Opens the Group Options dialog to specify the sort order of the group values and define the number of the group values that will be displayed in the chart.

Summaries

Lists the summaries used as size-by/color-by or displayed in the innermost rectangle.

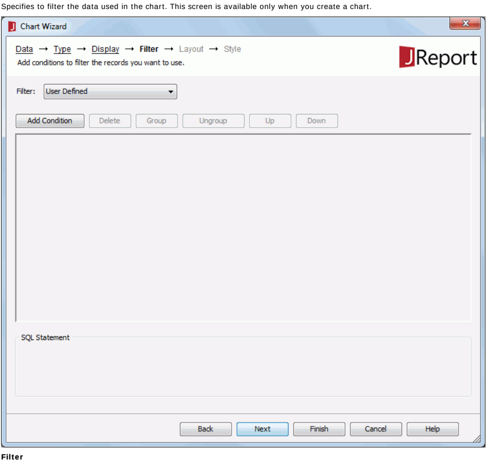
The summaries should match the groups. For example, if the groups level is A > B > C, the static summaries grouped by C can be inserted into the Summaries box, but the static summaries grouped by A, B or other fields cannot.

If no group is specified in the Groups box, you can insert any static summary. And its group-by field will be inserted into the Groups box automatically.

- **Size By**
  Specifies to size by one summary or none.
- **Color By**
  Specifies to color by one summary or none.
- **Label By**
  Specifies whether to show a summary in the innermost rectangle. 0-n summaries can be used as the label-by fields.

Filter screen

Specifies to filter the data used in the chart. This screen is available only when you create a chart.



Filter

It lists all the predefined filters of the specified business view. Select one from the drop-down list to apply, or select User Defined and define a new filter according to your requirements.

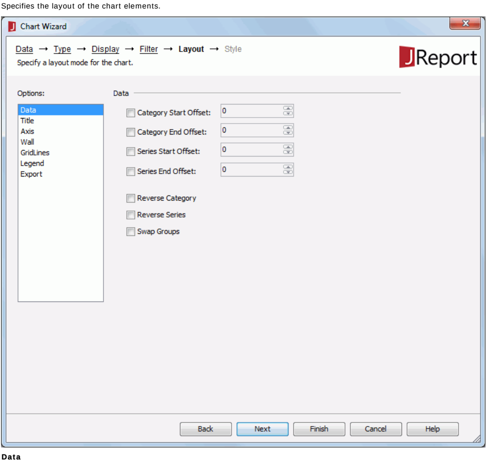
For details about options in the screen, see Edit Filter dialog.

Layout screen

The Layout screen differs with chart type: common chart types, org chart, or heat map.

Layout screen for common chart types

Specifies the layout of the chart elements.



Data

- **Category Start Offset**
  Specifies the start offset of category from which the chart will be displayed.
- **Category End Offset**
  Specifies the end offset of category to which the chart will be displayed.
- **Series Start Offset**
  Specifies the start offset of series from which the chart will be displayed.
- **Series End Offset**
  Specifies the end offset of series to which the chart will be displayed.
- **Reverse Category**
  Specifies whether to reverse the display order of the categories.
- **Reverse Series**
  Specifies whether to reverse the display order of the series.
- **Swap Groups**
  Specifies to display values from different data fields by switching data between the category and series axes, the category and values axes.

Notes:

- If values are set to be less than 0, it means the categories or series will be displayed without any offset.
- If values are set to be greater than 0 and the value of Category/Series Start Offset is greater than that of Category/Series End Offset, all of the categories or series will be displayed.

Title

- **Chart Title**
  Specifies the title of the chart.
- **Category (X) Axis Title**
  Specifies the title of the X-axis.
- **Value (Y) Axis Title**
  Specifies the title of the Y-axis.
- **Value (Y2) Axis Title**
  Specifies the title of the Y2-axis.

Axis

- **Show Category (X) Axis**
  Specifies whether to show the category (X) axis in this chart.
- **Show Value (Y) Axis**
  Specifies whether to show the value (Y) axis in this chart.
- **Show Value (Y2) Axis**
  Specifies whether to show the value (Y2) axis in this chart.
- **Show Series (Z) Axis**
  Specifies whether to show the series (Z) axis in this chart.

Wall

- **Show Wall**
  Specifies whether to show the wall in this chart.
- **Show Floor**
  Specifies whether to show the floor in this chart.

Gridlines

- **Show Category (X) Axis Gridlines**
  Specifies whether to show the gridlines of category (X) axis.
- **Show Value (Y) Axis Gridlines**
  Specifies whether to show the gridlines of value (Y) axis.
- **Show Value (Y2) Axis Gridlines**
  Specifies whether to show the gridlines of value (Y2) axis.
- **Show Series (Z) Axis Gridlines**
  Specifies whether to show the gridlines of series (Z) axis.

Legend

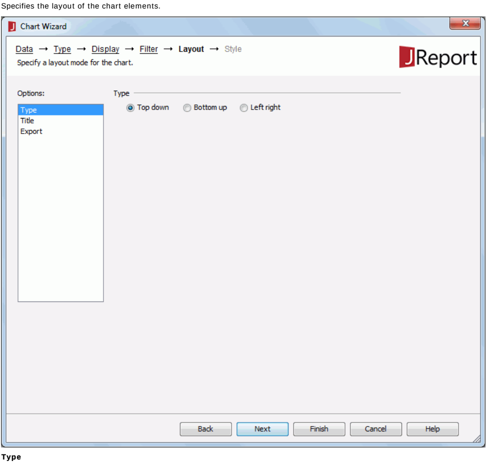
- **Show Legend**
  Specifies whether to show the legend in this chart.

Export

- **Mapping Component**
  Specifies the component to which the chart will be mapped. For details, see Creating dynamic chart in Excel.

Layout screen for organization chart (org)

Specifies the layout of the chart elements.



Type

Specifies the layout mode of the org chart.

- **Top down**
  The org chart tree expands from top to bottom.
- **Bottom up**
  The org chart tree expands from bottom to top. It is the reverse of the top down type.
- **Left right**
  The org chart tree expands from left to right.

Title

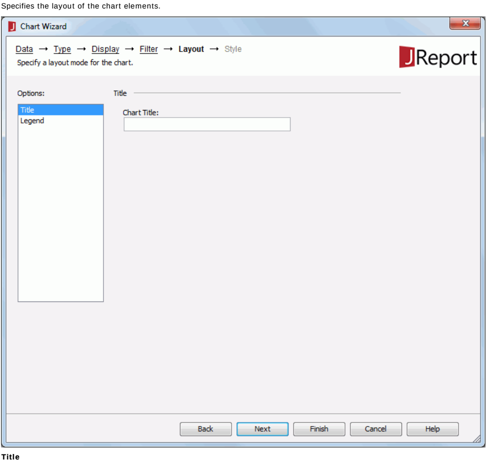
- **Chart Title**
  Specifies the title of the chart.

Export

- **Mapping Component**
  Specifies the component to which the chart will be mapped. For details, see Creating dynamic chart in Excel.

Layout screen for heat map

Specifies the layout of the chart elements.



Title

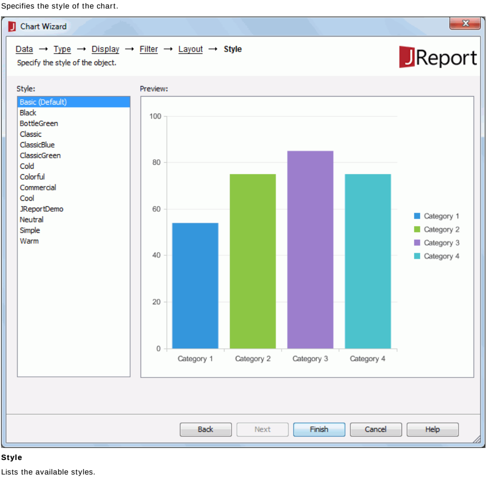
- **Chart Title**
  Specifies the title of the chart.

Legend

- **Show Legend**
  Specifies whether to show the legend in this chart.

Style screen

Specifies the style of the chart.



Style

Lists the available styles.

- **<Custom>**
  There is no any style information on it and it is only used to cover previous-version report which did not bind any style or the bound style cannot be found in the style list.

Preview

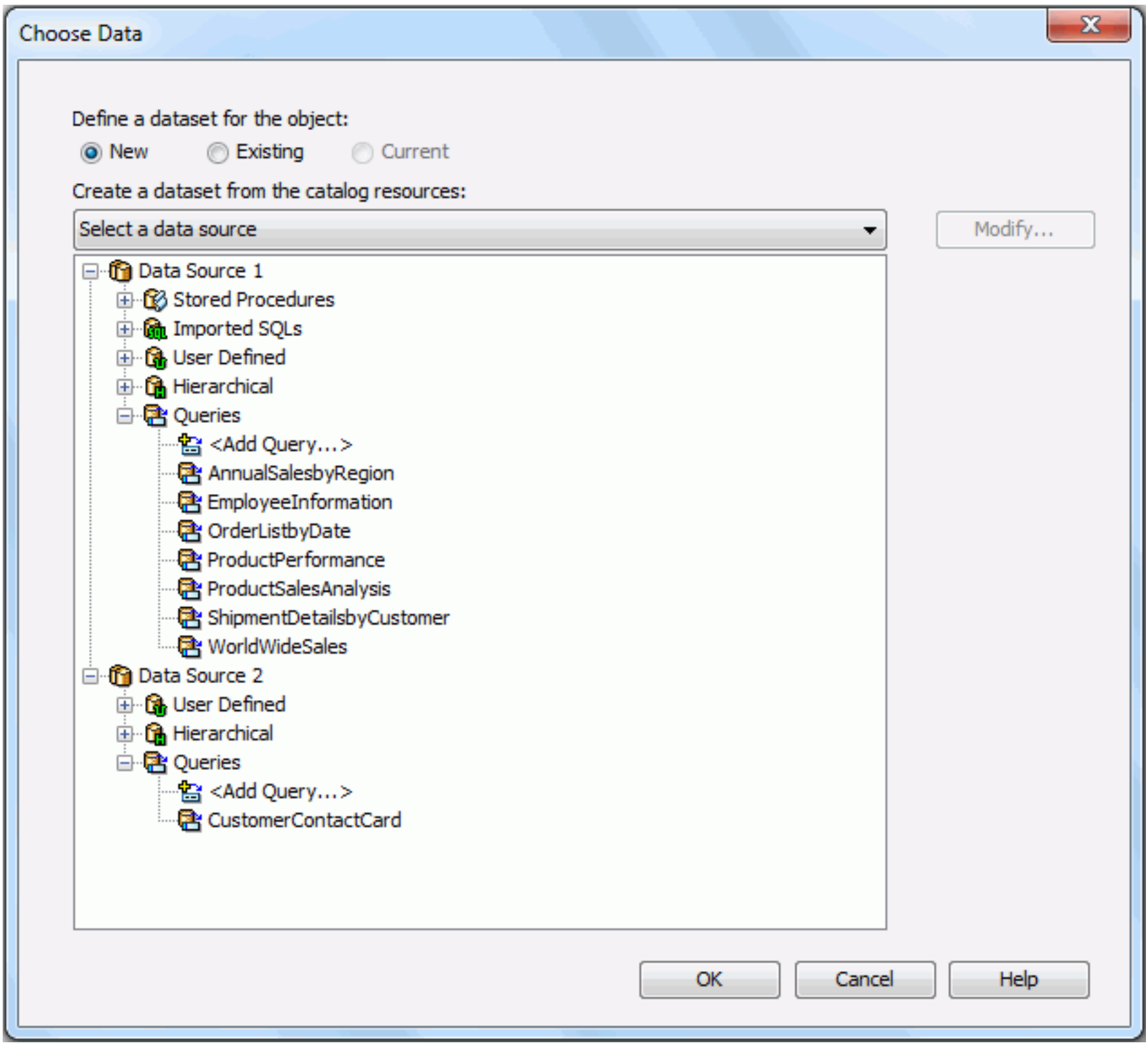
Displays a diagram illustrating the effect of the selected style on the chart.

Related topics:

- Creating a library component
- Modifying a chart

Choose Data dialog

This dialog appears when you select the Blank layout and click OK in the New Page Report Tab dialog or New Page Report dialog. It helps you to choose the dataset you want to use to create the report.



The following are details about options in the dialog:

Define a dataset for this object

Specifies the dataset from which the report will be created.

- **New**
  If checked, select a data source from the catalog resources to create a dataset that will be used to build the report. When you choose to create the dataset from a query, you can click the Modify button to edit the query if required.
- **Existing**
  If checked, select a dataset from the ones existing in the open report to create the report. Click the Modify button to edit the query if required.

OK

Accepts changes and closes this dialog.

Cancel

Does not retain changes and closes this dialog.

Help

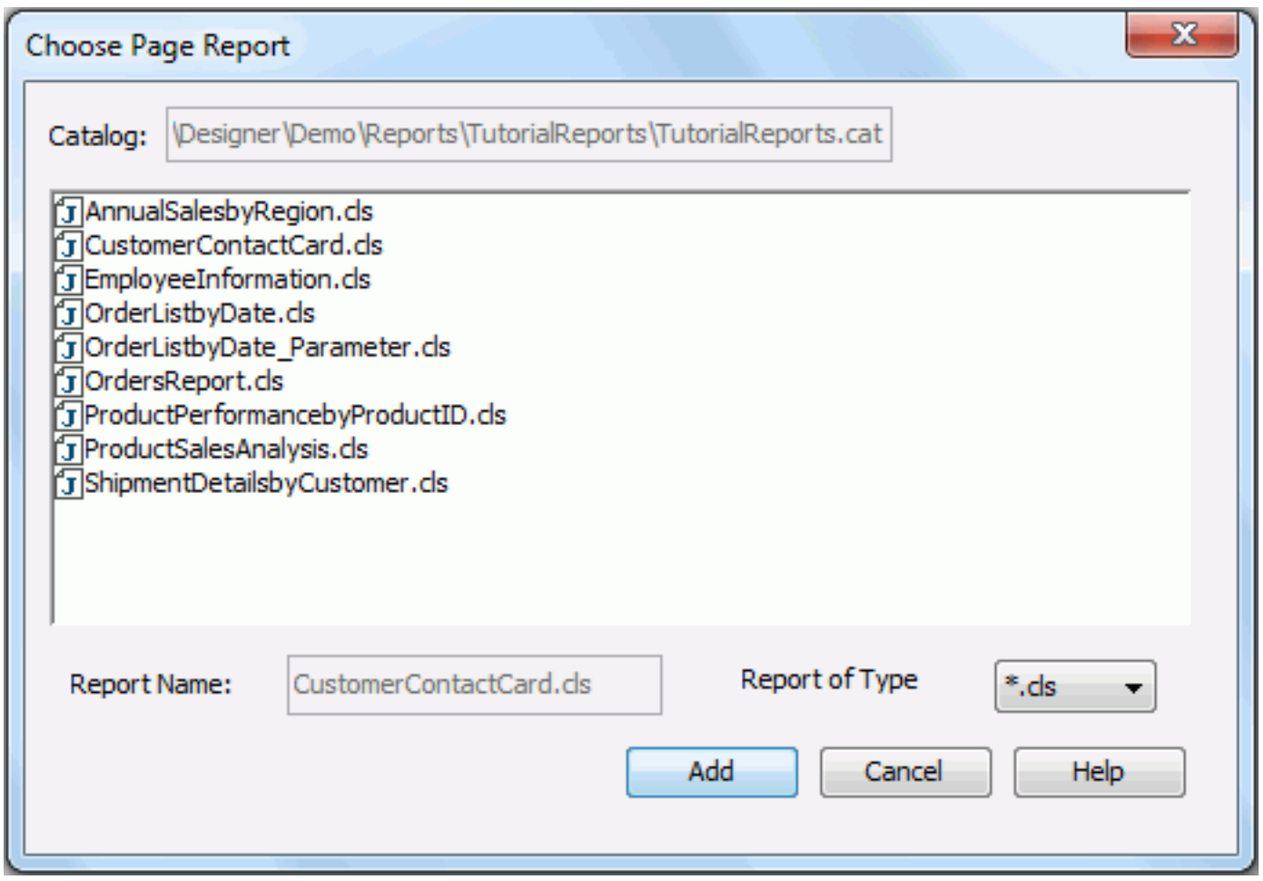
Displays the help document about this feature.

Relaled topics:

- Creating a blank report

Choose Page Report dialog

This dialog appears when you right-click a folder node in the Reports tab of the Catalog Browser and select Add Page Report. It helps you to add page reports saved in the current catalog into the Page Report Manager.



The following are details about options in the dialog:

Catalog

Displays the path and name of the current catalog.

Report Name

Displays the name of the selected report.

Report of Type

Specifies the type of the report. When saving a report, you can save it as four kinds of files: cls (report binary file), rpt (report source file), cls.xml (XML format report), or clx (self contained report).

Add

Adds the selected report to the folder.

- To add consecutive reports, click the first report, press and hold down **SHIFT**, and then click the last report, then click the **Add** button.
- To add reports that are not consecutive, press and hold down **CTRL**, click each report, and then click the **Add** button.

Cancel

Closes this dialog and leaves any changes unsaved.

Help

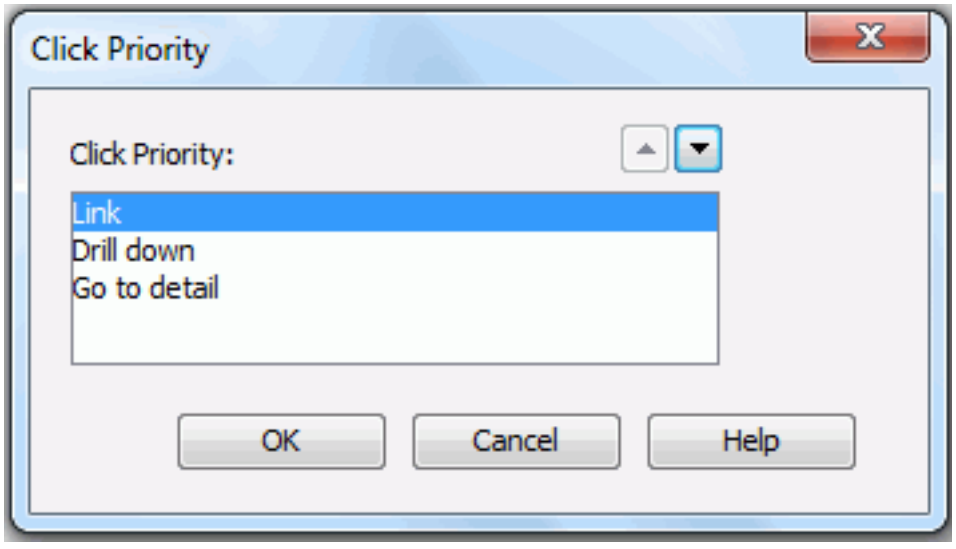
Displays the help document about this feature.

Related topics:

- Adding an item to the Page Report Manager

Click Priority dialog for page report

This dialog appears when you click the button ... in the value cell of the report level property Click Priority in the Others category in the Report Inspector. It is used to set the priority of the actions that will be triggered at runtime when you click on certain objects in a page report which are bound with some actions.



The following are details about options in the dialog:

Click Priority

Specifies the click priority of the actions. Move the one you want to have the highest priority to the top of the list.

- Moves the selected action one step up.
- Moves the selected action one step down.

Note: When an object is bound with all of the actions in the list, when "Link" or "Go to detail" is specified the highest priority, the other actions on the object can only be performed by using shortcut menu command. If "Drill down" is specified the highest click priority, the link action will be triggered when the lowest level of the drilling hierarchy is reached.

OK

Applies the changes and closes the dialog.

Cancel

Does not retain any changes and closes the dialog.

Help

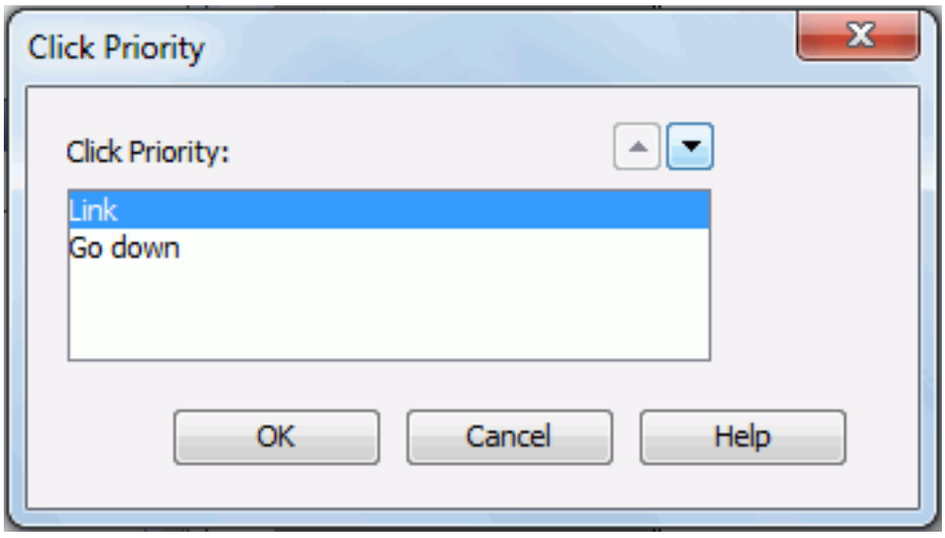
Displays the help document about this feature.

Click Priority dialog for web report/library component

This dialog appears in the following cases:

- For a web report, click the button ... in the value cell of the report level property Click Priority in the Others category in the Report Inspector.
- For a library component, click the button ... in the value cell of the property Click Priority on the Contents node in the Report Inspector.

It is used to set the priority of the actions that will be triggered at runtime when you click on certain objects in a web report or library component which are bound with some actions.



The following are details about options in the dialog:

Click Priority

Specifies the click priority of the actions. Move the one you want to have the highest priority to the top of the list.

- Moves the selected action one step up.
- Moves the selected action one step down.

Note: When an object is bound with both actions in the list, when "Link" is specified the highest priority, the other action on the object can only be performed by using the shortcut menu command. If "Go down" is specified the highest click priority, the link action will be triggered when the lowest level of the going hierarchy is reached.

OK

Applies the changes and closes the dialog.

Cancel

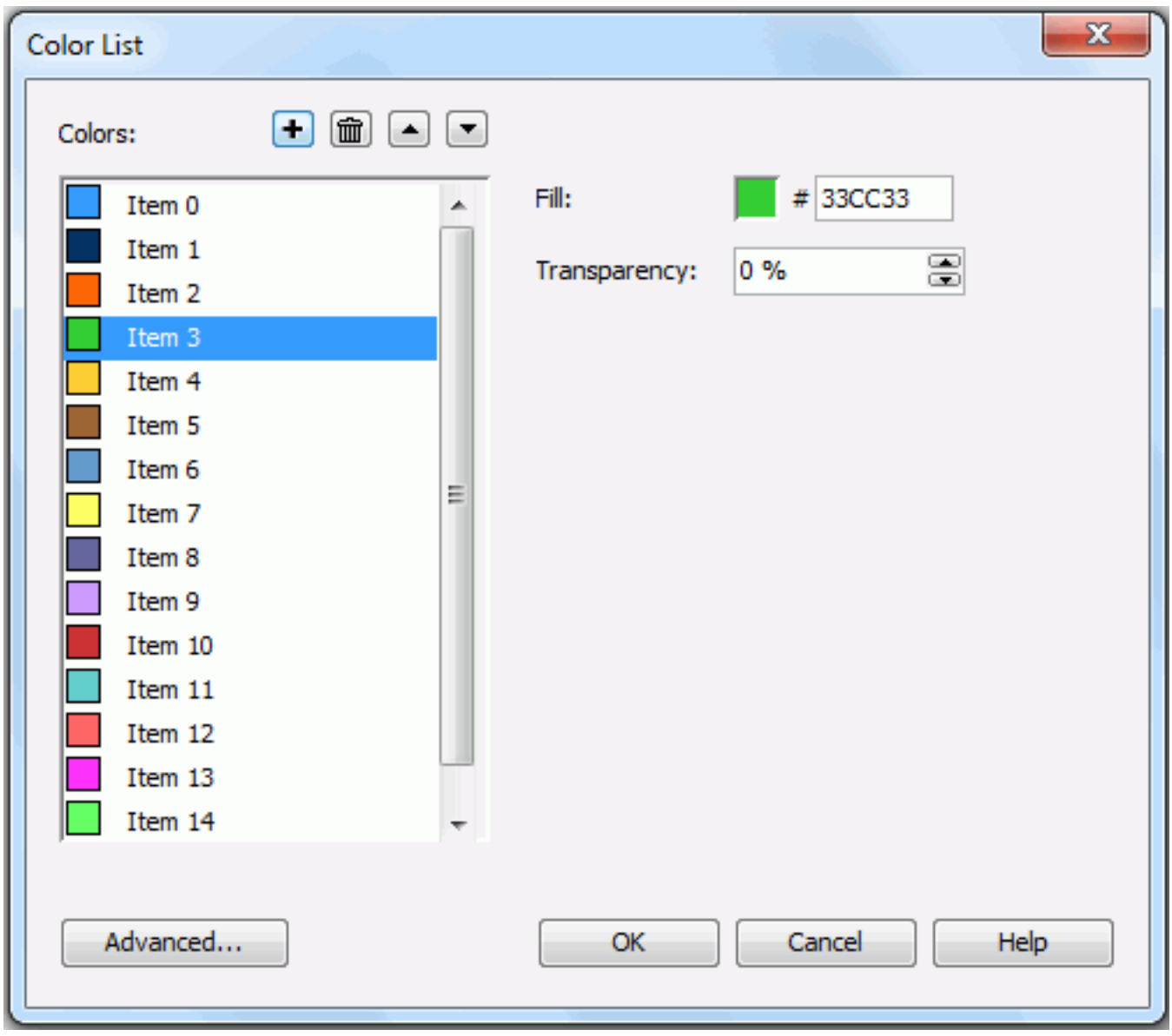
Does not retain any changes and closes the dialog.

Help

Displays the help document about this feature.

Color List dialog

This dialog appears when you click the Color List or Pattern List button in some chart element formatting dialogs. It is used to modify the fill effect of data markers in the same data series respectively.



The following are details about options in the dialog:

Colors

Specifies the color and the transparency for the data markers. Click the items in the box one by one to pick the pattern for the data markers.

- Adds a new pattern item.
- Removes the selected pattern item.
- Moves the selected pattern item one step up.
- Moves the selected pattern item one step down.
- **Fill**
  Specifies the color schema for the pattern.

  To specify the color, click the color image button and select a color swatch or type the hexadecimal value of the desired color (for example, 0xff0000) in the text box directly. If the color swatches cannot meet your requirement, you can click More Colors on the color palette to open the Pick a Color dialog, with which you can set a full range of color, or click More Fill Effects to specify a fill effect in the Fill Effects dialog.
- **Transparency**
  Specifies the transparency.

Advanced

Opens the Advanced dialog to specify the fill effect for certain values.

OK

Applies the changes and closes the dialog.

Cancel

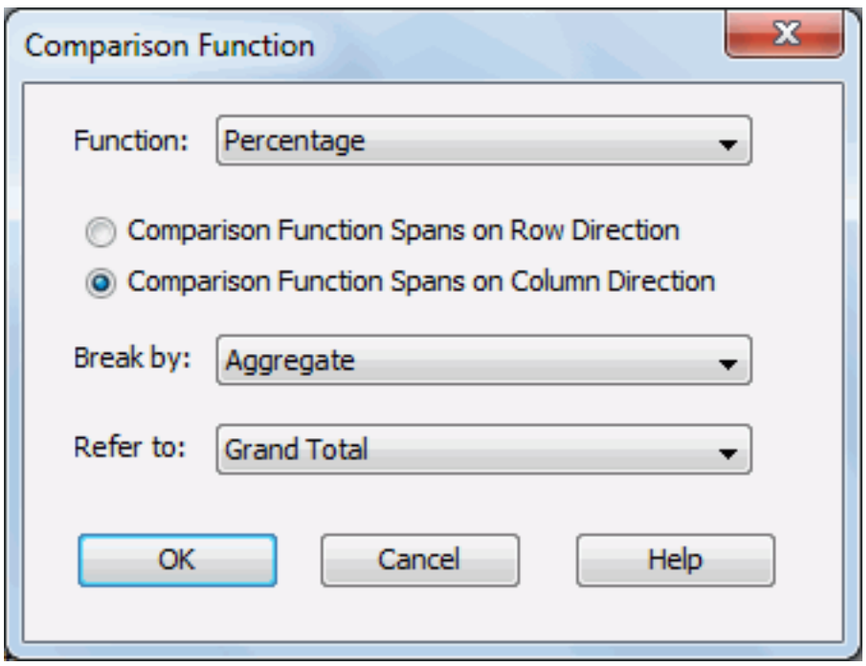
Does not retain any changes and closes the dialog.

Help

Displays the help document about this feature.

Comparison Function dialog

This dialog appears when you click the Comparison Function button in the Display screen of the crosstab wizard. It helps you to add a comparison function as an aggregation for a crosstab.



The following are details about options in the dialog:

Function

Specifies the function of the calculation: Percentage, Permillage, or Difference.

Comparison Function Spans on Row Direction

If checked, the comparison function will be placed into the column total cell of the crosstab.

Comparison Function Spans on Column Direction

If checked, the comparison function will be placed into the row total cell of the crosstab.

Break by

Items in the drop-down list vary with the position of the comparison function. It specifies the first parameter of the comparison function.

Refer to

Specifies the item as the other parameter of the comparison function.

OK

Accepts all changes and closes the dialog.

Cancel

Does not retain any changes and closes this dialog.

Help

Displays the help document about this feature.

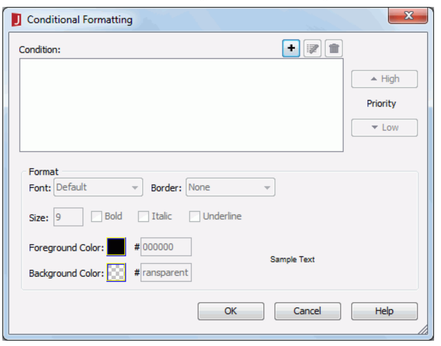
Related topics:

- Defining comparison functions in a crosstab

Conditional Formatting dialog

This dialog appears when you select a field/geographic map and then click Format > Conditional Formatting on the menu bar, or click the Conditional Formatting button on the toolbar, or right-click the field/geographic map and select Conditional Formatting from the shortcut menu. It helps you to add some conditional formats to values of the selected field or markers of specific group levels in a geographic map. The options in the dialog varies according to different sources it is opened from.

When you select a field to open the dialog, the options in the dialog are as follows.



Condition

Displays all the conditions you have already added.

- Adds a new condition in the Edit Conditions dialog.
- Edits the selected condition.
- Removes the selected condition.
- **Priority**
  Specifies the priority of each condition.
  - **High**
    Moves a condition up for a higher priority.
  - **Low**
    Moves a condition down for a lower priority.

Format

Specifies the format which will be applied to the field values when the specified condition is fulfilled.

- **Font**
  Specifies the font type for the field values.
- **Border**
  Specifies the border line style for the field values.
- **Size**
  Specifies the font size for the field values.
- **Bold**
  Specifies whether or not to bold the field values.
- **Italic**
  Specifies whether or not to make the field values italic.
- **Underline**
  Specifies whether or not to underline the field values.
- **Foreground Color**
  Specifies the foreground color for the field values.
- **Background Color**
  Specifies the background color for the field values.

  To specify the color, click the color image button and select a color swatch or type the hexadecimal value of the desired color (for example, 0xff0000) in the text box directly. If the color swatches cannot meet your requirement, you can click More Colors to open the Pick a Color dialog, with which you can set a full range of color.
- **Sample Text**
  Displays a preview sample of your selection.

OK

Applies the changes and closes the dialog.

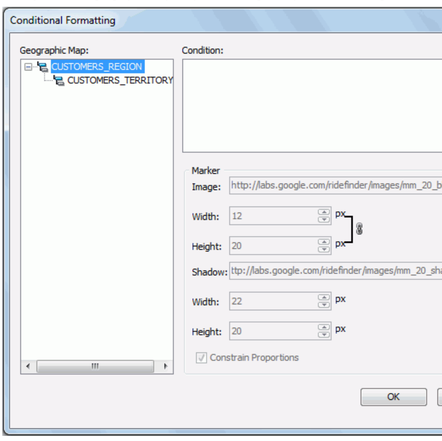
Cancel

Cancels the changes and exits the dialog.

Help

Displays the help document about this feature.

When you select a geographic map to open the dialog, the options in the dialog are as follows.



Geographic Map

Lists all the group levels in the geographic map.

Condition

Displays all the conditions you have already added.

- Adds a new condition in the Edit Conditions dialog.
- Edits the selected condition.
- Removes the selected condition.
- **Priority**
  Specifies the priority of each condition.
  - **High**
    Moves a condition up for a higher priority.
  - **Low**
    Moves a condition down for a lower priority.

Marker

Specifies the properties of the marker. Available only when there is at least one condition in the Condition box.

- **Image**
  Specifies the image of the marker. Input the URL for a web image or click Browse button to specify a local image.
  - **Width**
    Specifies the width of the foreground image in inches or centimeters.
  - **Height**
    Specifies the height of the foreground image in inches or centimeters.
- **Shadow**
  Specifies the shadow image of the marker. Input the URL for a web image or click Browse button to specify a local image. Only available for Google Maps types.
  - **Width**
    Specifies the width of the shadow image in inches or centimeters.
  - **Height**
    Specifies the height of the shadow image in inches or centimeters.

Constrain Proportions

Specifies whether to change the width and height for the marker image/shadow at the same time in a certain proportion. If checked, when the width/height is changed, the height/width will also be changed in a certain proportion.

OK

Applies the changes and closes the dialog.

Cancel

Cancels the changes and exits the dialog.

Help

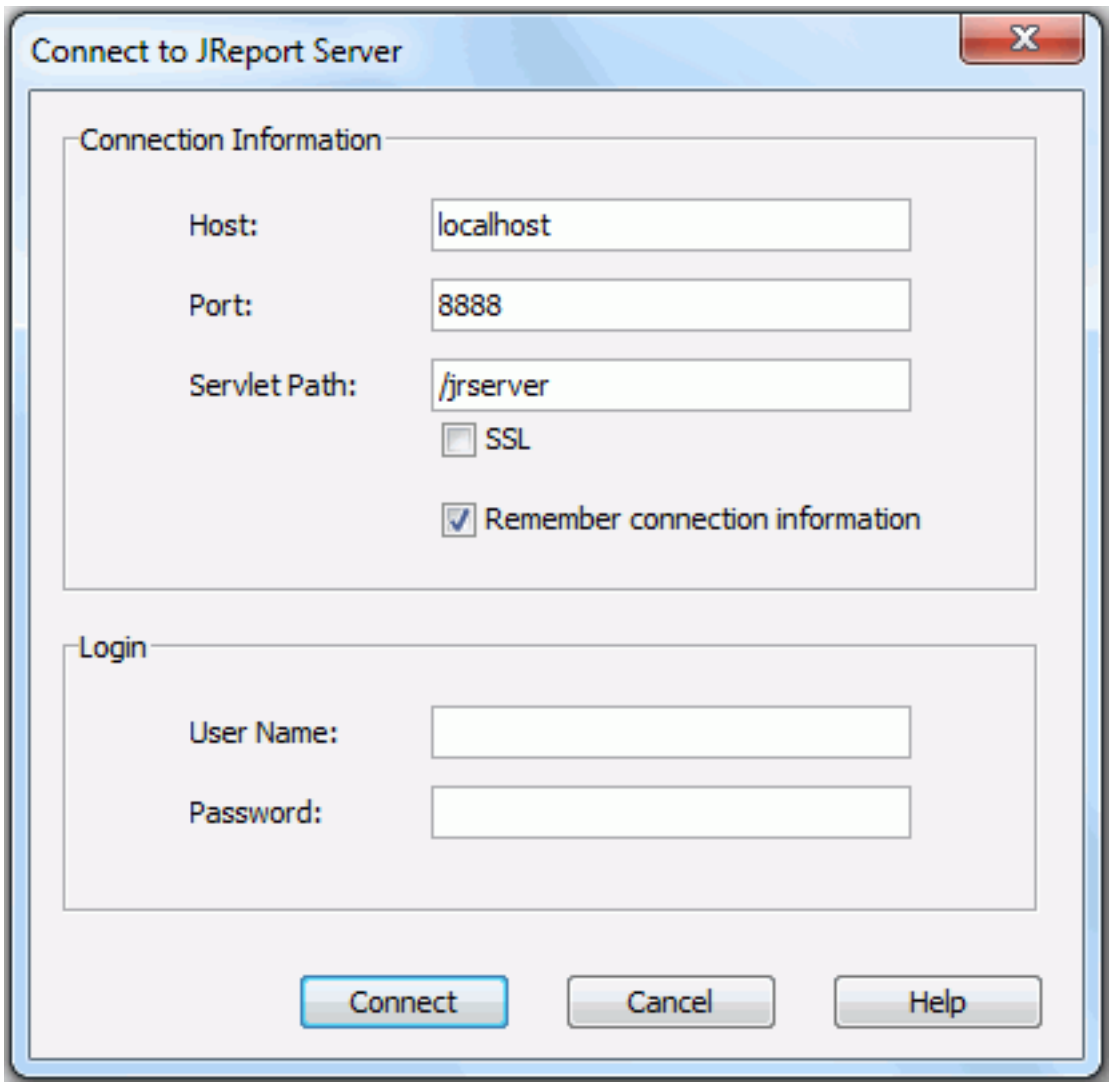
Displays the help document about this feature.

Related topics:

- Adding conditional formats to a DBField
- Adding conditional formats to geographic map markers

Connect to JReport Server dialog

This dialog helps you to connect to JReport Server in order to publish resources to the server, download resources from the server, get global NLS resources defined on the server, or import users, groups and roles from the server.



The following are details about options in the dialog:

Connection Information

Specifies information for connecting to the server.

- **Host**
  Specifies the host of the remote web server.
- **Port**
  Specifies the port that the web server listens to.
- **Servlet Path**
  It is `/jrserver`, and is used in the URL to access the servlet.
- **SSL**
  Abbreviation of Security Socket Layer. Check this option to specify whether to create an SSL connection when the server you are using is integrated with another web server which supports SSL.
- **Remember connection information**
  If checked, the connection information will be remembered by JReport Designer, and next time when you open this dialog, the connection information you have specified will be automatically displayed in the dialog.

Login

Specifies the user name and password for login to the server.

- **User Name**
  Specifies the user name used to access JReport Server.

  When the organization feature is enabled and when the user is an organization user, User Name should include the organization name. Use \ to separate the organization name and the user name, for example, org1\user1.
- **Password**
  Specifies the password of the user name.

Connect

Connects to JReport Server.

Cancel

Does not retain the changes and closes the dialog.

Help

Displays the help document about this feature.

Create Banded Object wizard

This wizard appears when you click Insert > Banded Object on the menu bar, or click the Insert Banded Object button on the Insert toolbar. It helps you to insert a banded object into a report, and consists of the following screens:

- Data screen
- Display screen
- Group screen
- Summary screen
- Filter screen
- Style screen

Back
Goes back to the previous screen.

Next
Goes to the next screen.

Finish
Finishes creating the banded object and closes this dialog.

Cancel
Does not retain changes and closes this dialog.

Help
Displays the help document about this feature.

Data screen

Specifies the dataset that you want to use to create the banded report.

Define a dataset for the object

- **New**
  If checked, select a data source from the catalog resources to create a dataset that will be used to build the banded object. When you choose to create the dataset from a query, you can click the Modify button to edit the query in the Query Editor if required.
- **Existing**
  If checked, select a dataset from the ones existing in the open report to create the banded object. Click the Modify button to edit the dataset in the Dataset Editor if required.
- **Current**
  If checked, the current dataset used by the parent object will be applied to the banded object.

Display screen

Specifies the data fields that you want to display in the banded object.

Resources
Lists all the available data resources.

Resources
Lists all the available data resources.

Group By
Lists the fields that are used to group data in the banded object.

Sort
Specifies how groups at the specific group level will be sorted.

- **Ascend**
  Groups will be sorted in an ascending order (A, B, C).
- **Descend**
  Groups will be sorted in a descending order (C, B, A).
- **No Sort**
  Groups will be sorted in the original order in database.

- **Special Group**
  Opens the User Defined Group dialog to define grouping information.
  For example, if you placed a field named Region for grouping, and this field contains all 50 states of the United States; and if you want to see the data between Maryland (MD) and New York (NY), you can define the criteria by selecting the between operator to further define your grouping information.
- **Custom Sort**
  Opens the Custom Sort dialog to set how groups will be sorted.

Special Function
If the group by field is of Numeric/String/Date/Time type, you can select a special function for the field in the Special Function column to further specify to which level the data will be grouped by.
If Customers is selected, the Customized Function dialog will be displayed, in which you can set the function by your own.

Custom Sort
Specifies how to sort the groups. Activated only when you have selected Custom Sort from the Sort column to define the sorting manner of groups for the selected group level.

Special Group
Specifies how to group your information. Activated only when you have clicked Special Group from the Sort column to define a special group.

Select N
Opens the Select N dialog to specify the Select N condition.

Group Filter
Opens the Group Filter dialog to specify the group filter condition.

Summary screen

Specifies the fields on which you want to create aggregate functions.

Resources
Lists all the available data resources.

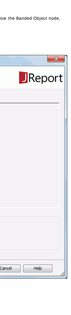
Summarized Fields
Lists all the fields that you want to display and to be summarized in the banded object.

Aggregate Function
Specifies the function to summarize the selected field.

Break Field
Displays a defined field on which the summary will be calculated. If a summary field is added below the Banded Object node, the break field is null and the summary will be calculated on the whole dataset.

Filter screen

Specifies to filter data displayed in the banded object.



For details about options in the screen, see Edit Filter dialog.

Style screen

Specifies the style of the banded object.

Group Report
Specifies the layout of the banded object.

- **Vertically**
  Creates a vertical banded object.
- **Horizontally**
  Creates a horizontal banded object.
- **Mailing Label Format**
  Creates a cross banded object in the form of a mailing label layout.

Style
Specifies the style of the banded object.

Preview
Displays the selected layout and style effects.

Inherit Style
Specifies whether to make the banded object take the style of its parent. Available only when the banded object is to be inserted into another banded object.

Related topics:

- Inserting a banded object

Create Chart wizard for page report

This wizard appears when you click Insert > Chart on the menu bar, click the Insert Chart button on the Insert toolbar, or drag Chart from the Toolbox panel into a page report. It helps you to create a chart in a page report, and consists of the following screens:

- Data Screen
- Type screen
- Display screen
- Filter screen
- Layout screen
- Style screen

Back

Goes back to the previous screen.

Next

Goes to the next screen.

Finish

Finishes creating a chart and closes this wizard.

Cancel

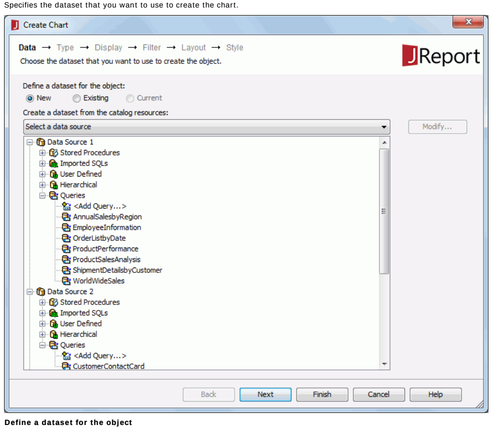
Does not retain changes and closes this wizard.

Help

Displays the help document about this feature.

Data screen

Specifies the dataset that you want to use to create the chart.



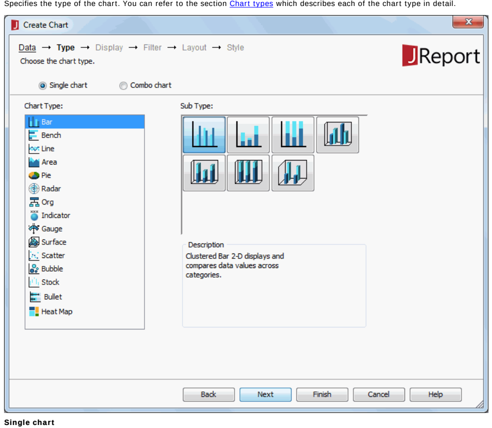
Define a dataset for the object

Specifies the dataset on which the chart will be built.

- **New**
  If checked, select a data source from the catalog resources to create a dataset that will be used to build the chart. When you choose to create the dataset from a query, you can click the Modify button to edit the query in the Query Editor if required.
- **Existing**
  If checked, select a dataset from the ones existing in the open report to create the chart. Click the Modify button to edit the dataset in the Dataset Editor if required.
- **Current**
  If checked, the current dataset used by the parent object will be applied to the chart.

Type screen

Specifies the type of the chart. You can refer to the section Chart types which describes each of the chart type in detail.



Single chart

Specifies to create a single chart.

- **Chart Type**
  Lists all the chart types that can be used to create a single chart.
- **Sub Type**
  Lists all the sub types of the selected chart type.
- **Description**
  Displays the description of the selected chart type.

Combo chart

Specifies to make more than one type use the primary or secondary axis. The types should be featured as Combination charts. Two types of data markers are used to represent different data values.

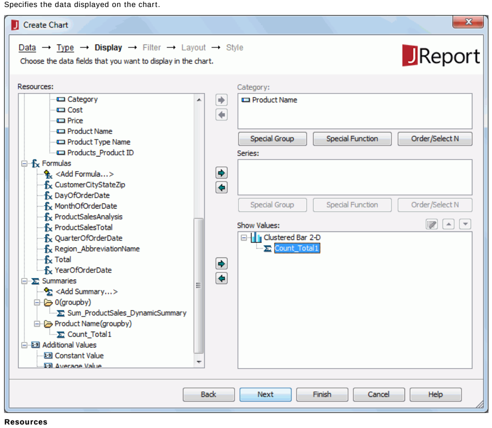
- **Chart Type**
  Lists all the chart types that can be added as a combo chart.
- **Sub Type**
  Lists all the sub types of the selected chart type.
- **Description**
  Displays the description of the selected chart type.
- **Chart Type Groups**
  Lists all the chart types that are to be used in the combo chart.
  - **Primary Axis**
    - **<Add Combo Type>**
      Adds a chart type to the primary axis.
  - **Secondary Axis**
    - **<Add Combo Type>**
      Adds a chart type to the secondary axis.
- Removes a specified chart type from the Chart Type Groups box.

Display screen

The Display screen differs with chart type: common chart types, org chart, or heat map.

Display screen for common chart types

Specifies the data displayed on the chart.



Resources

Lists all the available data resources in the selected dataset.

Adds the selected fields to be displayed in the chart.

Removes the selected fields from the chart if they are not required.

Category/Series

Lists the field you want to display on the category/series axis of the chart. For a real time chart, if no field is specified on the category axis, the text Use System Refresh Time will be automatically displayed in the Category box, namely, the time at which the chart refreshes itself will be used as the category value; the series axis will be disabled.

- **Special Group**
  Opens the User Defined Group dialog to define how to group information.
- **Special Function**
  Opens the Special Function dialog to define some special function to the field.
- **Order/Select N**
  Opens the Category Options dialog or Series Options dialog to specify the sort order of the category or series values and define the number of the category or series values that will be displayed in the chart.

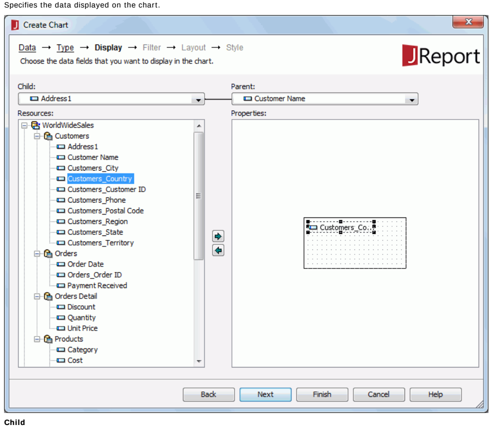
Show Values

Lists the values you want to show in the chart. For a real time chart, the values you add must be of Numeric type and cannot be summary fields.

- Opens the Edit Additional Value dialog to edit an additional value. Available only when a constant value or an average value is selected in the Show Values box.
- Moves the selected value one step up.
- Moves the selected value one step down.
- Adds a new pair of Y Axis and Radius for the bubble chart.
- Removes the selected field from the bubble chart if they are not required.
- **X Axis**
  Lists the value you want to show on the X axis of the bubble chart.
- **Y Axis**
  Lists the value you want to show on the Y axis of the bubble chart.
- **Radius**
  Lists the value you want to show as the bubble radius.

Display screen for org chart

Specifies the data displayed on the chart.



Child

Specify a field from the drop-down list which identifies the entity.

Parent

Specify a field from the drop-down list which shows the "reporting to" relationship among the entity members, that is, which child field member reports to or belongs to which child field member.

Resources

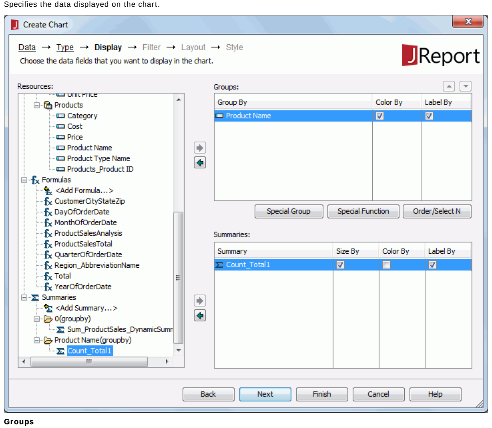
Lists all the available resources that can be added in the org chart.

Properties

The Properties box presents a node model of the org chart. Data fields, labels and images can be inserted into the node as the information about the entity in the org chart.

Display screen for heat map

Specifies the data displayed on the chart.



Groups

Lists the fields used to group the data. There should be at least one group.

- **Color By**
  Specifies whether to color by a group. 0-n groups can be used as the color-by fields.
- **Label By**
  Specifies whether to show a group field in the innermost rectangle. 0-n groups can be used as the label-by fields.
- **Special Group**
  Opens the User Defined Group dialog to define how to group information.
- **Special Function**
  Opens the Special Function dialog to define some special function to the field.
- **Order/Select N**
  Opens the Group Options dialog to specify the sort order of the group values and define the number of the group values that will be displayed in the chart.

Summaries

Lists the summaries used as size-by/color-by or displayed in the innermost rectangle.

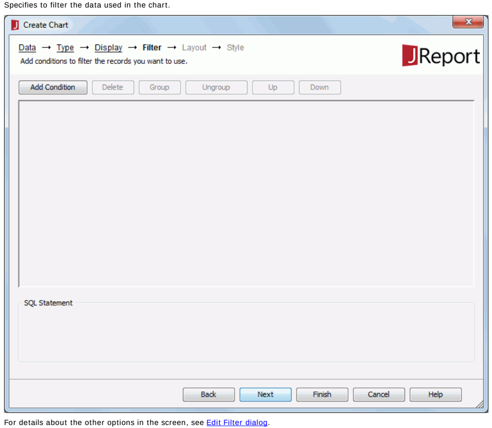
The summaries should match the groups. For example, if the groups level is A > B > C, the static summaries grouped by C can be inserted into the Summaries box, but the static summaries grouped by A, B or other fields cannot.

If no group is specified in the Groups box, you can insert any static summary. And its group-by field will be inserted into the Groups box automatically.

- **Size By**
  Specifies to size by one summary or none.
- **Color By**
  Specifies to color by one summary or none.
- **Label By**
  Specifies whether to show a summary in the innermost rectangle. 0-n summaries can be used as the label-by fields.

Filter screen

Specifies to filter the data used in the chart.



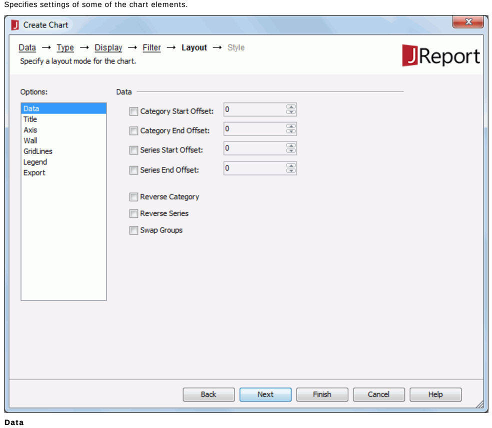
For details about the other options in the screen, see Edit Filter dialog.

Layout screen

The Layout screen differs with chart type: common chart types, org chart, or heat map.

Layout screen for common chart types

Specifies settings of some of the chart elements.



Data

- **Category Start Offset**
  Specifies the start offset of category from which the chart will be displayed.
- **Category End Offset**
  Specifies the end offset of category to which the chart will be displayed.
- **Series Start Offset**
  Specifies the start offset series from which the chart will be displayed.
- **Series End Offset**
  Specifies the end offset of series to which the chart will be displayed.
- **Reverse Category**
  Specifies whether to reverse the display order of the categories.
- **Reverse Series**
  Specifies whether to reverse the display order of the series.
- **Swap Groups**
  Specifies to display values from different data fields by switching data between the category and series axes, the category and values axes.

Notes:

- If values are set to be less than 0, it means the categories or series will be displayed without any offset.
- If values are set to be greater than 0 and the value of Category/Series Start Offset is greater than that of Category/Series End Offset, all of the categories or series will be displayed.

Title

- **Chart Title**
  Specifies the title of the chart.
- **Category (X) Axis Title**
  Specifies the title of the X-axis.
- **Value (Y) Axis Title**
  Specifies the title of the Y-axis.
- **Value (Y2) Axis Title**
  Specifies the title of the Y2-axis.

Axis

- **Show Category (X) Axis**
  Specifies whether to show the category (X) axis in this chart.
- **Show Value (Y) Axis**
  Specifies whether to show the value (Y) axis in this chart.
- **Show Value (Y2) Axis**
  Specifies whether to show the value (Y2) axis in this chart.
- **Show Series (Z) Axis**
  Specifies whether to show the series (Z) axis in this chart.

Wall

- **Show Wall**
  Specifies whether to show the wall in this chart.
- **Show Floor**
  Specifies whether to show the floor in this chart.

Gridlines

- **Show Category (X) Axis Gridlines**
  Specifies whether to show the gridlines of category (X) axis.
- **Show Value (Y) Axis Gridlines**
  Specifies whether to show the gridlines of value (Y) axis.
- **Show Value (Y2) Axis Gridlines**
  Specifies whether to show the gridlines of value (Y2) axis.
- **Show Series (Z) Axis Gridlines**
  Specifies whether to show the gridlines of series (Z) axis.

Legend

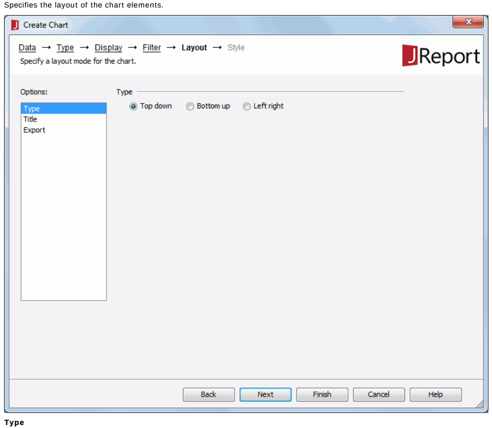
- **Show Legend**
  Specifies whether to show the legend in this chart.

Export

- **Mapping Component**
  This option is disabled when creating a chart. It specifies the component to which the chart will be mapped (for details, see Creating dynamic chart in Excel).

Layout screen for organization chart (org)

Specifies the layout of the chart elements.



Type

Specifies the layout mode of the org chart.

- **Top down**
  The org chart tree expands from top to bottom.
- **Bottom up**
  The org chart tree expands from bottom to top. It is the reverse of the top down type.
- **Left right**
  The org chart tree expands from left to right.

Title

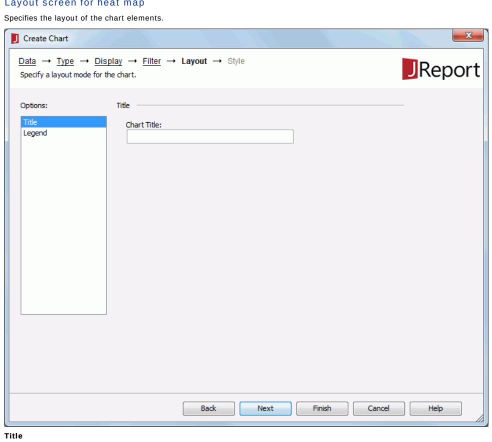
- **Chart Title**
  Specifies the title of the chart.

Export

- **Mapping Component**
  Specifies the component to which the chart will be mapped. For details, see Creating dynamic chart in Excel.

Layout screen for heat map

Specifies the layout of the chart elements.



Title

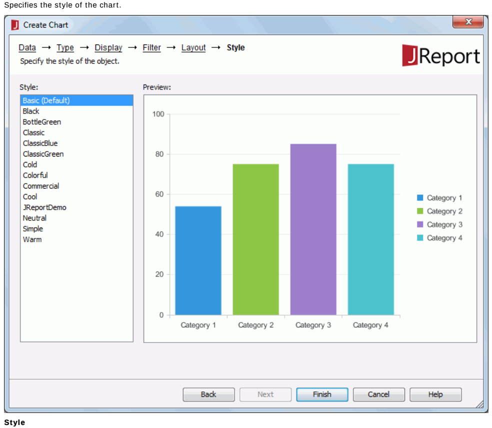
- **Chart Title**
  Specifies the title of the chart.

Legend

- **Show Legend**
  Specifies whether to show the legend in this chart.

Style screen

Specifies the style of the chart.



Style

Specifies the style of the chart.

Preview

Displays the selected style and layout effects.

Inherit Style

Specifies whether to make the chart take the style of its parent. Available only when the chart is to be inserted into a table or banded object in a page report.

Related topics:

- Inserting a chart

Create Chart wizard for web report/library component

This wizard appears when you click Insert > Chart on the menu bar, click the Insert Chart button on the Insert toolbar, or drag Chart from the Toolbox panel into a web report or library component. It helps you to create a chart in a web report or library component, and consists of the following screens:

- Data Screen
- Type screen
- Display screen
- Filter screen
- Layout screen
- Style screen

Back

Goes back to the previous screen.

Next

Goes to the next screen.

Finish

Finishes creating a chart and closes this wizard.

Cancel

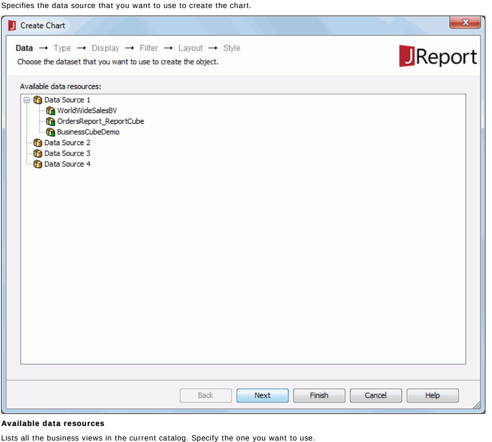
Does not retain changes and closes this wizard.

Help

Displays the help document about this feature.

Data screen

Specifies the data source that you want to use to create the chart.

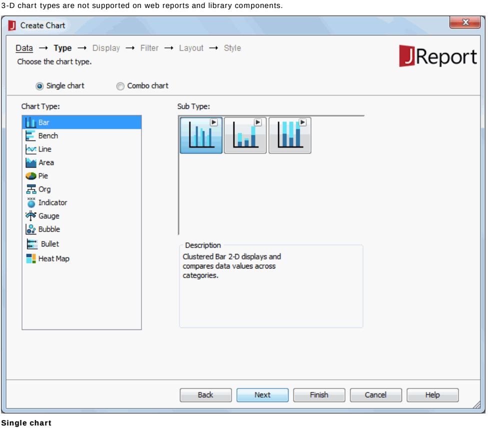


Available data resources

Lists all the business views in the current catalog. Specify the one you want to use.

Type screen

Specifies the type of the chart. You can refer to the section Chart types which describes each of the chart type in detail. All the 3-D chart types are not supported on web reports and library components.



Single chart

Specifies to create a single chart.

- **Chart Type**
  Lists all the chart types that can be used to create a single chart.
- **Sub Type**
  Lists all the sub types of the selected chart type.
- **Description**
  Displays the description of the selected chart type.

Combo chart

Specifies to make more than one type use the primary or secondary axis. The types should be featured as Combination charts. Two types of data markers are used to represent different data values.

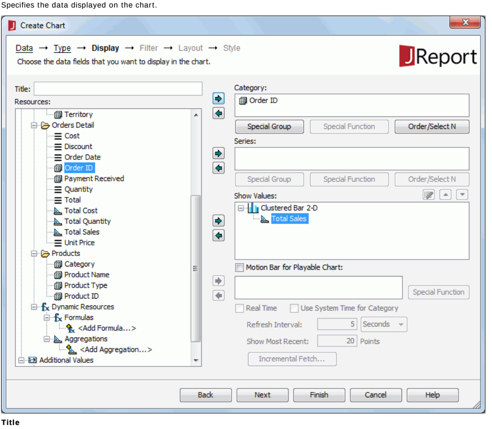
- **Chart Type**
  Lists all the chart types that can be added as a combo chart.
- **Sub Type**
  Lists all the sub types of the selected chart type.
- **Description**
  Displays the description of the selected chart type.
- **Chart Type Groups**
  Lists all the chart types that are to be used in the combo chart.
  - **Primary Axis**
    - **<Add Combo Type>**
      Adds a chart type to the primary axis.
  - **Secondary Axis**
    - **<Add Combo Type>**
      Adds a chart type to the secondary axis.
- Removes a specified chart type from the Chart Type Groups box.

Display screen

The Display screen differs with chart type: common chart types, org chart, or heat map.

Display screen for common chart types

Specifies the data displayed on the chart.



Title

Specifies the title of the chart.

Resources

Lists all the available data resources in the selected dataset. You can also create dynamic formulas and aggregations to use in the chart.

Adds the selected fields to be displayed in the chart.

Removes the selected fields from the chart if they are not required.

Category/Series

Lists the field you want to display on the category/series axis of the chart. For a real time chart, if no field is specified on the category axis, Use System Refresh Time will be automatically displayed in the Category box, namely, the time at which the chart refreshes itself will be used as the category value; the series axis will be disabled.

- **Special Group**
  Opens the User Defined Group dialog to define how to group information.
- **Special Function**
  Opens the Special Function dialog to define some special function to the field.
- **Order/Select N**
  Opens the Category Options dialog or Series Options dialog to specify the sort order of the category or series values and define the number of the category or series values that will be displayed in the chart.

Show Values

Lists the values you want to show in the chart. For a real time chart, the values you add must be of Numeric type and cannot be summary fields.

- Opens the Edit Additional Value dialog to edit an additional value. Available only when a constant value or an average value is selected in the Show Values box.
- Moves the selected value one step up.
- Moves the selected value one step down.
- Adds a new pair of Y Axis and Radius for the bubble chart.
- Removes the selected field from the bubble chart if they are not required.
- **X Axis**
  Lists the value you want to show on the X axis of the bubble chart.
- **Y Axis**
  Lists the value you want to show on the Y axis of the bubble chart.
- **Radius**
  Lists the value you want to show as the bubble radius.

Motion Bar for Playable Chart

Lists the field you want to use as the motion field. A motion field can only be of Integer, Date or Time data type. Available to single bar, bench and bubble charts in web reports and library components only.

- **Special Function**
  Opens the Special Function dialog to define some special function to the motion field. Available only when the motion field is of Date data type.

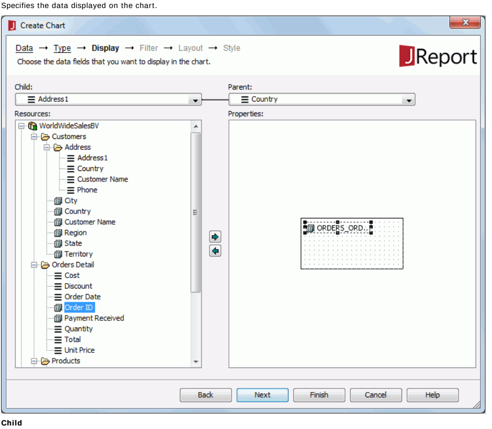
Real Time

Specifies to run the chart in real time mode, which means it will be updated automatically by using real time data after it is published to JReport Server. Available to single bar, bench, line, and area charts in web reports and library components only.

- **Use System Time for Category**
  Specifies to use the time at which the chart refreshes itself as the category value.
- **Refresh Interval**
  Specifies the time interval at which the chart will get data and refresh itself automatically.
- **Show Most Recent N Points**
  Specifies the number of points that will be kept for the real time data on the chart.
- **Incremental Fetch**
  Opens the Unique Key dialog to configure a unique key for the real time chart.

Display screen for org chart

Specifies the data displayed on the chart.



Child

Specify a field from the drop-down list which identifies the entity.

Parent

Specify a field from the drop-down list which shows the "reporting to" relationship among the entity members, that is, which child field member reports to or belongs to which child field member.

Resources

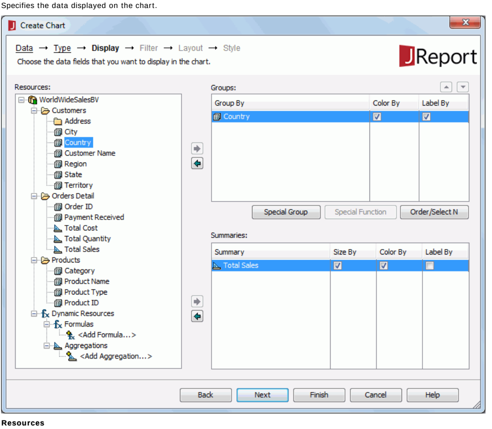
Lists all the available resources that can be added to the org chart. You can also create dynamic formulas and aggregations to use in the chart.

Properties

The Properties box presents a node model of the org chart. Data fields, labels and images can be inserted into the node as the information about the entity in the org chart.

Display screen for heat map

Specifies the data displayed on the chart.



Resources

Lists all the available data resources in the selected dataset. You can also create dynamic formulas and aggregations to use in the chart.

Groups

Lists the fields used to group the data. There should be at least one group.

- **Color By**
  Specifies whether to color by a group. 0-n groups can be used as the color-by fields.
- **Label By**
  Specifies whether to show a group field in the innermost rectangle. 0-n groups can be used as the label-by fields.
- **Special Group**
  Opens the User Defined Group dialog to define how to group information.
- **Special Function**
  Opens the Special Function dialog to define some special function to the field.
- **Order/Select N**
  Opens the Group Options dialog to specify the sort order of the group values and define the number of the group values that will be displayed in the chart.

Summaries

Lists the summaries used as size-by/color-by or displayed in the innermost rectangle.

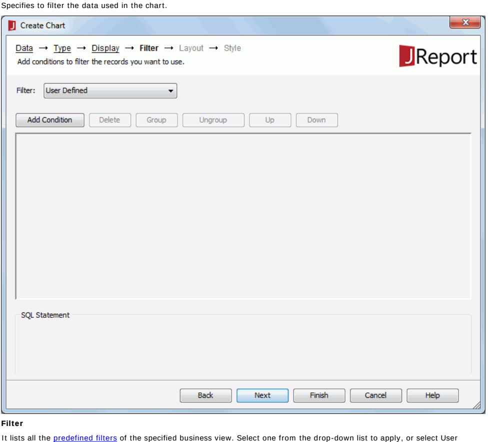
The summaries should match the groups. For example, if the groups level is A > B > C, the static summaries grouped by C can be inserted into the Summaries box, but the static summaries grouped by A, B or other fields cannot.

If no group is specified in the Groups box, you can insert any static summary. And its group-by field will be inserted into the Groups box automatically.

- **Size By**
  Specifies to size by one summary or none.
- **Color By**
  Specifies to color by one summary or none.
- **Label By**
  Specifies whether to show a summary in the innermost rectangle. 0-n summaries can be used as the label-by fields.

Filter screen

Specifies to filter the data used in the chart.



Filter

It lists all the predefined filters of the specified business view. Select one from the drop-down list to apply, or select User Defined and define a new filter according to your requirements.

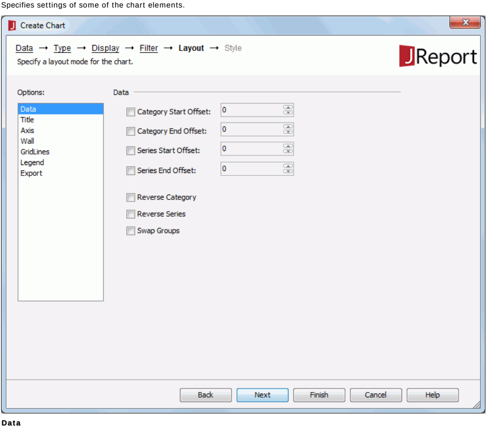
For details about the other options in the screen, see Edit Filter dialog.

Layout screen

The Layout screen differs with chart type: common chart types, org chart, or heat map.

Layout screen for common chart types

Specifies settings of some of the chart elements.



Data

- **Category Start Offset**
  Specifies the start offset of category from which the chart will be displayed.
- **Category End Offset**
  Specifies the end offset of category to which the chart will be displayed.
- **Series Start Offset**
  Specifies the start offset series from which the chart will be displayed.
- **Series End Offset**
  Specifies the end offset of series to which the chart will be displayed.
- **Reverse Category**
  Specifies whether to reverse the display order of the categories.
- **Reverse Series**
  Specifies whether to reverse the display order of the series.
- **Swap Groups**
  Specifies to display values from different data fields by switching data between the category and series axes, the category and values axes.

Notes:

- If values are set to be less than 0, it means the categories or series will be displayed without any offset.
- If values are set to be greater than 0 and the value of Category/Series Start Offset is greater than that of Category/Series End Offset, all of the categories or series will be displayed.

Title

- **Chart Title**
  Specifies the title of the chart.
- **Category (X) Axis Title**
  Specifies the title of the X-axis.
- **Value (Y) Axis Title**
  Specifies the title of the Y-axis.
- **Value (Y2) Axis Title**
  Specifies the title of the Y2-axis.

Axis

- **Show Category (X) Axis**
  Specifies whether to show the category (X) axis in this chart.
- **Show Value (Y) Axis**
  Specifies whether to show the value (Y) axis in this chart.
- **Show Value (Y2) Axis**
  Specifies whether to show the value (Y2) axis in this chart.
- **Show Series (Z) Axis**
  Specifies whether to show the series (Z) axis in this chart.

Wall

- **Show Wall**
  Specifies whether to show the wall in this chart.
- **Show Floor**
  Specifies whether to show the floor in this chart.

Gridlines

- **Show Category (X) Axis Gridlines**
  Specifies whether to show the gridlines of category (X) axis.
- **Show Value (Y) Axis Gridlines**
  Specifies whether to show the gridlines of value (Y) axis.
- **Show Value (Y2) Axis Gridlines**
  Specifies whether to show the gridlines of value (Y2) axis.
- **Show Series (Z) Axis Gridlines**
  Specifies whether to show the gridlines of series (Z) axis.

Legend

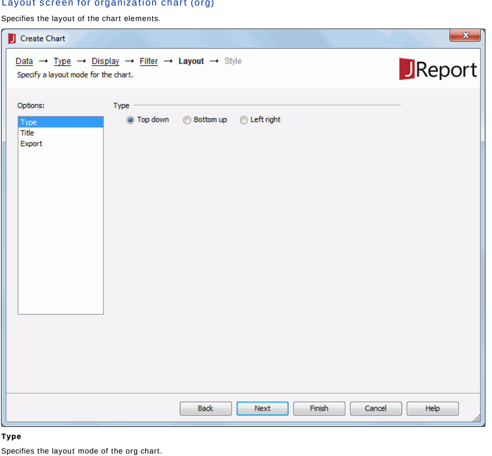
- **Show Legend**
  Specifies whether to show the legend in this chart.

Export

- **Mapping Component**
  This option is disabled when creating a chart. It specifies the component to which the chart will be mapped (for details, see Creating dynamic chart in Excel).

Layout screen for organization chart (org)

Specifies the layout of the chart elements.



Type

Specifies the layout mode of the org chart.

- **Top down**
  The org chart tree expands from top to bottom.
- **Bottom up**
  The org chart tree expands from bottom to top. It is the reverse of the top down type.
- **Left right**
  The org chart tree expands from left to right.

Title

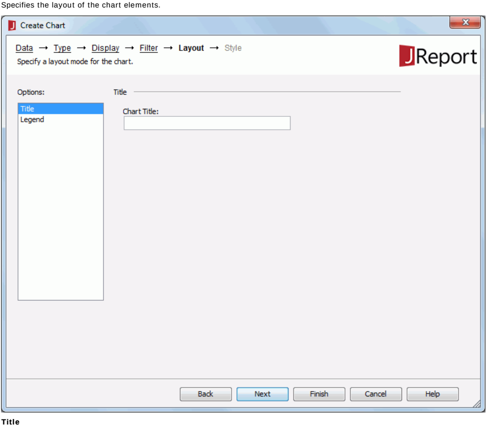
- **Chart Title**
  Specifies the title of the chart.

Export

- **Mapping Component**
  Specifies the component to which the chart will be mapped. For details, see Creating dynamic chart in Excel.

Layout screen for heat map

Specifies the layout of the chart elements.



Title

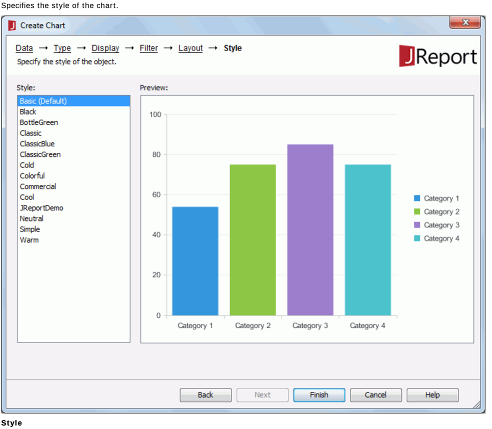
- **Chart Title**
  Specifies the title of the chart.

Legend

- **Show Legend**
  Specifies whether to show the legend in this chart.

Style screen

Specifies the style of the chart.



Style

Specifies the style of the chart.

Preview

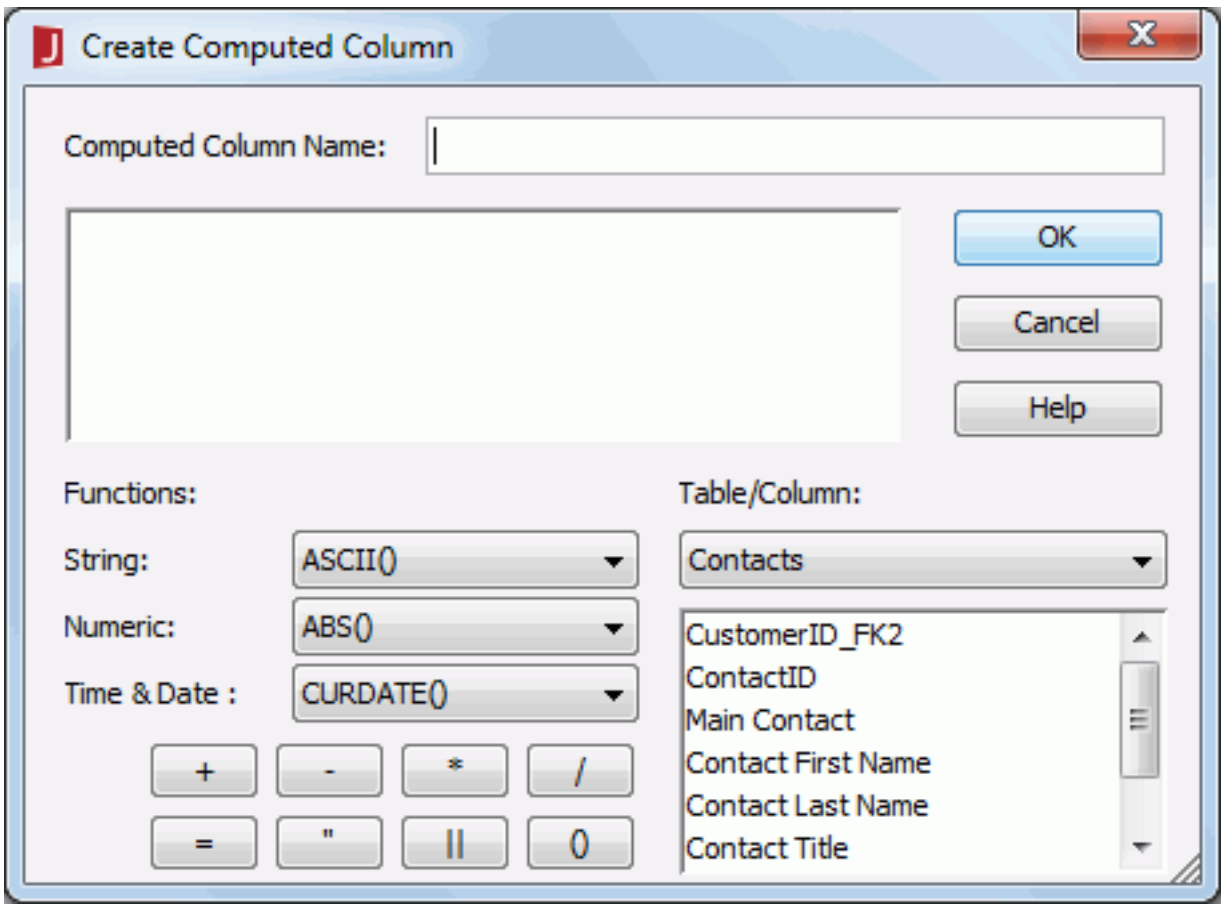
Displays the selected style and layout effects.

Related topics:

- Inserting a chart

Create Computed Column dialog

This dialog appears when you click Column > Create Computed Column on the menu bar of the Query Editor Window. It helps you to create functions quickly and easily instead of having to write formulas.



The following are details about options in the dialog:

Computed Column Name

Specifies the name of the computed column.

Functions

The functions here are not from JReport system. They are from the connected database you are using. Thus, if you change your data source, some of these functions may no longer exist. For each database, the different supported functions list will be returned.

- **String**
  Specifies String formulas to create a function.
- **Numeric**
  Specifies Numeric formulas to create a function.
- **Time & Date**
  Specifies Time or Date formulas to create a function.
- **+**
  Adds the numbers or fields together in the Expression menu.
- **-**
  Subtracts numbers or fields together in the Expression menu.
- *****
  Multiplies numbers or fields together in the Expression menu.
- **/**
  Divides numbers or fields together in the Expression menu.
- **=**
  Equates fields together.
- **"**
  Places quotations on long character strings or names that have blanks in them. For example, you should place quotes on values such as "New York" or "Washington DC").
- **||**
  Places fields together in the same Expression menu. For example, "New York" || "Washington DC").
- **()**
  Places your fields in parentheses.

Table/Column

Lists all the tables and fields which are available for creating the computed column.

OK

Applies all changes and closes the dialog.

Cancel

Does not retain any changes and closes the dialog.

Help

Displays the help document about this feature.

Related topics:

- Creating computed columns in a query

Create Crosstab wizard for page report

This wizard appears when you click Insert > Crosstab on the menu bar, or click the Insert Crosstab button on the Insert toolbar. It helps you to create a crosstab in a page report, and consists of the following screens:

- Data screen
- Display screen
- Filter screen
- Layout screen
- Style screen

Back
Goes back to the previous screen.

Next
Goes to the next screen.

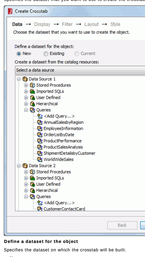
Finish
Finishes creating the crosstab and closes this dialog.

Cancel
Does not retain changes and closes this dialog.

Help
Displays the help document about this feature.

Data screen

Specifies the dataset that you want to use to create the crosstab.

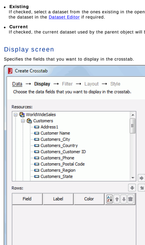


Define a dataset for the object
Specifies the dataset on which the crosstab will be built.

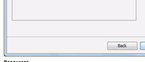
- **New**
  If checked, select a data source from the catalog resources to create a dataset that will be used to build the crosstab. When you choose to create the dataset from a query, you can click the Modify button to edit the query in the Query Editor if required.
- **Existing**
  If checked, select a dataset from the ones existing in the open report to create the crosstab. Click the Modify button to edit the dataset in the Dataset Editor if required.
- **Current**
  If checked, the current dataset used by the parent object will be applied to the crosstab.

Display screen

Specifies the fields that you want to display in the crosstab.



Resources
Lists all the available data resources.



Adds the selected field to be displayed on the columns of the crosstab.

Adds the selected field to be displayed on the rows of the crosstab.

Adds the selected field on which to create summaries.

Columns/Rows

- **Field**
  Lists the fields that will be displayed on the columns/rows of the crosstab.
- **Label**
  Specifies the display names for the selected fields. By default these are blank and no names will be created for the fields to label the columns/rows. You can double-click the cells to edit them if required.
- **Color**
  Specifies the background color of the selected fields.

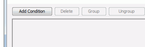
Summaries

- **Field**
  Lists the fields that you select to create summaries.
- **Aggregate**
  Specifies the functions used to summarize data of the selected fields. For details about each function, see Math functions.
- **Label**
  Specifies the display names for the selected fields. By default these are blank and no names will be created for the fields to label the summaries. You can double click the cells to edit them if required.
- **Comparison Function**
  Opens the Comparison Function dialog to add a comparison function as an aggregate for the crosstab.

Specifies in which manner to sort the field values.

Moves the specified field one step up.

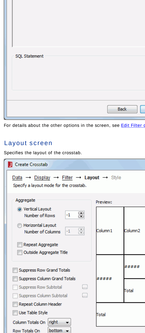
Moves the specified field one step down.



Removes the specified field that is not required in the crosstab.

Filter screen

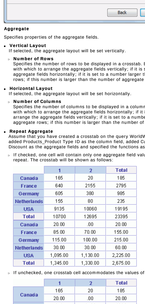
Specifies to filter data displayed in the crosstab.



For details about the other options in the screen, see Edit Filter dialog.

Layout screen

Specifies the layout of the crosstab.



Aggregate
Specifies properties of the aggregate fields.

- **Vertical Layout**
  If selected, the aggregate layout will be set vertically.
  - **Number of Rows**
    Specifies the number of rows to be displayed in a crosstab. By default, it is -1, which means that there is only one column with which to arrange the aggregate fields vertically. If it is set to 1, that means there will be only one row to arrange the aggregate fields horizontally. If it is set to 2, that means there will be only two columns to arrange the aggregate fields. If this number is larger than the number of aggregate fields in the crosstab, it will be treated as -1.
- **Horizontal Layout**
  If selected, the aggregate layout will be set horizontally.
  - **Number of Columns**
    Specifies the number of columns to be displayed in a column. By default, it is -1, which means that there is only one row with which to arrange the aggregate fields horizontally. If it is set to 1, that means there will be only one column to arrange the aggregate fields vertically. If it is set to 2, that means there will be only two rows to arrange the aggregate fields. If this number is larger than the number of aggregate fields in the crosstab, it will be treated as -1.
- **Repeat Aggregate**
  Assume that you have created a crosstab on the query WorldWideSales in the catalog file SampleReports.cat as follows: added Products, Product Type ID as the column field, added Customers, Country as the row field, added Quantity and Discount as the aggregate fields and specified the functions as Sum, and applied the style Classic to the crosstab.
  - If checked, one cell will contain only one aggregate field value, while for the other aggregate fields, the crosstab will repeat. The crosstab will be shown as follows:

	1	2	Total
Canada	165	20	185
France	640	2155	2795
Germany	605	380	985
Netherlands	155	80	235
USA	9135	10860	19195
Total	10700	13495	23395
Canada	20.00	00	20.00
France	85.00	70.00	155.00
Germany	115.00	100.00	215.00
Netherlands	30.00	30.00	60.00
USA	1,095.00	1,130.00	2,225.00
Total	1,345.00	1,330.00	2,675.00

  - If unchecked, one crosstab cell accommodates the values of all aggregate fields. The crosstab will be shown as follows:

	1	2	Total
Canada	165	20	185
	20.00	.00	20.00
France	640	2155	2795
	85.00	70.00	155.00
Germany	605	380	985
	115.00	100.00	215.00
Netherlands	155	80	235
	30.00	30.00	60.00
USA	9135	10860	19195
	1,095.00	1,130.00	2,225.00
Total	10700	12695	23395
	1,345.00	1,330.00	2,675.00

- **Outside Aggregate Title**
  Specifies to place the summary labels to the left of the row values if true, right of the row values if false.

Suppress Row Grand Totals
Specifies whether or not to show the grand totals of rows.

Suppress Column Grand Totals
Specifies whether or not to show the grand totals of columns.

Suppress Row Subtotal
Specifies whether or not to show the subtotals of rows. If required, you can click [...] to open the Suppress Row Subtotal dialog to specify which subtotals of the rows will be suppressed, and which will be shown.

Suppress Column Subtotal
Specifies whether or not to show the subtotals of columns. If required, you can click [...] to open the Suppress Column Subtotal dialog to specify which subtotals of the columns will be suppressed, and which will be shown.

Note: The settings of the above four options will only take effect in view mode.

Repeat Column Header
Specifies whether or not the header of column will be appeared on every page, if a crosstab spans more than one page.

Use Table Style
Specifies whether to add headers to the Total rows and columns.

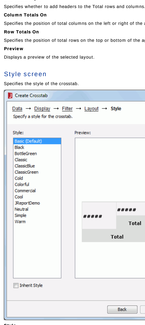
Column Totals On
Specifies the position of total columns on the left or right of the aggregates.

Row Totals On
Specifies the position of total rows on the top or bottom of the aggregates.

Preview
Displays a preview of the selected layout.

Style screen

Specifies the style of the crosstab.



Style
Specifies the style of the crosstab.

Preview
Shows a sketch of the selected style.

Inherit Style
Specifies whether to make the crosstab take the style of its parent. Available only when the crosstab is to be inserted into a banded object.

Related topics:

- Inserting a crosstab

Create Crosstab wizard for web report/library component

This wizard appears when you click Insert > Crosstab on the menu bar, click the Insert Crosstab button on the Insert toolbar, or drag Crosstab from the Toolbox panel into a web report or library component. It helps you to create a crosstab in a web report or library component, and consists of the following screens:

- Data screen
- Display screen
- Filter screen
- Layout screen
- Style screen

Back
Goes back to the previous screen.

Next
Goes to the next screen.

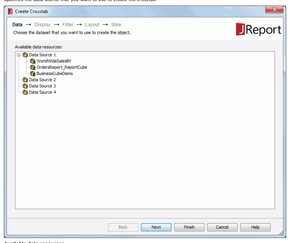
Finish
Finishes creating the crosstab and closes this dialog.

Cancel
Does not retain changes and closes this dialog.

Help
Displays the help document about this feature.

Data screen

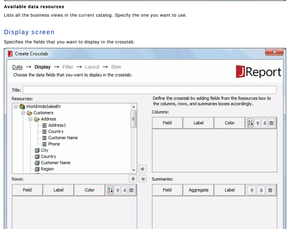
Specifies the data source that you want to use to create the crosstab.



Available data resources
Lists all the business views in the current catalog. Specify the one you want to use.

Display screen

Specifies the fields that you want to display in the crosstab.



Title
Specifies the title of the crosstab.

Resources
Lists all the available data resources. You can also create dynamic formulas and aggregations to use in the crosstab.

Adds the selected field to be displayed on the columns of the crosstab.

Adds the selected field to be displayed on the rows of the crosstab.

Adds the selected field on which to create summaries.

Columns/Rows

- **Field**
  Lists the fields that will be displayed on the columns/rows of the crosstab.
- **Label**
  Specifies the display names for the selected fields. By default these are blank and no names will be created for the fields to label the columns/rows. You can double-click the cells to edit them if required.
- **Color**
  Specifies the background color of the selected fields.

Summaries

- **Field**
  Lists the fields that you select to create summaries.
- **Aggregate**
  Specifies the functions used to summarize data of the selected fields. For details about each function, see Math functions.
- **Label**
  Specifies the display names for the selected fields. By default these are blank and no names will be created for the fields to label the summaries. You can double-click the cells to edit them if required.
- **Comparison Function**
  Opens the Comparison Function dialog to add a comparison function as an aggregate for the crosstab.

Specifies in which manner to sort the field values.

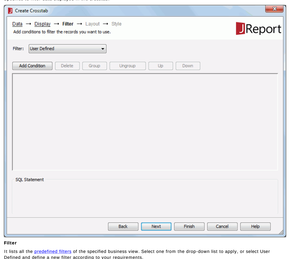
Moves the specified field one step up.

Moves the specified field one step down.

Removes the specified field that is not required in the crosstab.

Filter screen

Specifies to filter data displayed in the crosstab.

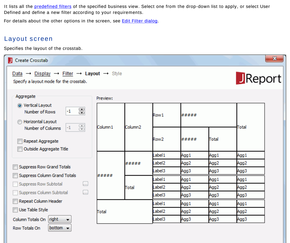


Filter
It lists all the predefined filters of the specified business view. Select one from the drop-down list to apply, or select User Defined and define a new filter according to your requirements.

For details about the other options in the screen, see Edit Filter dialog.

Layout screen

Specifies the layout of the crosstab.



Aggregate
Specifies properties of the aggregate fields.

- **Vertical Layout**
  If selected, the aggregate layout will be set vertically.
  - **Number of Rows**
    Specifies the number of rows to be displayed in a crosstab. By default, it is -1, which means that there is only one column with which to arrange the aggregate fields vertically; if it is set to 1, that means there will be only one row to arrange the aggregate fields; if it is set to 2, that means there will be two rows to arrange the aggregate fields; if the number of aggregate fields is larger than the number of aggregate fields in the crosstab, it will be treated as -1.
- **Horizontal Layout**
  If selected, the aggregate layout will be set horizontally.
  - **Number of Columns**
    Specifies the number of columns to be displayed in a column. By default, it is -1, which means that there is only one row with which to arrange the aggregate fields horizontally; if it is set to 1, that means there will be only one column to arrange the aggregate fields vertically; if it is set to a number larger than 1, that means there will be three columns to arrange the aggregate fields; if the number is larger than the number of aggregate fields in the crosstab, it will be treated as -1.
- **Repeat Aggregate**
  Assume that you created a crosstab on the query WorldWideSales in the catalog file SampleReports.cat as follows: added Products_Product Type ID as the column field, added Customers_Country as the row field, added Quantity and Discount as the aggregate fields and specified the functions as Sum, and applied the style Classic to the crosstab.
  - If checked, one cell will contain only one aggregate field value, while for the other aggregate fields, the crosstab will repeat. The crosstab will be shown as follows:

	1	2	Total
Canada	165	20	185
France	640	2155	2795
Germany	605	380	985
Netherlands	155	80	235
USA	9135	10060	19195
Total	10700	12695	23395
Canada	20.00	.00	20.00
France	85.00	70.00	155.00
Germany	115.00	100.00	215.00
Netherlands	30.00	30.00	60.00
USA	1,095.00	1,130.00	2,225.00
Total	1,345.00	1,330.00	2,675.00

  - If unchecked, one crosstab cell accommodates the values of all aggregate fields. The crosstab will be shown as follows:

	1	2	Total
Canada	165	20	185
	20.00	.00	20.00
France	640	2155	2795
	85.00	70.00	155.00
Germany	605	380	985
	115.00	100.00	215.00
Netherlands	155	80	235
	30.00	30.00	60.00
USA	9135	10060	19195
	1,095.00	1,130.00	2,225.00
Total	10700	12695	23395
	1,345.00	1,330.00	2,675.00

- **Outside Aggregate Total**
  Specifies to place the summary labels to the left of the row values if true, right of the row values if false.

Suppress Row Grand Totals
Specifies whether or not to show the grand totals of rows.

Suppress Column Grand Totals
Specifies whether or not to show the grand totals of columns.

Suppress Row Subtotal
Specifies whether or not to show the subtotals of rows. If required, you can click to open the Suppress Row Subtotal dialog to specify which subtotals of the rows will be suppressed, and which will be shown.

Suppress Column Subtotal
Specifies whether or not to show the subtotals of columns. If required, you can click to open the Suppress Column Subtotal dialog to specify which subtotals of the columns will be suppressed, and which will be shown.

Note: The settings of the above four options will only take effect in view mode.

Repeat Column Header
Specifies whether or not the header of column will be appeared on every page, if a crosstab spans more than one page.

Use Table Style
Specifies whether to add headers to the Total rows and columns.

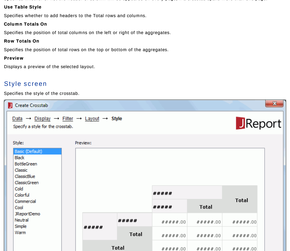
Column Totals On
Specifies the position of total columns on the left or right of the aggregates.

Row Totals On
Specifies the position of total rows on the top or bottom of the aggregates.

Preview
Displays a preview of the selected layout.

Style screen

Specifies the style of the crosstab.



Style
Specifies the style of the crosstab.

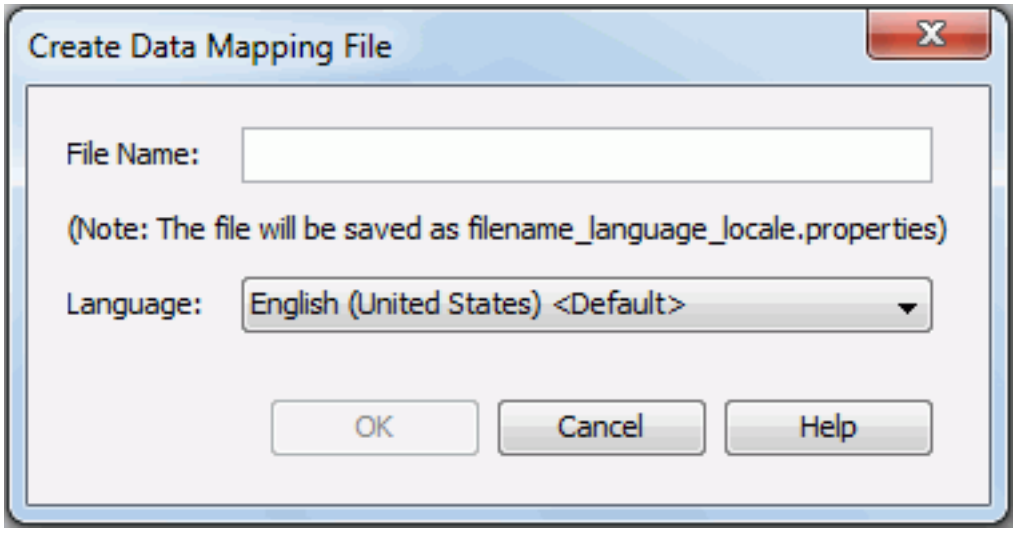
Preview
Shows a sketch of the selected style.

Related topics:

- Inserting a crosstab

Create Data Mapping File dialog

This dialog appears when you click the New button in the Data Mapping Editor dialog. It helps you to create a data mapping file that saves information of a certain language.



The following are details about options in the dialog:

File Name

Specifies the name of the data mapping file, which will be saved as filename_language_locale.properties.

Language

Specifies the language of the data mapping file. One language corresponds to one data mapping file.

OK

Applies the changes and closes the dialog.

Cancel

Cancels the changes and exits the dialog.

Help

Displays the help document about this feature.

Related topics:

- Mapping data values to different languages

Create Geographic Map wizard for library component

This wizard appears when you click Insert > Map > Geographic Map on the menu bar, click the Insert Geographic Map button on the insert toolbar, or drag Geographic Map from the Toolbox panel into a library component. It helps you to insert a geographic map into a library component, and consists of the following screens:

- Data screen
- Display screen
- Filter screen

Back
Goes back to the previous screen.

Next
Goes to the next screen.

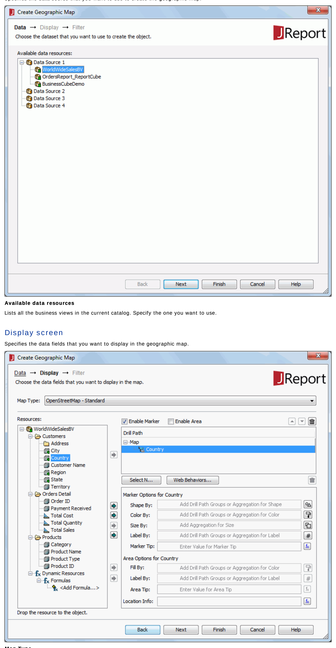
Finish
Finishes creating the geographic map and closes this dialog.

Cancel
Does not retain changes and closes this dialog.

Help
Displays the help document about this feature.

Data screen

Specifies the data source that you want to use to create the geographic map.



Available data resources
Lists all the business views in the current catalog. Specify the one you want to use.

Display screen

Specifies the data fields that you want to display in the geographic map.

Map Type
Specifies the type for the geographic map from the drop-down list.

- **OpenStreetMap - Standard**
  Displays the street map.
- **OpenCycleMap - Cycle Map**
  Displays the cycle route map.
- **OpenCycleMap - Transport Map**
  Displays the transport map.
- **Google Maps - Roadmap**
  Displays Google road map.
- **Google Maps - Satellite**
  Displays Google Earth satellite images.
- **Google Maps - Terrain**
  Displays a physical map based on terrain information.

Resources
Lists all the available data resources. You can also create dynamic formulas to use in the map.

Adds the selected field to the geographic map.

Enable Marker
Enables specifying the marker properties for all the groups.

Enable Area
Enables specifying the area properties for all the groups.

One option between Enable Marker and Enable Area must be checked at least and Enable Marker is checked by default.

Drill Path
Lists the fields that are used to group data in the geographic map.

Moves the specified group one step up.

Moves the specified group one step down.

Removes the selected fields that are not required from the geographic map.

Select N
Opens the Select N dialog to specify the Select N condition for the selected group level.

Web Behaviors
Opens the Web Behaviors dialog to specify some web behaviors to the markers and areas of the selected group level.

Marker Options
Specifies the marker properties of the selected group in the Drill Path box. Available only when Enable Marker is checked.

- **Shape By**
  Specifies the field used to control the shapes of the markers for the selected group level. Only the group field that represents the currently selected group level, or a group field of a higher level than the current level in the Drill Path box, or an aggregation field can be added.
  - Specifies the shapes of the markers.
  - If no field is added, select a shape from the shape list, or check and click the Advanced option to open the Advanced dialog to specify an image as the shape of the markers.
  - If a field is added, opens the Edit Shapes dialog to specify shapes for the markers.
- **Color By**
  Specifies the field used to control the colors of the markers for the selected group level. Only the group field that represents the currently selected group level, or group fields of higher levels than the current level in the Drill Path box, or an aggregation field can be added. When more than one group field is added, you can adjust the order of the fields by dragging and dropping them in the text box.
  - Specifies the colors of the markers.
  - If no field is added, select a color from the color palette or click **More Colors** on the color palette to open the Pick a Color dialog to specify a color for the markers.
  - If one or more group fields are added, opens the Edit Color dialog to specify colors for the markers.
  - If an aggregation field is added, opens the Edit Gradient Color dialog to specify a gradient color for the markers.
- **Size By**
  Specifies the field used to control the size of the markers for the selected group level. Only an aggregation field can be added.
  - Specifies the size of the markers.
  - **Zoom percentage bar**
    Specifies the zoom percentage based on the default size.
  - **Reset**
    Specifies whether to restore the default size.
- **Label By**
  Specifies the field to control the labels shown above the markers for the selected group level. When there are more than one field, you can adjust the order of the fields by dragging and dropping them in the text box.
  - Arial
    Specifies the font format of the label text.
  - 9
    Specifies the font size of the label text.
  - Specifies whether to make the label text bold.
  - Specifies whether to make the label text italic.
  - Specifies whether to underline the text.
  - Specifies the color of the label text.
  - Specifies the format for the field you currently select from the preview box below.
  - **Preview box**
    Displays the specified text format effects.
  - **OK**
    Applies the specified font format and closes the dialog.
- **Marker Tip**
  Specifies the tip information when you hover the mouse on the markers at runtime.

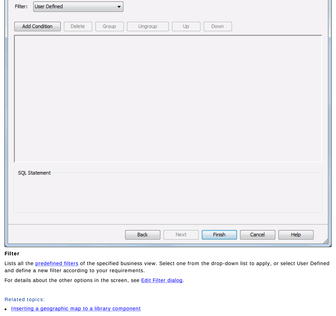
Area Options
Specifies the area properties of the selected group in the Drill Path box. Available only when Enable Area is checked.

- **Fill By**
  Specifies the field to control the colors that are used to fill the areas of the selected group level. Only the group field that represents the currently selected group level, or group fields of higher levels than the current level in the Drill Path box, or an aggregation field can be added. When more than one group field is added, you can adjust the order of the fields by dragging and dropping them in the text box.
  - Specifies the colors to fill the areas.
  - If no field is added, select a color from the color palette or click **More Colors** on the color palette to open the Pick a Color dialog to specify a color for the areas.
  - If one or more group fields are added, opens the Edit Color dialog to specify colors for the areas.
  - If an aggregation field is added, opens the Edit Gradient Color dialog to specify a gradient color for the areas.
- **Label By**
  Specifies the field to control the labels shown above the areas for the selected group level. When more than one field is added, you can adjust the order of the fields by dragging and dropping.
  - Arial
    Specifies the font format of the label text.
  - 9
    Specifies the font size of the label text.
  - Specifies whether to make the label text bold.
  - Specifies whether to make the label text italic.
  - Specifies whether to underline the text.
  - Specifies the color of the label text.
  - Specifies the format for the field you currently selected from the preview box below.
  - **Preview box**
    Displays the specified text format effects.
  - **OK**
    Applies the specified font format and closes the dialog.
- **Area Tip**
  Specifies the tip information when you hover the mouse on the areas at runtime.

Location Info
Specifies the mapping location of the markers or areas for the selected group by field in the Drill Path box. For each group level that you want to show on the map, specify a DBfield or formula that contains the value matching the folder name in <install_root\qlikfo\geocode used by the geographic map. For example you may group by Country and State. For State, you would want to use a formula fStateCountry that concatenates state and country so Google Maps, OpenStreetMap or OpenCycleMap knows which state to use such as "New York, USA" or "California, USA".

Filter screen

Specifies to filter the data used in the geographic map.



Filter
Lists all the predefined filters of the specified business view. Select one from the drop-down list to apply, or select User Defined and define a new filter according to your requirements.

For details about the other options in the screen, see Edit Filter dialog.

Related topics:

- Inserting a geographic map to a library component

Create Geographic Map wizard for page report

This wizard appears when you click Insert > Map > Geographic Map on the menu bar, click the Insert Geographic Map button on the Insert toolbar, or drag Geographic Map from the Toolbox panel into a page report. It helps you to insert a geographic map into a page report, and consists of the following screens:

- Data screen
- Group screen
- Filter screen
- Layout screen
- Marker screen

Back
Goes back to the previous screen.

Next
Goes to the next screen.

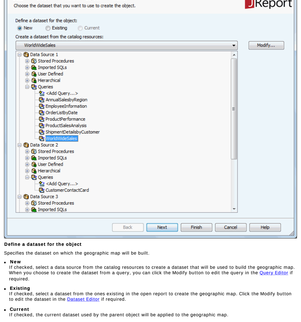
Finish
Finishes creating the geographic map and closes this dialog.

Cancel
Does not retain changes and closes this dialog.

Help
Displays the help document about this feature.

Data screen

Specifies the dataset that you want to use to create the geographic map.

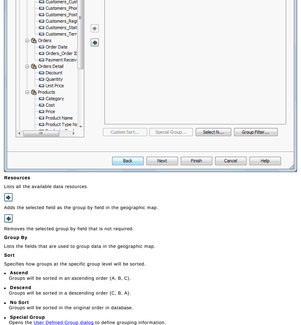


Define a dataset for the object
Specifies the dataset on which the geographic map will be built.

- **New**
  If checked, select a data source from the catalog resources to create a dataset that will be used to build the geographic map. When you choose to create the dataset from a query, you can click the Modify button to edit the query in the Query Editor if required.
- **Existing**
  If checked, select a dataset from the ones existing in the open report to create the geographic map. Click the Modify button to edit the dataset in the Dataset Editor if required.
- **Current**
  If checked, the current dataset used by the parent object will be applied to the geographic map.

Group screen

Specifies the data fields that are used to group the data in the geographic map.



Resources
Lists all the available data resources.

Adds the selected field as the group by field in the geographic map.

Removes the selected group by field that is not required.

Group By
Lists the fields that are used to group data in the geographic map.

Sort
Specifies how groups at the specific group level will be sorted.

- **Asceld**
  Groups will be sorted in an ascending order (A, B, C).
- **Descend**
  Groups will be sorted in a descending order (C, B, A).
- **No Sort**
  Groups will be sorted in the original order in database.
- **Special Group**
  Opens the User Defined Group dialog to define grouping information.

  For example, if you placed a field named Region for grouping, and this field contains all 50 states of the United States, and if you want to see the data between Maryland (MD) and New York (NY), you can define the criteria by selecting the between operator to further define your grouping information.
- **Custom Sort**
  Opens the Custom Sort dialog to set how groups will be sorted.

Special Function
If the group by field is of Numeric/String/Date/Time type, you can select a special function for the field in the Special Function column to further specify to which level the data will be grouped by.

If Customsrc is selected, the Customized Function dialog will be displayed, in which you can set the function by your own.

Custom Sort
Specifies how to sort the groups. Activated only when you have selected Custom Sort from the Sort column to define the sorting manner of groups for the selected group level.

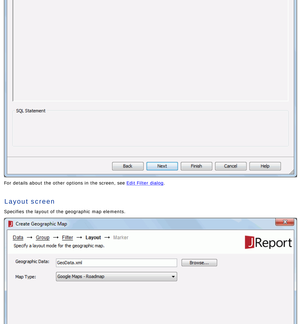
Special Group
Specifies how to group your information. Activated only when you have selected Special Group from the Sort column to define a special group.

Select N
Opens the Select N dialog to specify the Select N condition.

Group Filter
Opens the Group Filter dialog to specify the group filter condition.

Filter screen

Specifies to filter the data used in the geographic map.



For details about the other options in the screen, see Edit Filter dialog.

Layout screen

Specifies the layout of the geographic map elements.

Geographic Data
Specifies the geographical data for the geographic map with an XML file. If the XML file that you specify does not exist it will be created in the <installdir>\Crystalreports\Geo\ folder. The geographic data XML file is used to map the geographic latitude and longitude information in Google Maps, OpenStreetMap or OpenCycleMap based on the location information specified for each row in the database. In the XML file, the location information in a group value matches the "nm" field in the XML file, you will get the related latitude and longitude value for this group value.

You can also import an XML file which contains predefined geographic data including latitude and longitude for the specific "nm" field. By default, the XML file is located in the <installdir>\Crystalreports\Geo\ directory. If not, the file will be copied to this directory after it is imported. When you publish the report from Designer you can specify the gisinfo files to publish to the server from the Geographic Information tab.

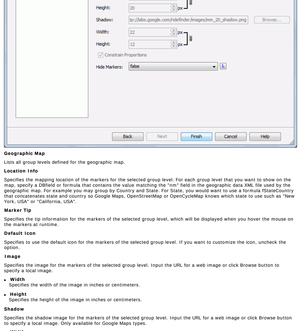
If you do not publish the geographical information it will be created automatically when you first run the report.

Map Type
Specifies the type for the geographic map from the drop-down list.

- **Google Maps - Roadmap**
  Displays Google road map.
- **Google Maps - Satellite**
  Displays Google Earth satellite images.
- **Google Maps - Terrain**
  Displays a topological map based on terrain information.
- **OpenStreetMap - Standard**
  Displays the street map.
- **OpenCycleMap - Cycle Map**
  Displays the cycle route map.
- **OpenCycleMap - Transport Map**
  Displays the transport map.

Marker screen

Specifies properties of the markers bound with the group levels.



Geographic Map
Lists all group levels defined for the geographic map.

Location Info
Specifies the mapping location of the markers for the selected group level. For each group level that you want to show on the map, select a field in the dataset/formula that contains the value matching the "nm" field in the geographic data XML file used by the geographic map. For example, you may group by Country and State. For State, you would want to use a formula fSetCountry("State", "Country") to set the state and country so Google Maps, OpenStreetMap or OpenCycleMap knows which state to use such as "New York, USA" or "California, USA".

Marker Tip
Specifies the tip information for the markers of the selected group level, which will be displayed when you hover the mouse on the markers at runtime.

Default Icon
Specifies to use the default icon for the markers of the selected group level. If you want to customize the icon, uncheck the option.

Image
Specifies the image for the markers of the selected group level. Input the URL for a web image or click Browse button to specify a local image.

- **Width**
  Specifies the width of the image in inches or centimeters.
- **Height**
  Specifies the height of the image in inches or centimeters.

Shadow
Specifies the shadow image for the markers of the selected group level. Input the URL for a web image or click Browse button to specify a local image. Only available for Google Maps types.

- **Width**
  Specifies the width of the shadow image in inches or centimeters.
- **Height**
  Specifies the height of the shadow image in inches or centimeters.

Constrain Proportions
Specifies whether to change the width and height for the marker image/shadow at the same time in a certain proportion. If checked, when the width/height is changed, the height/width will also be changed in a certain proportion.

Hide Markers
Specifies whether to show or hide the markers of the selected group level.

Related topics:

- Inserting a geographic map to a page report

Create Geographic Map wizard for web report

This wizard appears when you click Insert > Map > Geographic Map on the menu bar, click the Insert Geographic Map button on the Insert toolbar, or drag Geographic Map from the Toolbox panel into a web report. It helps you to insert a geographic map into a web report, and consists of the following screens:

- Data screen
- Display screen
- Filter screen

Back
Goes back to the previous screen.

Next
Goes to the next screen.

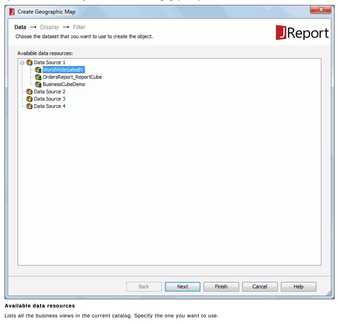
Finish
Finishes creating the geographic map and closes this dialog.

Cancel
Does not retain changes and closes this dialog.

Help
Displays the help document about this feature.

Data screen

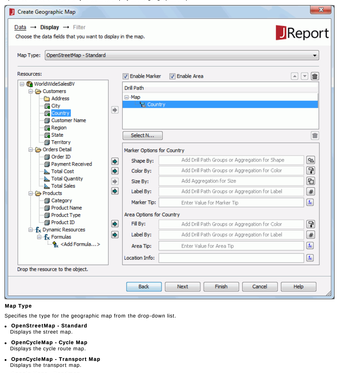
Specifies the data source that you want to use to create the geographic map.



Available data resources
Lists all the business views in the current catalog. Specify the one you want to use.

Display screen

Specifies the data fields that you want to display in the geographic map.



Map Type
Specifies the type for the geographic map from the drop-down list.

- **OpenStreetMap - Standard**
  Displays the street map.
- **OpenCycleMap - Cycle Map**
  Displays the cycle route map.
- **OpenCycleMap - Transport Map**
  Displays the transport map.
- **Google Maps - Roadmap**
  Displays Google road map.
- **Google Maps - Satellite**
  Displays Google Earth satellite images.
- **Google Maps - Terrain**
  Displays a physical map based on terrain information.

Resources
Lists all the available data resources. You can also create dynamic formulas to use in the map.

Adds the selected field to the geographic map.

Enable Marker
Enables specifying the marker properties for all the groups.

Enable Area
Enables specifying the area properties for all the groups.

One option between Enable Marker and Enable Area must be checked at least and Enable Marker is checked by default.

Drill Path
Lists the fields that are used to group data in the geographic map.

Moves the specified group one step up.

Moves the specified group one step down.

Removes the selected fields that are not required from the geographic map.

Select N
Opens the Select N dialog to specify the Select N condition for the selected group level.

Marker Options
Specifies the marker properties of the selected group in the Drill Path box. Available only when Enable Marker is checked.

- **Shape By**
  Specifies the field used to control the shapes of the markers for the selected group level. Only the group field that represents the currently selected group level, or a group field of a higher level than the current level in the Drill Path box, or an aggregation field can be added.
  - Specifies the shapes of the markers.
  - If no field is added, select a shape from the shape list, or check and click the Advanced option to open the Advanced dialog to specify an image as the shape of the markers.
  - If a field is added, opens the Edit Shapes dialog to specify shapes for the markers.
- **Color By**
  Specifies the field used to control the colors of the markers for the selected group level. Only the group field that represents the currently selected group level, or group fields of higher levels than the current level in the Drill Path box, or an aggregation field can be added. When more than one group field is added, you can adjust the order of the fields by dragging and dropping them in the text box.
  - Specifies the colors of the markers.
  - If no field is added, select a color from the color palette or click **More Colors** on the color palette to open the Pick a Color dialog to specify a color for the markers.
  - If one or more group fields are added, opens the Edit Color dialog to specify colors for the markers.
  - If an aggregation field is added, opens the Edit Gradient Color dialog to specify a gradient color for the markers.
- **Size By**
  Specifies the field used to control the size of the markers for the selected group level. Only an aggregation field can be added.
  - Specifies the size of the markers.
  - **Zoom percentage bar**
    Specifies the zoom percentage based on the default size.
  - **Reset**
    Resets the marker to restore the default size.
- **Label By**
  Specifies the field to control the labels shown above the markers for the selected group level. When there are more than one field, you can adjust the order of the fields by dragging and dropping them in the text box.
  - Specifies the font format of the label text.
    - Arial
      Specifies the font of the label text.
    - 9
      Specifies the font size of the label text.
    - Specifies whether to make the label text bold.
    - Specifies whether to make the label text italic.
    - Specifies whether to underline the text.
    - Specifies the color of the label text.
    - $
      Specifies the format for the field you currently select from the preview box below.
    - **Preview box**
      Displays the specified font format effects.
    - **OK**
      Applies the specified font format and closes the dialog.
- **Marker Tip**
  Specifies the tip information when you hover the mouse on the markers at runtime.

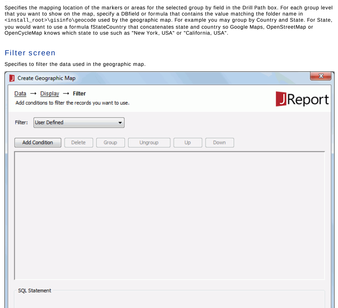
Area Options
Specifies the area properties of the selected group in the Drill Path box. Available only when Enable Area is checked.

- **Fill By**
  Specifies the field to control the colors that are used to fill the areas of the selected group level. Only the group field that represents the currently selected group level, or group fields of higher levels than the current level in the Drill Path box, or an aggregation field can be added. When more than one group field is added, you can adjust the order of the fields by dragging and dropping them in the text box.
  - Specifies the colors to fill the areas.
  - If no field is added, select a color from the color palette or click **More Colors** on the color palette to open the Pick a Color dialog to specify a color for the areas.
  - If one or more group fields are added, opens the Edit Color dialog to specify colors for the areas.
  - If an aggregation field is added, opens the Edit Gradient Color dialog to specify a gradient color for the areas.
- **Label By**
  Specifies the field to control the labels shown above the areas for the selected group level. When more than one field is added, you can adjust the order of the fields by dragging and dropping.
  - Specifies the font format of the label text.
    - Arial
      Specifies the font of the label text.
    - 9
      Specifies the font size of the label text.
    - Specifies whether to make the label text bold.
    - Specifies whether to make the label text italic.
    - Specifies whether to underline the text.
    - Specifies the color of the label text.
    - $
      Specifies the format for the field you currently selected from the preview box below.
    - **Preview box**
      Displays the specified font format effects.
    - **OK**
      Applies the specified font format and closes the dialog.
- **Area Tip**
  Specifies the tip information when you hover the mouse on the areas at runtime.

Location Info
Specifies the mapping location of the markers or areas for the selected group by field in the Drill Path box. For each group level that you want to show on the map, specify a DBfield or formula that contains the value matching the folder name in `<install_root>\jsinfo\geocode` used by the geographic map. For example you may group by Country and State. For State, you would want to use a formula fStatsCountry that concatenates state and country so Google Maps, OpenStreetMap or OpenCycleMap knows which state to use such as "New York, USA" or "California, USA".

Filter screen

Specifies to filter the data used in the geographic map.



Filter
Lists all the predefined filters of the specified business view. Select one from the drop-down list to apply, or select User Defined and define a new filter according to your requirements.

For details about the other options in the screen, see Edit Filter dialog.

Related topics:

- Inserting a geographic map to a web report

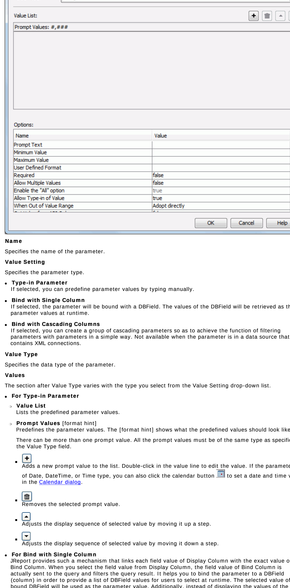
Create Parameter dialog

This dialog appears when you do any of the following:

- Right-click the Parameters node in the Catalog Browser and select Add Parameter from the shortcut menu.
- In the Resource View panel of the main window, click <Add Parameter>; in the Local Parameters node.
- Right-click a data component in a web report or library component and select Format Filter from the shortcut menu. In the Edit Filter dialog, click next to the value text field, then in the Values dialog, click <Add Parameter>; in the Local Parameters node.

It helps you to add parameters to a catalog, or create local parameters to use in the current web report or library component.

The following are details about options in the dialog.



Name

Specifies the name of the parameter.

Value Setting

Specifies the parameter type.

- **Type-in Parameter**
  If selected, you can predefine parameter values by typing manually.
- **Bind with Single Column**
  If selected, the parameter will be bound with a DBField. The values of the DBField will be retrieved as the parameter values at runtime.
- **Bind with Cascading Columns**
  If selected, you can select a group of cascading parameters so as to achieve the function of filtering parameters with parameters in a simple way. Not available when the parameter is in a data source that contains XML connections.

Value Type

Specifies the data type of the parameter.

Values

The section after Value Type varies with the type you select from the Value Setting drop-down list.

- **For Type-in Parameter**
  - **Value List**
    Lists the predefined parameter values.
    - **Prompt Values** [format hint]
      Predefines the parameter values. The [format hint] shows what the predefined values should look like.

    There can be more than one prompt value. All the prompt values must be of the same type as specified in the Value Type field.
    - Adds a new prompt value to the list. Double-click in the value line to edit the value. If the parameter is of Date, DateTime, or Time type, you can also click the calendar button to set a date and time value in the Calendar dialog.
    - Removes the selected prompt value.
    - Adjusts the display sequence of selected value by moving it up a step.
    - Adjusts the display sequence of selected value by moving it down a step.
- **For Bind with Single Column**
  JReport provides such a mechanism that links each field value of Display Column with the exact value of Bind Column. When you select the field value from Display Column, the field value of Bind Column is actually sent to the query and filters the query result. It helps you to bind the parameter to a DBField (column) in order to provide a list of DBField values for users to select at runtime. The selected value of the bound DBField will be used as the parameter value. Additionally, instead of displaying the values of the Display Column, you can also choose to display values of other DBfield, which probably makes more sense to the report end users. For example, it might be confusing if you provide a list of customer ID numbers for users to select at runtime since the ID numbers could probably mean nothing to them. In cases like this, it is wise for you to display values of other fields, which could probably make more sense. For this case, you might prefer to display customer names instead of ID numbers. When the report end users select a customer name from the list, its ID number is passed to the query as the parameter value so that the parameter value can be fulfilled. So, before the report with a parameter bound to a DBfield is processed, a query statement will be executed first to retrieve the values of the bound field and display field.
  - **Source**
    Specifies the source type from which to retrieve the columns which can be used to bind with the parameter. The available source types are: Tables and Views (including synonyms), Stored Procedures, Imported SQLs, and User Defined. However, if the catalog data source in which the parameter is contains only a MongoDB connection, only Tables and User Defined are available in the Source drop-down list.
  - **Bind Column**
    Specifies the DBField to be bound with the parameter.
  - **Display Column**
    When viewing a report which contains the parameter, the field values related to the column name you selected in the drop-down list of Display Column will be displayed in the Enter Parameter Values dialog at runtime.
- **For Bind with Cascading Columns**
  - **Data Source**
    Specifies the data source from which to retrieve the columns which can be used to bind with the parameters. The available data sources are tables, views, synonyms, stored procedures, imported SQLs, and user defined data sources in the catalog data source in which the parameter is. However, if the catalog data source contains only a MongoDB connection, only tables and user defined data sources are available.
  - **Value List**
    Defines a group of cascading parameters.
    - Adds a new parameter.
    - Removes the selected parameter.
    - Adjusts the parameter level by moving it up a step. The higher the position is, the higher the level is. The lower level field values are controlled by the higher level field values.
    - Adjusts the parameter level by moving it down a step.
  - **Bind Column**
    Specifies the DBField to be bound with the parameter.
  - **Display Column**
    When viewing a report which contains the parameter, the field values related to the column name you selected in the drop-down list of Display Column will be displayed in the Enter Parameter Values dialog at runtime.
- **Parameter**
  Specifies whether to set a group of Bind Column and Display Column as a parameter which will become a member of the cascading parameter group. All parameters in the cascading group will be created as independent parameters in the catalog data source in which the parameter is.

  The parameter name is provided by JReport automatically, and the format is *[The name specified as parameter name]-[Bind Column name]*.

Options

Specifies options for the parameter. The options vary with the type you select from the Value Setting drop-down list.

- **Prompt Text**
  Specifies the text for prompting users to provide the value of the parameter.

  Tips
  - To help users avoid typing a value out of range, you can add the value range to the end of the prompt text. For example, "Type a number (0-10)".
  - To help users avoid typing a value in a wrong pattern, you can add the value pattern to the end of the prompt text. For example, "Type an order date (MMM-dd-yy)".
- **Minimum Value/Length**
  Specifies the minimum value allowed for the parameter. If the parameter value type is String, this option is defined as Minimum Length, which means that you can set the minimum allowed length of the string.
- **Maximum Value/Length**
  Specifies the maximum value allowed for the parameter. If the parameter value type is String, this option is defined as Maximum Length, which means that you can set the maximum allowed length of the string.
- **String Encoding**
  Specifies the encoding for the parameter. Available only when the parameter's data type is defined as String.
- **User Defined Format**
  For certain types of parameters, the value format supplied by JReport may not satisfy your requirement. In this case, you can define your own preferred format. When you have defined the value format, all the values you supply at runtime should be based on it. Otherwise there will be an error message. For details, see User defined format for parameters.

  Notes:
  - The letters that stand for the user defined format should follow the JDK standard. In addition, the user defined format only limits the format of the input value, which does not mean that the display value will be in the same format. To change the display format, you will need to specify the Format property of this parameter in the Report Inspector.
  - The user defined format will show different appearance under different locale, and you need input values according to the displayed format. For example, under the Spanish or French locale, #,###.## will be displayed as #0,## and you should follow #0,## to input values.
- **Required**
  If set to true, the parameter will be treated as a required parameter. A required parameter must be fulfilled, which means that you must supply a value in the Enter Parameter Values dialog before the report can be run. If set to false, the parameter will be treated as an optional parameter. An optional parameter can be left blank, which means you can either type a value, or leave it as it is.

  In the Enter Parameters Value dialog, the prompt text of a required parameter will be marked with a * sign.

  Note: If the parameter is optional and you do not expect it to take effect in your report, in the Enter Parameter Values dialog, clear the content in the value field. Otherwise, any value left in the field will function in reports.
- **Hide Parameter When Single Value Returned**
  Specifies to hide the parameter at runtime when only a single value is returned for the parameter. The returned single value will be used as default value of the parameter.
- **Allow Multiple Values**
  Enables multiple parameter values to be selected. If it is set to true, when running a report with the parameter, the Enter Values dialog is provided for you to specify multiple values.

  Note: Multi-value parameters can only be supported in dataset filters, query filters, imported SQL files, and parameterized report conditions.
- **Enable the "All" Option**
  Displays the All checkbox in the Enter Values dialog which means that the parameter value result is all the values that match the parameter data type. Available when Allow Multiple Values is set to true.
- **Treat Blank as Null**
  Available only when the parameter data type is defined as String. If it is true, the parameter value will be null; if false, the default parameter value is "".
- **Use Current Date-Time**
  If set to true, the current date time in your system will be used as value of the parameter. Available only when the parameter's data type is defined as Date, Time or DateTime.

  If the type of the parameter is set to Date, Time or DateTime, you can choose whether to get parameter value dynamically from the system's date time by enabling one of the above options - Use Current or Use Current When Blank. However, if you want to use this feature, you need to make sure your system's DateTime format is set to yyyy-MM-dd H:mm:ss as it is the only format JReport supports; otherwise, you need to make use of the User Defined Parameter Format feature. Also, if one of the two options is enabled, you need to provide the parameter's prompt value according to the data type of your parameter.
  - When the data type of the parameter is Date, you can specify the value according to the format yyyy-MM-dd.
    - yyyy-01-01: Year will be dynamically changed.
    - 2007-MM-01: Month will be dynamically changed.
    - 2007-01-dd: Date will be dynamically changed.
    - yyyy-MM-01: Year and month will be dynamically changed.
    - yyyy-01-dd: Year and date will be dynamically changed.
    - yyyy-MM-dd: Year, month, and date will be dynamically changed.
    - 2007-MM-dd: Month and date will be dynamically changed.
  - When the data type of the parameter is Time, you can specify the value according to the format H:mm:ss.
    - H:00:00: Hour will be dynamically changed.
    - 12:mm:00: Minute will be dynamically changed.
    - 12:00:ss: Second will be dynamically changed.
    - H:mm:00: Hour and minute will be dynamically changed.
    - 12:mm:ss: Minute and second will be dynamically changed.
    - H:00:ss: Hour and second will be dynamically changed.
    - H:mm:ss: Hour, minute and second will be dynamically changed.
  - When the data type of the parameter is DateTime, specify the value in a combination of the above.
- **Record Level Security**
  Specifies the RLS policy defined on the bound column in the data source to the parameter. For details, refer to Applying RLS to a parameter.
- **Sort**
  Specifies how to sort values of cascading parameters.
  - **Ascend**
    Sorts parameter values in an ascending order.
  - **Descend**
    Sorts parameter values in a descending order.
  - **No Sort**
    The parameter values are displayed in the original order as in database.
- **Sort By**
  Sorts parameter values by sorting other fields in the same data source as the cascading group. When Sort By is selected, the Sort By dialog will be displayed for specifying fields and their sorting orders.
- **Distinct**
  If set to true, when viewing a report that contains the parameter, in the Enter Parameter Values dialog, identical values will appear only once and no duplicate values will be listed.
- **Import SQL**
  Displays the SQL statement for retrieving values of the bound column and display column. You can edit the statement if required. To do this, click in the value cell, then click and edit the statement in the Import SQL dialog.

  Notes:
  - If you have made changes to the SQL statement, the later changes for Bind Column and Display Column will not take effect unless you remove the SQL lines or bind the parameter all over again. To remove the SQL lines, click in the value cell and clear the content in the Import SQL dialog.
  - The special field User Name can also be used in the SQL statement as *@username*.
- **Allow Type-in of Value**
  Specifies whether to enable typing values besides selecting values when setting parameter values. The properties enabled for Type-in Parameter by default.
- **When Out of Value Range**
  Specifies how JReport will respond if the typed value is out of the predefined value list. The property is enabled only when the option Allow Type-in of Value is set to true.

  There are three solutions as follows:
  - **Adopt directly**
    The typed value will be used directly.
  - **Use default**
    The default value defined when creating the parameter will be used.
  - **Warn**
    Displays a warning message telling that "Value xxx is out of the available value list".

    Note: For a multi-value parameter, if one of the input values is not in the value list, the whole input result will be treated as out of range.
- **Get Value from API Only**
  Specifies whether the parameter values can be input via GUI or not.

  If true, the parameter values should only be obtained from API and there will be no parameter dialogs available on any JReport Designer or Server GUI for inputting values. Users will have to give the values of the parameter by API such as URLs, coding, and sessions, otherwise the default value will be used.

  If false, the parameter can be obtained from both GUI and API.

  If a member of a cascading parameter group is set to get values from API only, it means that all the higher level members shall get values from API only. This also applies to parameters that some other parameters depend on. If in JDashboard several parameters share values, when any of them is set to get values from API only, they will share the values from API.
- **On Parameter Value Change**
  Specifies to define an action that will be called when the parameter value changes. Choose a formula from the drop-down list to develop an action, which will be called before the other actions defined in the report or library component. It can work as a parameter value validation rule.

OK

Applies all changes and closes the dialog.

Cancel

Does not retain any changes and closes the dialog.

Help

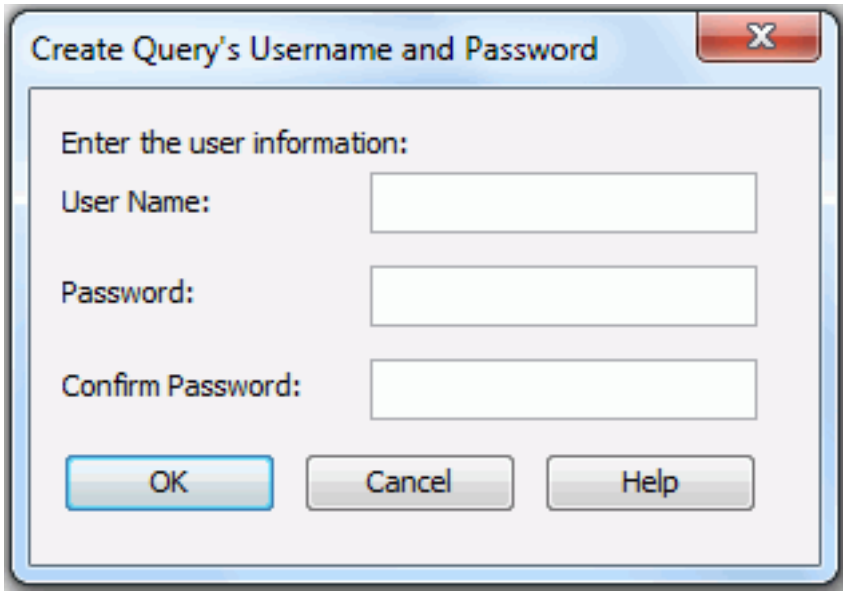
Displays the help document about this feature.

Related topics:

- Creating a parameter

Create Query's Username and Password dialog

This dialog appears when you right-click a query and then select Create User from the shortcut menu. It helps you to set a password to restrict users from editing the query at random.



The following are details about options in the dialog:

User Name

Specifies the user name that can edit the query.

Password

Specifies the password to restrict users from editing the query.

Confirm password

Confirms the password you have specified in the Password text field.

OK

Applies all changes and closes the dialog.

Cancel

Cancels the changes and exits the dialog.

Help

Displays the help document about this feature.

Related topics:

- Locking queries

Create Table wizard for page report

This wizard appears when you choose a table type and click OK in the Table Type dialog. It helps you to create a table in a page report, and consists of the following screens:

- Data screen
- Display screen
- Group screen
- Summary screen
- Filter screen
- Style screen

Back
Goes back to the previous screen.

Next
Goes to the next screen.

Finish
Finishes creating the table and closes this dialog.

Cancel
Does not retain changes and closes this dialog.

Help
Displays the help document about this feature.

Data screen

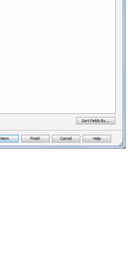
Specifies the dataset that you want to use to create the table.

Define a dataset for the object
Specifies the dataset on which the table will be built.

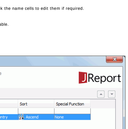
- **New**
If checked, select a data source from the catalog resources to create a dataset that will be used to build the table. When you choose to create the dataset from a query, you can click the Modify button to edit the query in the Query Editor if required.
- **Existing**
If checked, select a dataset from the ones existing in the open report to create the table. Click the Modify button to edit the dataset in the Dataset Editor if required.
- **Current**
If checked, the current dataset used by the parent object will be applied to the table.

Display screen

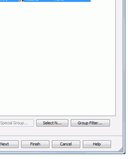
Specifies the fields that you want to display in the table.



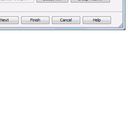
Resources
Lists all the available data resources.



Adds the specified field to use in the table.



Removes the specified field that is not required in the table.



Moves the specified field one step up.

Moves the specified field one step down.

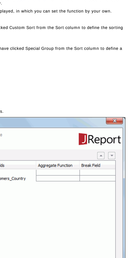
Display Fields
Lists the fields that have been selected to display in the table.

Display Name
Shows the display names of the selected fields. You can double-click the name cells to edit them if required.

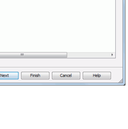
Sort Fields By
Opens the Sort Fields By dialog to specify how to sort data in the table.

Group screen

Specifies the fields that you want to use to group the data.



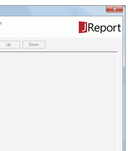
Resources
Lists all the available data resources.



Adds the selected field as the group by field in the table.

Removes the selected group by field that is not required.

Moves the specified group one step up.



Moves the specified group one step down.

Group By
Lists the fields that are used to group data in the table.

Sort
Specifies how groups at the specific group level will be sorted.

- **Ascend**
Groups will be sorted in an ascending order (A, B, C).
- **Descend**
Groups will be sorted in a descending order (C, B, A).
- **No Sort**
Groups will be sorted in the original order in database.
- **Special Group**
Opens the User Defined Group dialog to define grouping information.

For example, if you placed a field named Region for grouping, and the field contains all 50 states of the United States; and if you want to see the data between Maryland (MD) and New York (NY), you can define the criteria by selecting the between operator to further define your grouping information.

- **Custom Sort**
Opens the Custom Sort dialog to set how groups will be sorted.

Special Function
If the group by field is of Numeric/String/Date/Time type, you can select a special function for the field in the Special Function column to further specify to which level the data will be grouped by.
If Customers is selected, the Customized Function dialog will be displayed, in which you can set the function by your own.

Custom Sort
Opens now to sort the groups. Activated only when you have clicked Custom Sort from the Sort column to define the sorting manner of groups for the selected group level.

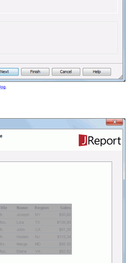
Special Group
Specifies how to group your information. Activated only when you have clicked Special Group from the Sort column to define a special group.

Select N
Opens the Select N dialog to specify the Select N condition.

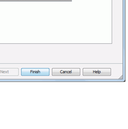
Group Filter
Opens the Group Filter dialog to specify the group filter condition.

Summary screen

Specifies the fields on which you want to create aggregate functions.



Resources
Lists all the available data resources.



Adds the selected field as the summary field to the table.

Removes the selected summary field that is not required.

Moves the specified summary field one step up.

Moves the specified summary field one step down.

Summarized Fields
Lists all the fields that you want to display and to be summarized in the table.

Aggregate Function
Specifies the function to summarize the selected field.

Break Field
Displays a defined field on which the summary will be calculated. If a summary field is added below the Table node, the break field is null and the summary will be calculated on the whole dataset.

Position
Works together with the Column option to specify the location where the summary field will be put.

- **Header**
Specifies to put the summary field in the header row. If the summary is calculated on a group by field, it will be put in the group header row of the corresponding group; if the summary is calculated on the whole dataset, it will be put in the table header row.
- **Footer**
Specifies to put the summary field in the footer row. If the summary is calculated on a group by field, it will be put in the group footer row of the corresponding group; if the summary is calculated on the whole dataset, it will be put in the table footer row.

Column
Works together with the Position option to specify the location where the summary field will be put.

- **Detail**
The summary field will be put in a detail column.
- **Summary**
The summary field will be displayed in a separate summary column.

Filter screen

Specifies to filter data displayed in the table.

For details about the other options in the screen, see Edit Filter dialog.

Style screen

Specifies the style of the table.

Group Report
Specifies the layout of the table.

- **Vertically**
Creates a vertical table.
- **Horizontally**
Creates a horizontal table.

Style
Specifies the style of the table.

Preview
Displays the selected layout and style effects.

Inherit Style
Specifies whether to make the table take the style of its parent. Available only when the table is to be inserted into a banded object.

Related topics:

- Inserting a table

Create Table wizard for web report/library component

This wizard appears when you choose a table type and click OK in the Table Type dialog. It helps you to create a table in a web report or library component, and consists of the following screens:

- Data screen
- Display screen
- Group screen
- Summary screen
- Filter screen
- Style screen

Back
Goes back to the previous screen.

Next
Goes to the next screen.

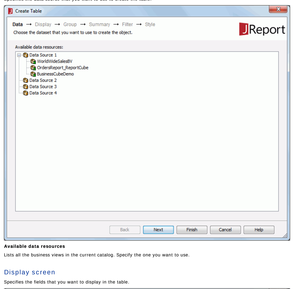
Finish
Finishes creating the table and closes this dialog.

Cancel
Does not retain changes and closes this dialog.

Help
Displays the help document about this feature.

Data screen

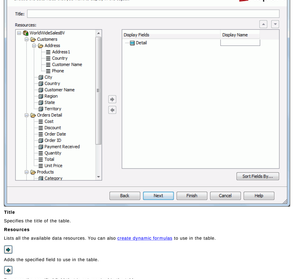
Specifies the data source that you want to use to create the table.



Available data resources
Lists all the business views in the current catalog. Specify the one you want to use.

Display screen

Specifies the fields that you want to display in the table.



Title
Specifies the title of the table.

Resources
Lists all the available data resources. You can also create dynamic formulas to use in the table.

Adds the specified field to use in the table.

Removes the specified field that is not required in the table.

Moves the specified field one step up.

Moves the specified field one step down.

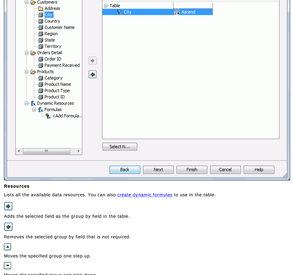
Display Fields
Lists the fields that have been selected to display in the table.

Display Name
Shows the display names of the selected fields. You can double-click the name cells to edit them if required.

Sort Fields By
Opens the Sort Fields By dialog to specify how to sort data in the table.

Group screen

Specifies the fields that you want to use to group the data.



Resources
Lists all the available data resources. You can also create dynamic formulas to use in the table.

Adds the selected field as the group by field in the table.

Removes the selected group by field that is not required.

Moves the specified group one step up.

Moves the specified group one step down.

Group By
Lists the fields that are used to group data in the table.

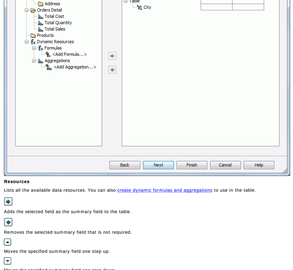
Sort
Specifies how groups at the specific group level will be sorted.

- **Ascend**
  Groups will be sorted in an ascending order (A, B, C).
- **Descend**
  Groups will be sorted in a descending order (C, B, A).
- **No Sort**
  Groups will be sorted in the original order in database.
- **Custom Sort**
  Opens the Custom Sort dialog to set how groups will be sorted.

Select N
Opens the Select N dialog to specify the Select N condition.

Summary screen

Specifies the fields on which you want to create aggregate functions.



Resources
Lists all the available data resources. You can also create dynamic formulas and aggregations to use in the table.

Adds the selected field as the summary field to the table.

Removes the selected summary field that is not required.

Moves the specified summary field one step up.

Moves the specified summary field one step down.

Summarized Fields
Lists all the fields that you want to display and to be summarized in the table.

Position
Works together with the Column option to specify the location where the summary field will be put.

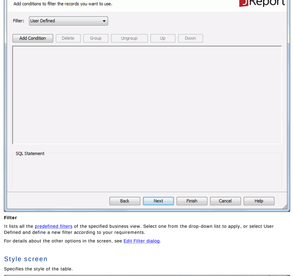
- **Header**
  Specifies to put the summary field in the header row. If the summary is calculated on a group by field, it will be put in the group header row of the corresponding group; if the summary is calculated on the whole dataset, it will be put in the table header row.
- **Footer**
  Specifies to put the summary field in the footer row. If the summary is calculated on a group by field, it will be put in the group footer row of the corresponding group; if the summary is calculated on the whole dataset, it will be put in the table footer row.

Column
Works together with the Position option to specify the location where the summary field will be put.

- **Detail**
  The summary field will be put in a detail column.
- **Summary**
  The summary field will be displayed in a separate summary column.

Filter screen

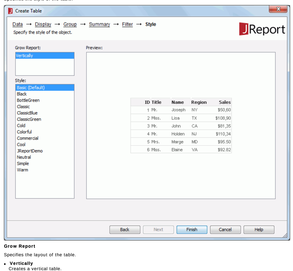
Specifies to filter data displayed in the table.



Filter
It lists all the predefined filters of the specified business view. Select one from the drop-down list to apply, or select User Defined and define a new filter according to your requirements.
For details about the other options in the screen, see Edit Filter dialog.

Style screen

Specifies the style of the table.



Grow Report
Specifies the layout of the table.

- **Vertically**
  Creates a vertical table.

Style
Specifies the style of the table.

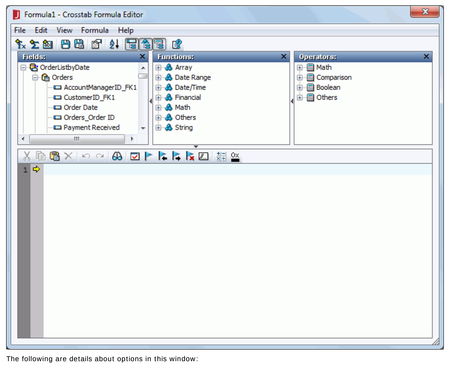
Preview
Displays the selected layout and style effects.

Related topics:

- Inserting a table

Crosstab Formula Editor window

This window appears when you select a crosstab component, on the Resource View panel click <Add Crosstab Formula...> under the Crosstab Formulas node, input a name for the new crosstab formula and then click OK, or right-click a crosstab formula and click Edit Formula on its shortcut menu. It helps you to create custom aggregate functions in a crosstab.



The following are details about options in this window:

File menu

- **New Formula**
  Creates a new formula.
- **New Summary**
  Opens the New Summary dialog to create a new summary.
- **New Parameter**
  Opens the Create Parameter dialog to create a new parameter.
- **Import UDF Classes**
  Opens the UDF Classes dialog to import user defined formula classes.
- **Save**
  Saves the crosstab formula to the Formulas node.
- **Save As**
  Saves the crosstab formula as a new crosstab formula by specifying a new name.
- **Close**
  Closes the window.

Edit menu

- **Undo**
  Undoes an action.
- **Redo**
  Redoes an action.
- **Cut**
  Cuts the selected text in the editing panel.
- **Copy**
  Copies the selected text in the editing panel.
- **Paste**
  Pastes the text that was previously cut or copied in the editing panel.
- **Delete**
  Deletes the selected text in the editing panel.
- **Search**
  Brings out the Find/Replace dialog. You can then search for text in the editing panel, and replace the found text with different text.

View menu

- **Fields Panel**
  Specifies whether to show the Fields panel.
- **Functions Panel**
  Specifies whether to show the Functions panel.
- **Operators Panel**
  Specifies whether to show the Operators panel.
- **Sort Tree**
  Sorts functions in the Functions panel and fields in the Fields panel by names.

Formula menu

- **Check**
  Tests the syntax of your crosstab formula. If the syntax is incorrect, JReport gives the opportunity to correct the errors.
- **Add Bookmark**
  Adds a bookmark to a specific position.
- **Go to Previous Bookmark**
  Goes to the previous bookmark.
- **Go to Next Bookmark**
  Goes to the next bookmark.
- **Clear all Bookmarks**
  Clears all of the bookmarks.
- **Comment/Uncomment**
  Adds or removes comments.
- **Add Operators**
  Specifies a general operator to be used in the editing panel.
- **Color Converter**
  A color palette is provided for you to insert the HEX code of a color simply by clicking the corresponding color in the color palette instead of inputting the HEX code manually. If the color palette cannot meet your requirement, click More Color to open the Pick a Color dialog, then you can set a full range of color in the dialog.
- **Auto Finish**
  Enables the automatic input of the other part of a sign pair right after you type in the first part when editing crosstab formula.
- **Formula References**
  Shows the formulas that reference the selected UDF function in the Functions panel.
- **Properties**
  Shows properties of the crosstab formula in the Formula Properties dialog.
  - **Data Type**
    The data type of the returned value of the crosstab formula.
  - **Precision**
    The precision of the returned value of the crosstab formula. If the precision is set to 0, all characters of the returned value will be displayed; if it is set to N, only the first N characters will be displayed.
  - **Scale**
    The number of digits to the right of the decimal point for the returned value of the crosstab formula.

Help menu

- **Crosstab Formula Editor help**
  Displays the help document about this feature.

Toolbar

The following are commands on the toolbar:

- **New Formula**
  Creates a new crosstab formula.
- **New Summary**
  Creates a new summary.
- **New Parameter**
  Creates a new parameter.
- **Save**
  Saves the formula to the Formulas node.
- **Save As**
  Saves the formula as a new formula by specifying a new name.
- **Properties**
  Specifies the precision and scale properties of the crosstab formula.
- **Sort Tree**
  Sorts functions in the Functions panel and fields in the Fields panel by names.
- **Fields Panel**
  Specifies whether to show the Fields panel.
- **Functions Panel**
  Specifies whether to show the Functions panel.
- **Operators Panel**
  Specifies whether to show the Operators panel.
- **Formula Editor Help**
  Displays the help document about this feature.
- **Cut**
  Cuts the selected text in the editing panel.
- **Copy**
  Copies the selected text in the editing panel.
- **Paste**
  Pastes the text that was previously cut or copied in the editing panel.
- **Delete**
  Deletes the selected text in the editing panel.
- **Undo**
  Undoes an action.
- **Redo**
  Cancels undoing an action.
- **Search**
  Searches for text in the editing panel, and replace the found text with different text.
- **Check**
  Tests the syntax of your crosstab formula. If the syntax is incorrect, JReport gives the opportunity to correct the errors.
- **Add Bookmark**
  Adds a bookmark to a specific position.
- **Go to Previous Bookmark**
  Goes to the previous bookmark.
- **Go to Next Bookmark**
  Goes to the next bookmark.
- **Clear All Bookmarks**
  Clears all of the bookmarks.
- **Comment/Uncomment**
  Adds or removes comments.
- **Add Operators**
  Selects a general operator to be used in the editing panel.
- **Color Converter**
  A color palette is provided for you to insert the HEX code of a color simply by clicking the corresponding color in the color palette instead of inputting the HEX code manually.

Fields panel

Displays a list of those fields that are available for crosstab formulas. Those fields contain table columns, formulas, parameters, special fields and other crosstab formulas. Select one field and double-click it to insert it into the editing panel at the insertion point.

Functions panel

Displays a list of JReport functions and imported user defined formula functions that are available for crosstab formulas. When you select one function and double-click it, JReport will insert the selected function into the editing panel at the insertion point completely with its required syntax items (parentheses, commas, and so on). For the usage of these functions, refer to section Built-in functions and User-defined formula functions.

Operators panel

Displays a list of operators that are available for crosstab formulas. Select one operator and double-click it to insert the selected operator into the editing panel at the insertion point. For the usage of these operators, refer to section Operators.

Formula text panel

In this panel, you can build and edit your crosstab formula. There are several ways to work with crosstab formula:

- Select formula components from the Fields, Functions and Operators panels on the top of Crosstab Formula Editor, and then double-click the components. JReport will then insert them in the crosstab formula;
- Type your crosstab formula in the editing panel directly;
- Use the above two methods together;
- Paste crosstab formula text from the text document of other programs.

The line numbers of the crosstab formula are marked in the left of the panel to help you check it easily.

Related topics:

- Using crosstab formulas

Crosstab Wizard for page report

This wizard appears when you select the Crosstab layout in the New Page Report Tab dialog or New Page Report dialog and click OK, or right-click a crosstab and select Crosstab Wizard from the shortcut menu. It helps you to create or modify a crosstab in a page report.

The wizard consists of the following screens:

- Data screen
- Display screen
- Chart screen
- Filter screen
- Layout screen
- Style screen

Back

Goes back to the previous screen.

Next

Goes to the next screen.

Finish

Finishes creating or modifying the crosstab and closes this wizard.

Cancel

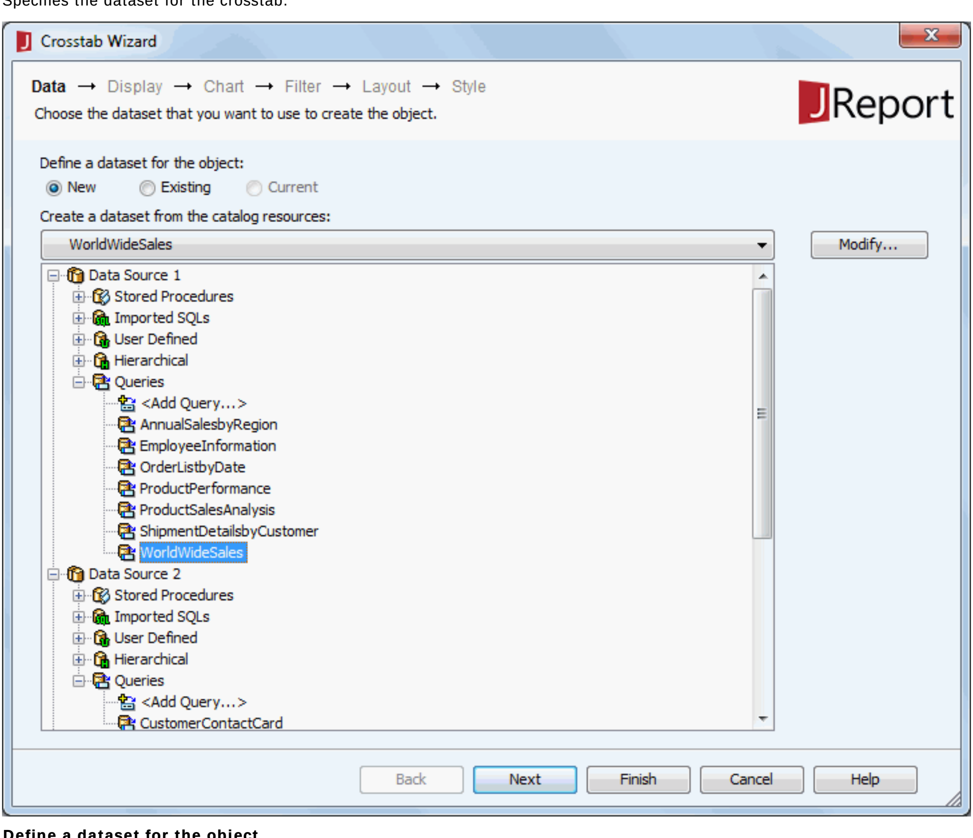
Does not retain changes and closes this wizard.

Help

Displays the help document about this feature.

Data screen

Specifies the dataset for the crosstab.



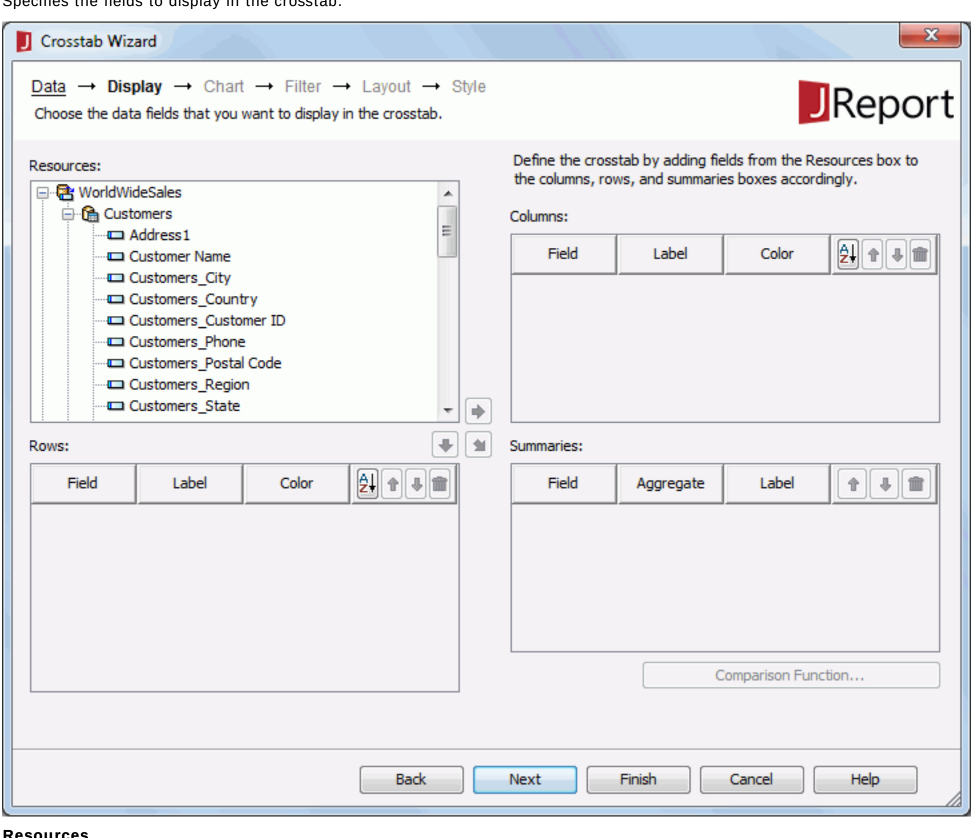
Define a dataset for the object

Specifies the dataset you want to use to create or edit the crosstab.

- **New**
  If checked, select a data source from the catalog resources to create a dataset that will be used to build the crosstab. When you choose to create the dataset from a query, you can click the Modify button to edit the query in the Query Editor if required.
- **Existing**
  If checked, select a dataset from the ones existing in the open report to create the crosstab. Click the Modify button to edit the dataset in the Dataset Editor if required.
- **Current**
  If checked, the current dataset used by the parent object will be applied to the crosstab.

Display screen

Specifies the fields to display in the crosstab.



Resources

Lists all the available data resources.

Adds the selected field to be displayed on the columns of the crosstab.

Adds the selected field to be displayed on the rows of the crosstab.

Adds the selected field on which to create summaries.

Replaces the selected field in the crosstab with the specified field in the Resources box.

Columns/Rows

- **Field**
  Lists the fields that will be displayed on the columns/rows of the crosstab.
- **Label**
  Specifies the display names for the selected fields. By default these are blank and no names will be created for the fields to label the columns/rows. You can double-click the cells to edit them if required.
- **Color**
  Specifies the background color of the selected cells.

Summaries

- **Field**
  Lists the fields that you select to create summaries.
- **Aggregate**
  Specifies the functions used to summarize data of the selected fields. For details about each function, see Math functions.
- **Label**
  Specifies the display names for the selected fields. By default these are blank and no names will be created for the fields to label the summaries. You can double-click the cells to edit them if required.
- **Comparison Function**
  Opens the Comparison Function dialog to add a comparison function as an aggregate for the crosstab.

Specifies in which manner to sort the field values.

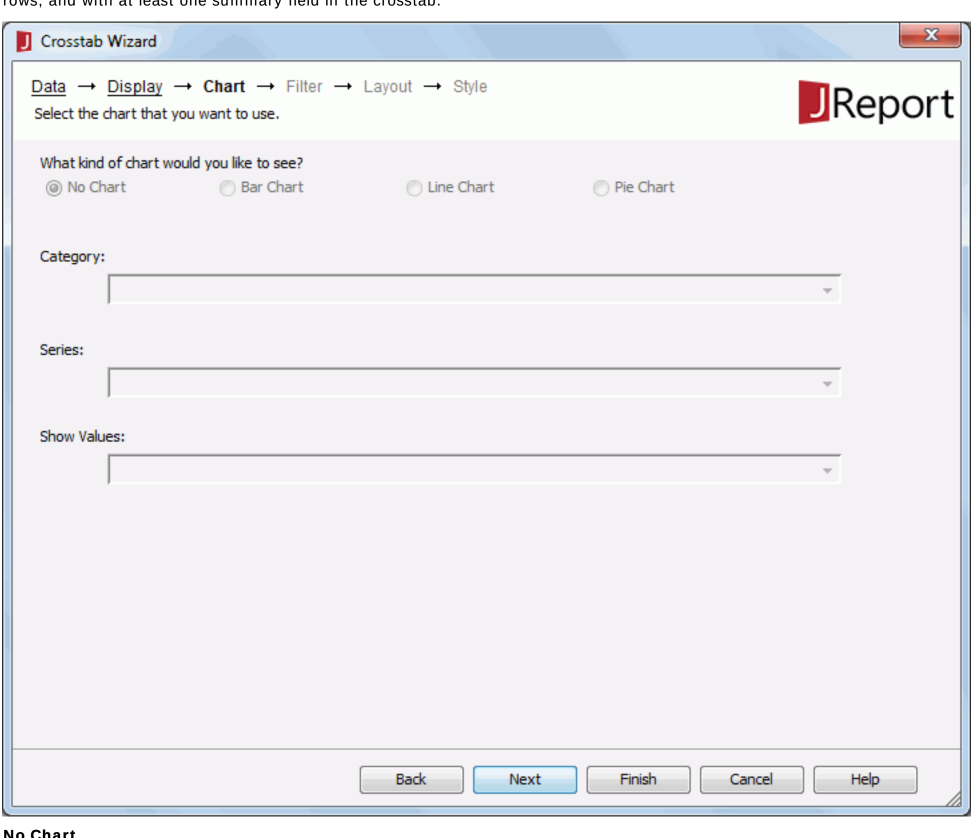
Moves the specified field one step up.

Moves the specified field one step down.

Removes the specified field that is not required in the crosstab.

Chart screen

Specifies to create a chart together with the crosstab, which will be placed above the crosstab in the report body. This screen is available only when you create a crosstab in a page report, and when there is at least one field inserted in the columns or rows, and with at least one summary field in the crosstab.



No Chart

Specifies not to create a chart.

Bar Chart

Specifies to create a Clustered Bar 2-D chart together with the crosstab.

Line Chart

Specifies to create a Line 2-D chart together with the crosstab.

Pie Chart

Specifies to create a Clustered Pie chart together with the crosstab.

Category

Lists the fields that have been added to the columns and rows of the crosstab. Choose the field you want to display on the category (X) axis of the chart from the drop-down list.

Series

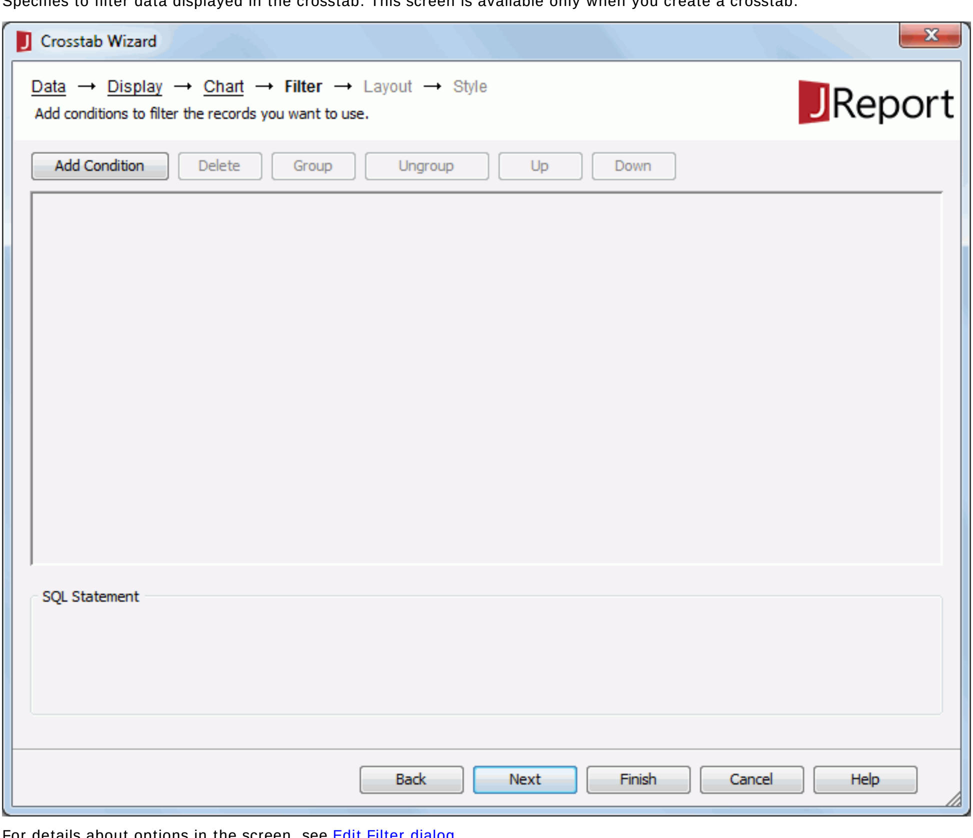
Lists the fields that have been added to the columns and rows of the crosstab. Choose the field you want to display on the series (Z) axis of the chart from the drop-down list.

Show Values

Lists the fields that have been selected to create summaries in the crosstab. Choose the value you want to display in the chart from the drop-down list.

Filter screen

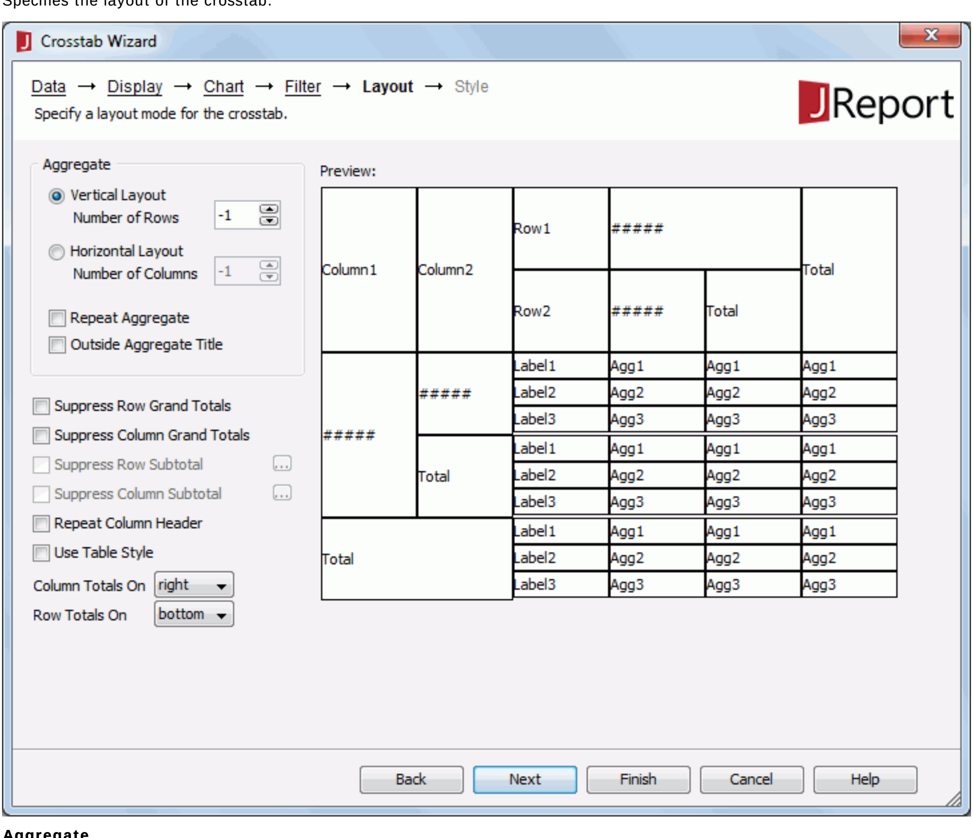
Specifies to filter data displayed in the crosstab. This screen is available only when you create a crosstab.



For details about options in the screen, see Edit Filter dialog.

Layout screen

Specifies the layout of the crosstab.



Aggregate

Specifies properties of the aggregate fields.

- **Vertical Layout**
  If selected, the aggregate layout will be set vertically.
  - **Number of Rows**
    Specifies the number of rows to be displayed in a crosstab. By default, it is -1, which means that there is only one column with which to arrange the aggregate fields vertically; if it is set to 1, that means there will be only one row to arrange the aggregate fields horizontally; if it is set to a number larger than 1, that means there will be this number of aggregate rows; if this number is larger than the number of aggregate fields in the crosstab, it will be treated as -1.
- **Horizontal Layout**
  If selected, the aggregate layout will be set horizontally.
  - **Number of Columns**
    Specifies the number of columns to be displayed in a column. By default, it is -1, which means that there is only one row with which to arrange the aggregate fields horizontally; if it is set to 1, that means there will be only one column to arrange the aggregate fields vertically; if it is set to a number larger than 1, that means there will be this number of aggregate rows; if this number is larger than the number of aggregate fields in the crosstab, it will be treated as -1.
- **Repeat Aggregate**
  Specifies whether or not to repeat the crosstab for different aggregate fields.

  Assume that you have created a crosstab on the query WorldWideSales in the catalog file SampleReports.cat as follows: added Products_Product Type ID as the column field, added Customers_Country as the row field, added Quantity and Discount as the aggregate fields and specified the functions as Sum, and applied the style Classic to the crosstab.

  - If checked, one cell will contain only one aggregate field value, while for the other aggregate fields, the crosstab will repeat. The crosstab will be shown as follows:

	1	2	Total
Canada	165	20	185
France	640	2155	2795
Germany	605	380	985
Netherlands	155	80	235
USA	9135	10060	19195
Total	10700	12695	23395
Canada	20.00	.00	20.00
France	85.00	70.00	155.00
Germany	115.00	100.00	215.00
Netherlands	30.00	30.00	60.00
USA	1,095.00	1,130.00	2,225.00
Total	1,345.00	1,330.00	2,675.00

  - If unchecked, one crosstab cell accommodates the values of all aggregate fields. The crosstab will be shown as follows:

	1	2	Total
Canada	165	20	185
	20.00	.00	20.00
France	640	2155	2795
	85.00	70.00	155.00
Germany	605	380	985
	115.00	100.00	215.00
Netherlands	155	80	235
	30.00	30.00	60.00
USA	9135	10060	19195
	1,095.00	1,130.00	2,225.00
Total	10700	12695	23395
	1,345.00	1,330.00	2,675.00

- **Outside Aggregate Title**
  Specifies whether to place the summary label on the leftmost column when the aggregate layout is vertical, or on the topmost row of the crosstab when the aggregate layout is horizontal.

Suppress Row Grand Totals

Specifies whether or not to show the grand totals of rows.

Suppress Column Grand Totals

Specifies whether or not to show the grand totals of columns.

Suppress Row Subtotal

Specifies whether or not to show the subtotals of rows. If required, you can click ... to open the Suppress Row Subtotal dialog to specify which subtotals of the rows will be suppressed, and which will be shown.

Suppress Column Subtotal

Specifies whether or not to show the subtotals of columns. If required, you can click ... to open the Suppress Column Subtotal dialog to specify which subtotals of the columns will be suppressed, and which will be shown.

Note: The settings of the above four options will only take effect in view mode.

Repeat Column Header

Specifies whether or not the header of column will be appeared on every page, if a crosstab spans more than one page.

Use Table Style

Specifies whether to add headers to the Total rows and columns.

Column Totals On

Specifies the position of total columns on the left or right of the aggregates.

Row Totals On

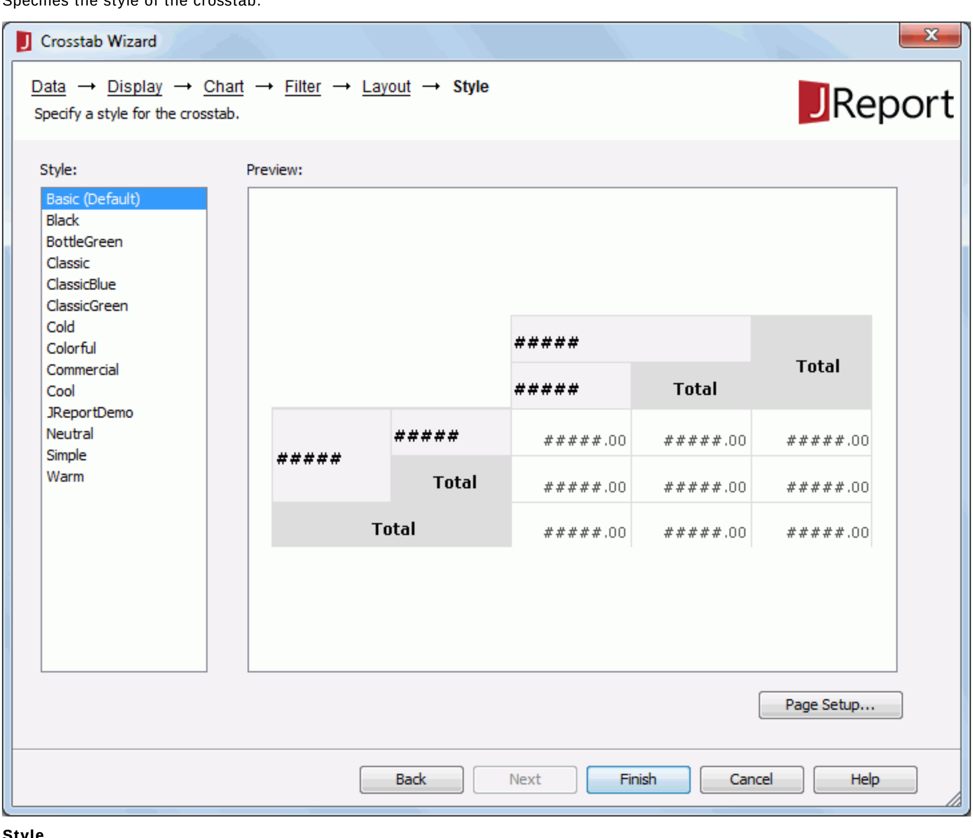
Specifies the position of total rows on the top or bottom of the aggregates.

Preview

Displays a preview of the selected layout.

Style screen

Specifies the style of the crosstab.



Style

Lists the available styles.

- **<Custom>**
  There is no any style information on it and it is only used to cover previous-version report which did not bind any style or the bound style cannot be found in the style list.

Preview

Displays a diagram illustrating the effect of the selected style on the crosstab.

Inherit Style

Specifies whether to make the crosstab take the style of its parent. This options is available only when you modify a crosstab and the crosstab is inserted into a banded object.

Page Setup

Opens the Page Setup dialog to specify page properties. Available only when creating a page report.

Related topics:

- Creating a crosstab report
- Modifying a crosstab

Crosstab Wizard for web report/library component

This wizard appears when you select Crosstab in the component type box and click OK in the New Library Component dialog, or right-click a crosstab and select Crosstab Wizard from the shortcut menu. It helps you to create or modify a crosstab in a web report or library component.

The wizard consists of the following screens:

- Data screen
- Display screen
- Filter screen
- Layout screen
- Style screen

Back
Goes back to the previous screen.

Next
Goes to the next screen.

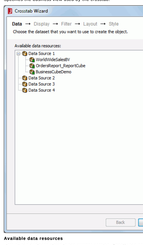
Finish
Finishes creating or modifying the crosstab and closes this wizard.

Cancel
Does not retain changes and closes this wizard.

Help
Displays the help document about this feature.

Data screen

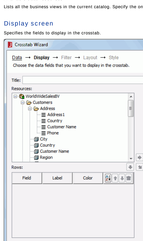
Specifies the business view used by the crosstab.



Available data resources
Lists all the business views in the current catalog. Specify the one you want to use.

Display screen

Specifies the fields to display in the crosstab.



Title
Specifies the title of the crosstab.

Resources
Lists all the available data resources. You can also create dynamic formulas and aggregations to use in the crosstab.

Adds the selected field to be displayed on the columns of the crosstab.

Adds the selected field to be displayed on the rows of the crosstab.

Adds the selected field on which to create summaries.

Replaces the selected field in the crosstab with the specified field in the Resources box.

Columns/Rows
- **Field**
  Lists the fields that will be displayed on the columns/rows of the crosstab.
- **Label**
  Specifies the display names for the selected fields. By default these are blank and no names will be created for the fields to label the columns/rows. You can double-click the cells to add them if required.
- **Color**
  Specifies the background color of the selected cells.

Summaries
- **Field**
  Lists the fields that you select to create summaries.
- **Aggregate**
  Specifies the functions used to summarize data of the selected fields. For details about each function, see Math functions.
- **Label**
  Specifies the display names for the selected fields. By default these are blank and no names will be created for the fields to label the summaries. You can double-click the cells to edit them if required.
- **Comparison Function**
  Opens the Comparison Function dialog to add a comparison function as an aggregate for the crosstab.

Specifies in which manner to sort the field values.

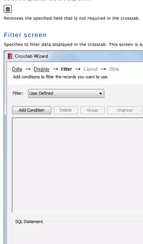
Moves the specified field one step up.

Moves the specified field one step down.

Removes the specified field that is not required in the crosstab.

Filter screen

Specifies to filter data displayed in the crosstab. This screen is available only when you create a crosstab.

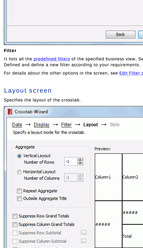


Filter
It lists all the predefined filters of the specified business view. Select one from the drop-down list to apply, or select User Defined and define a new filter according to your requirements.

For details about the other options in the screen, see Edit Filter dialog.

Layout screen

Specifies the layout of the crosstab.



Aggregate
Specifies properties of the aggregate fields.

- **Vertical Layout**
  If selected, the aggregate layout will be set vertically.
  - **Number of Rows**
    Specifies the number of rows to be displayed in a crosstab. By default, it is -1, which means that there is only one column with which to arrange the aggregate fields vertically; if it is set to 1, that means there will be only one row to arrange the aggregate fields horizontally; if it is set to a number larger than 1, that means there will be this number of aggregate rows; if this number is larger than the number of aggregate fields in the crosstab, it will be treated as -1.
- **Horizontal Layout**
  If selected, the aggregate layout will be set horizontally.
  - **Number of Columns**
    Specifies the number of columns to be displayed in a column. By default, it is -1, which means that there is only one row with which to arrange the aggregate fields horizontally; if it is set to 1, that means there will be only one column to arrange the aggregate fields vertically; if it is set to a number larger than 1, that means there will be this number of aggregate rows; if this number is larger than the number of aggregate fields in the crosstab, it will be treated as -1.
- **Repeat Aggregate**
  Specifies whether or not to repeat the crosstab for different aggregate fields.

  Assume that you have created a crosstab on the query WorldWideSales in the catalog file SampleReports.cat as follows: added Products_Product Type ID as the column field, added Customers_Country as the row field, added Quantity and Discount as the aggregate fields and specified the functions as Sum, and applied the style Classic to the crosstab.
  - If checked, one cell will contain only one aggregate field value, while for the other aggregate fields, the crosstab will repeat. The crosstab will be shown as follows:

	1	2	Total
Canada	166	20	186
France	640	2155	2795
Germany	605	380	985
Netherlands	155	80	235
USA	9135	10860	19195
Total	10700	13095	23795
Canada	20.00	.00	20.00
France	85.00	70.00	155.00
Germany	115.00	100.00	215.00
Netherlands	30.00	30.00	60.00
USA	1,095.00	1,130.00	2,225.00
Total	1,345.00	1,330.00	2,675.00

  - If unchecked, one crosstab cell accommodates the values of all aggregate fields. The crosstab will be shown as follows:

	1	2	Total
Canada	166	20	186
	20.00	.00	20.00
France	640	2155	2795
	85.00	70.00	155.00
Germany	605	380	985
	115.00	100.00	215.00
Netherlands	155	80	235
	30.00	30.00	60.00
USA	9135	10860	19195
	1,095.00	1,130.00	2,225.00
Total	10700	13095	23795
	1,345.00	1,330.00	2,675.00

- **Outside Aggregate Title**
  Specifies whether to place the summary label on the leftmost column when the aggregate layout is vertical, or on the rightmost row of the crosstab when the aggregate layout is horizontal.

Suppress Row Grand Totals
Specifies whether or not to show the grand totals of rows.

Suppress Column Grand Totals
Specifies whether or not to show the grand totals of columns.

Suppress Row Subtotal
Specifies whether or not to show the subtotals of rows. If required, you can click ... to open the Suppress Row Subtotal dialog to specify which subtotals of the rows will be suppressed, and which will be shown.

Suppress Column Subtotal
Specifies whether or not to show the subtotals of columns. If required, you can click ... to open the Suppress Column Subtotal dialog to specify which subtotals of the columns will be suppressed, and which will be shown.

Note: The settings of the above four options will only take effect in view mode.

Repeat Column Header
Specifies whether or not the header of column will be appeared on every page, if a crosstab spans more than one page.

Use Table Style
Specifies whether to add headers to the Total rows and columns.

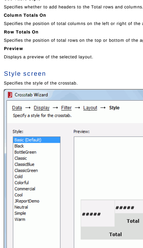
Column Totals On
Specifies the position of total columns on the left or right of the aggregates.

Row Totals On
Specifies the position of total rows on the top or bottom of the aggregates.

Preview
Displays a preview of the selected layout.

Style screen

Specifies the style of the crosstab.



Style
Lists the available styles.

- **<Custom>**
  There is no any style information on it and it is only used to cover previous-version report which did not bind any style or the bound style cannot be found in the style list.

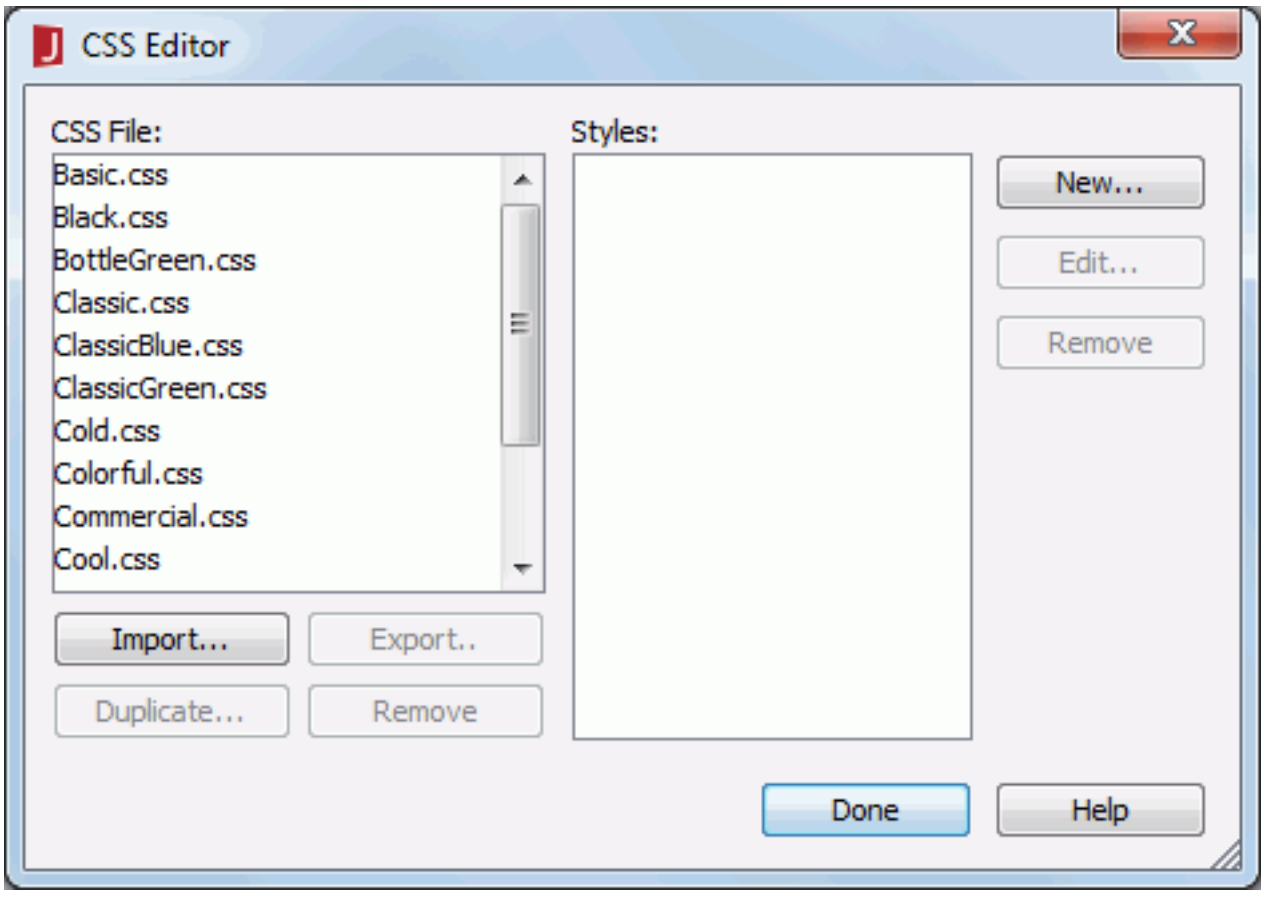
Preview
Displays a diagram illustrating the effect of the selected style on the crosstab.

Related topics:

- Creating a library component
- Modifying a crosstab

CSS Editor dialog

This dialog appears when you click Edit > CSS Editor on the menu bar. It helps you edit the styles in a specified CSS file.



The following are details about options in the dialog:

CSS File

Lists all the available CSS files. Select the specified CSS file to edit its styles.

Styles

Displays all the available styles in the selected CSS file.

Import

Imports a CSS file from a specified directory.

Export

Exports the selected CSS file to a specified directory.

Duplicate

Duplicates the selected CSS file.

Remove

Removes the selected CSS file.

New

Opens the New CSS Style dialog to create a CSS style.

Edit

Opens the CSS Style Definition dialog to edit the selected style.

Remove

Removes the selected style.

Done

Applies the changes and closes the dialog.

Help

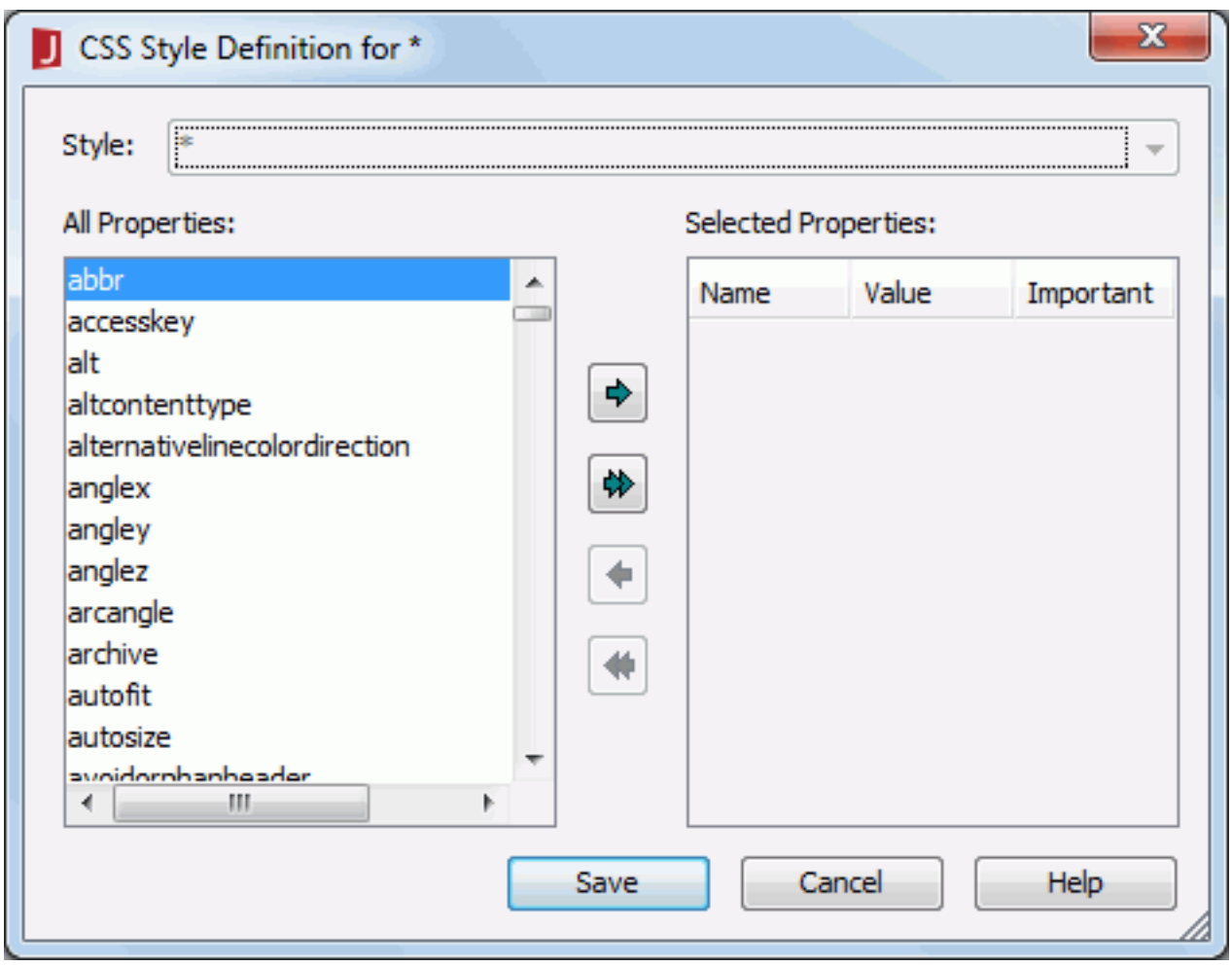
Displays the help document about this feature.

Related topics:

- Managing CSS styles

CSS Style Definition dialog

This dialog appears when you click the Edit button in the CSS Editor dialog or click OK when you select an existing CSS file in the New CSS Style dialog. It helps you to edit the specified style as required.



The following are details about options in the dialog:

Style

Shows the style that is to be edited.

All Properties

Lists all the properties of the selected style.

Selected Properties

Lists all the selected properties that you want to specify values to apply to this selected style.

- **Name**
  Lists the names of the properties.
- **Value**
  Specifies values for the selected properties.
- **Important**
  Specifies whether this property is important or not.

Adds a property to apply to the style.

Adds all the properties listed in the All Properties box to apply to the style.

Removes a property that is not required from the Selected Properties box.

Removes all the properties listed in the Selected Properties box.

Save

Applies the changes and closes this dialog.

Cancel

Does not retain the changes and closes this dialog.

Help

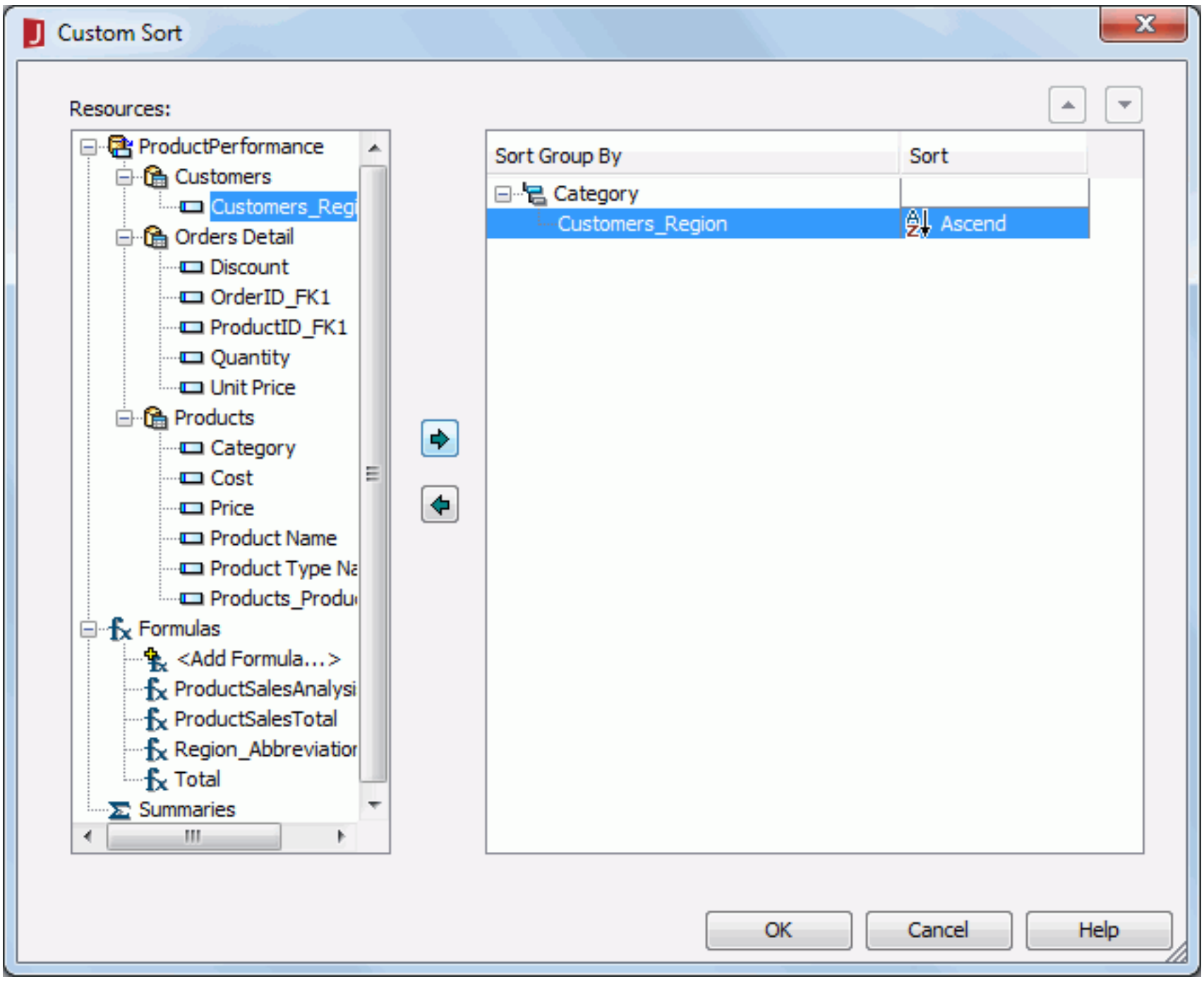
Displays the help document about this feature.

Related topics:

- Managing CSS styles

Custom Sort dialog

The dialog appears when you select Custom Sort from the Sort column in the Group screen of the report wizard. It helps you to set the sorting manner of groups at the specified group level.



The following are details of the dialog:

Resources

Lists all the available data resources.

Sort Group By

Displays the fields on which the groups will be sorted.

Sort

Specifies the sorting manner.

- **Ascend**
  If selected, groups will be sorted by the values of the first record in each group on the specified field in ascending order.
- **Descend**
  If selected, groups will be sorted by the values of the first record in each group on the specified field in descending order.

Adds the selected field as the sort by field.

Removes the specified sort by field if it is not required.

Moves the selected sort by field one step up.

Moves the selected sort by field one step down.

OK

Applies the changes and closes the dialog.

Cancel

Cancels the changes and exits the dialog.

Help

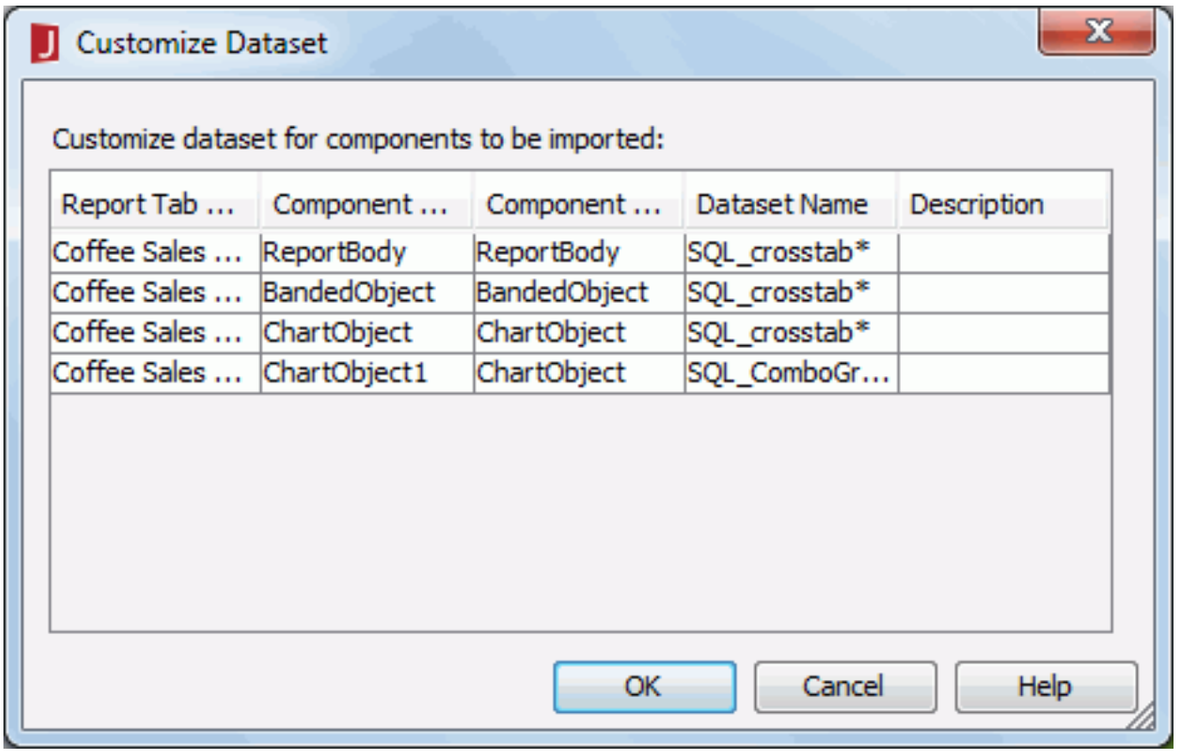
Displays the help document about this feature.

Related topics:

- Setting the sort manner

Customize Dataset dialog

This dialog appears when you check the Customize Dataset for Components option and click the OK button in the Select Report Tab dialog. It helps you to customize the datasets used by components of the to be imported report tabs.



The following are details about options in this dialog:

Report Tab Name

Displays the name of the report tab in which the component is.

Component Name

Displays the name of the component.

Component Type

Displays the type of the component.

Dataset Name

Lists the datasets you can choose for the component.

Description

Displays the description of the component.

OK

Applies the changes and closes the dialog.

Cancel

Does not retain any changes and closes the dialog.

Help

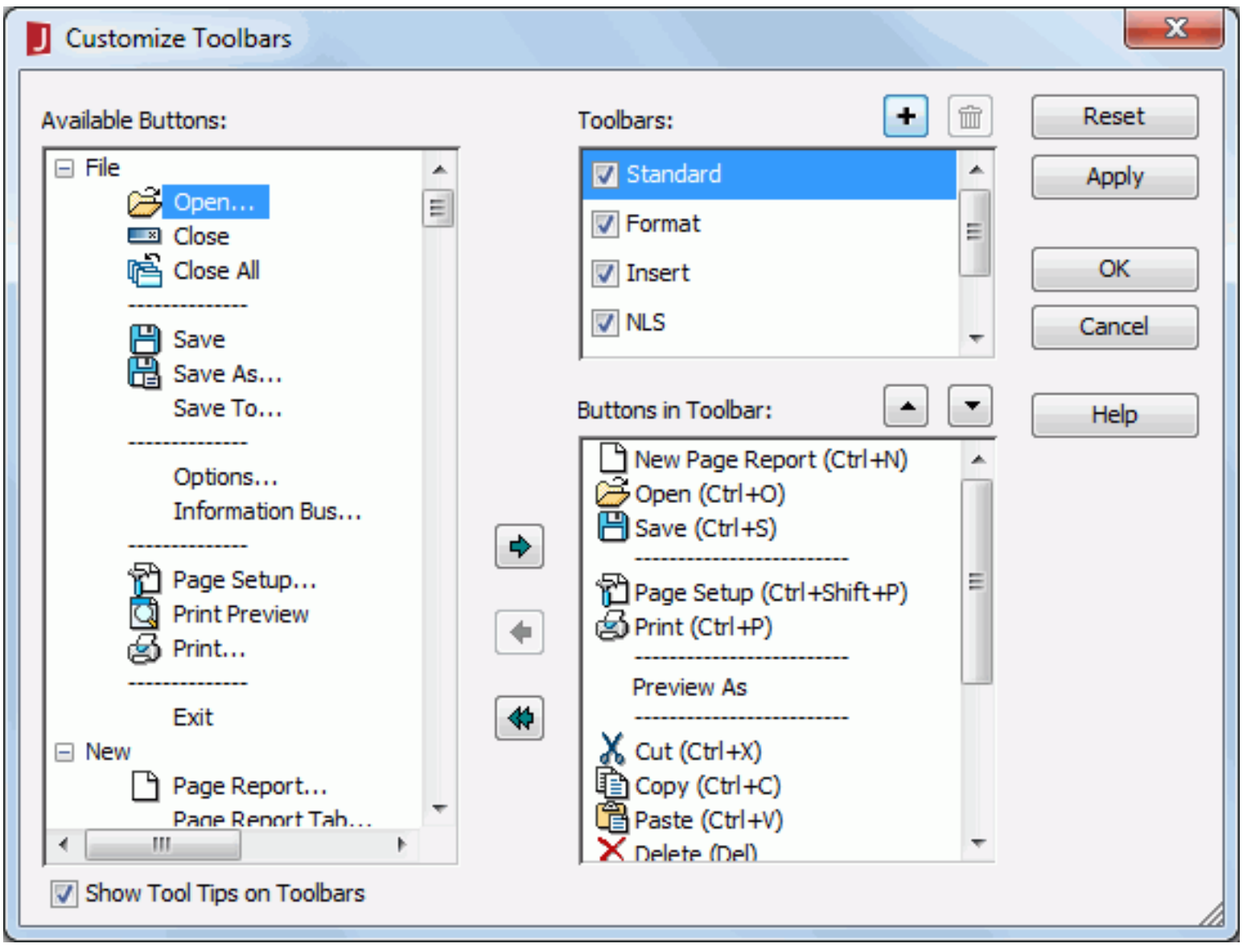
Displays the help document about this feature.

Related topics

- Importing report tabs into a page report

Customize Toolbars dialog

This dialog appears when you click View > Toolbar > Customize, or right-click the space on any toolbar and select Customize from the shortcut menu. To make JReport Designer more convenient to use, you can customize the toolbars in this dialog.



The following are details about options in the dialog:

Available Buttons

Lists all the buttons that are available to be shown on the toolbars.

Toolbars

Lists all the available toolbars. Check the checkboxes ahead of the toolbars you want to display in JReport Designer. You can click or to add more toolbars or remove a selected toolbar. The default toolbars cannot be removed.

Buttons in Toolbar

Lists all the buttons that are available on the toolbar selected in the Toolbars box. You can add more buttons to the toolbar or remove the unwanted ones.

Adds the selected buttons in the Available Buttons box into the specified toolbar.

Removes the selected buttons from the specified toolbar.

Removes all the buttons from the specified toolbar.

Clicks to add a new toolbar.

Removes the selected toolbar.

Moves the selected button one step up.

Moves the selected button one step down.

Show Tool Tips on Toolbars

Specifies whether to show the tooltips when you hover the mouse pointer over the buttons on the toolbars.

Reset

Resets the settings in this dialog to their default settings.

Apply

Applies the settings that you have made in the dialog.

OK

Accepts all changes and closes the dialog.

Cancel

Does not retain any changes and closes the dialog.

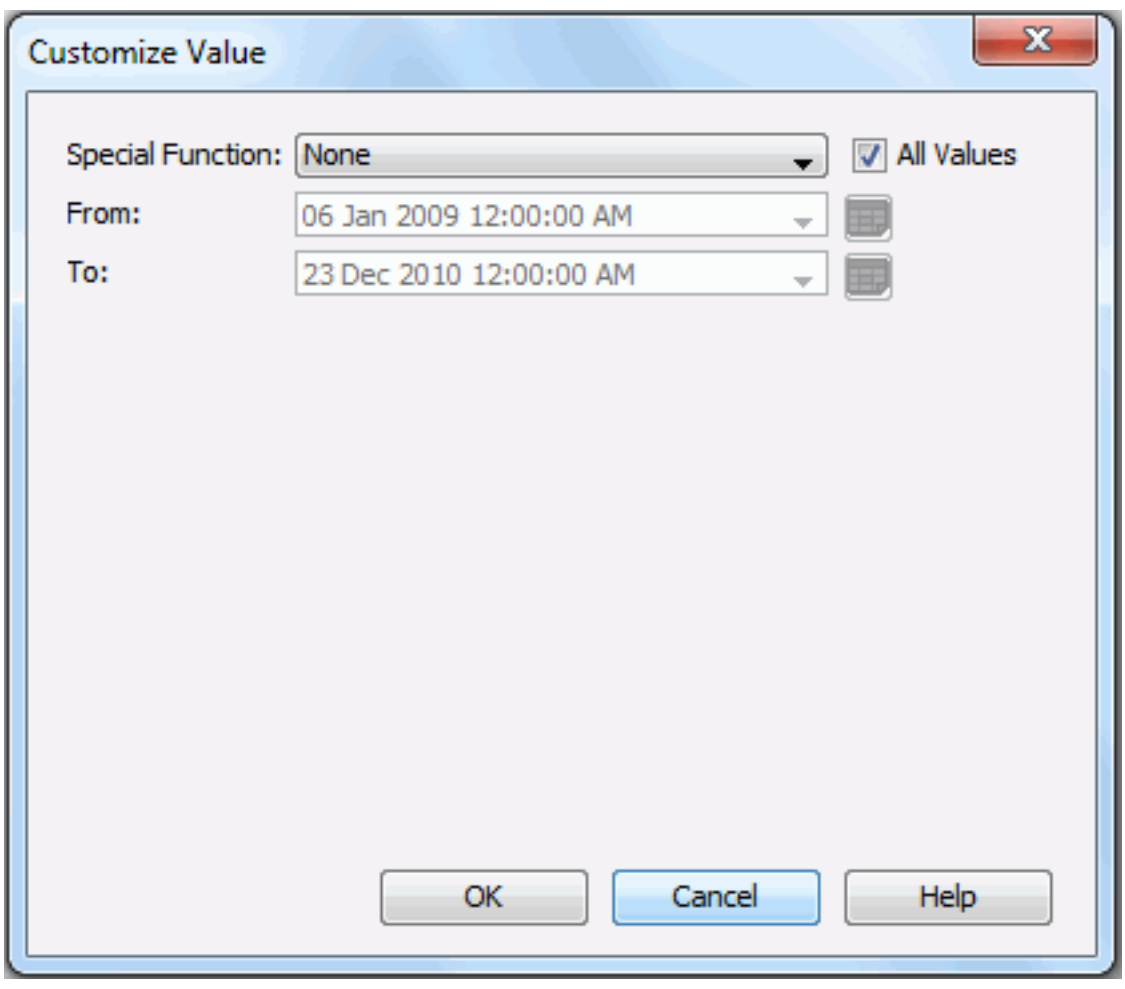
Help

Displays the help document about this feature.

Customize Value dialog

The dialog appears when you specify the fields to bind to a slider and click the Customize button in the Insert Slider dialog or Edit Slider dialog. It helps you to customize the values in the selected fields that you want to use on the slider. The dialog varies with the slider type: Range of Values or Single Value.

If the slider type is Range of Values, options in the dialog are as follows.



Special Function

Specifies a special function for the selected fields. Available only when the selected fields are of Date/Time type.

All Values

Specifies whether or not to use all values in the selected fields for the slider.

From

Specifies the start value of the value range. If the selected fields are of Date/Time type, you can also click the calendar button to select the date. Available only when All Values is unchecked.

To

Specifies the end value of the value range. If the selected fields are of Date/Time type, you can also click the calendar button to select the date. Available only when All Values is unchecked.

OK

Applies the values to the slider and closes the dialog.

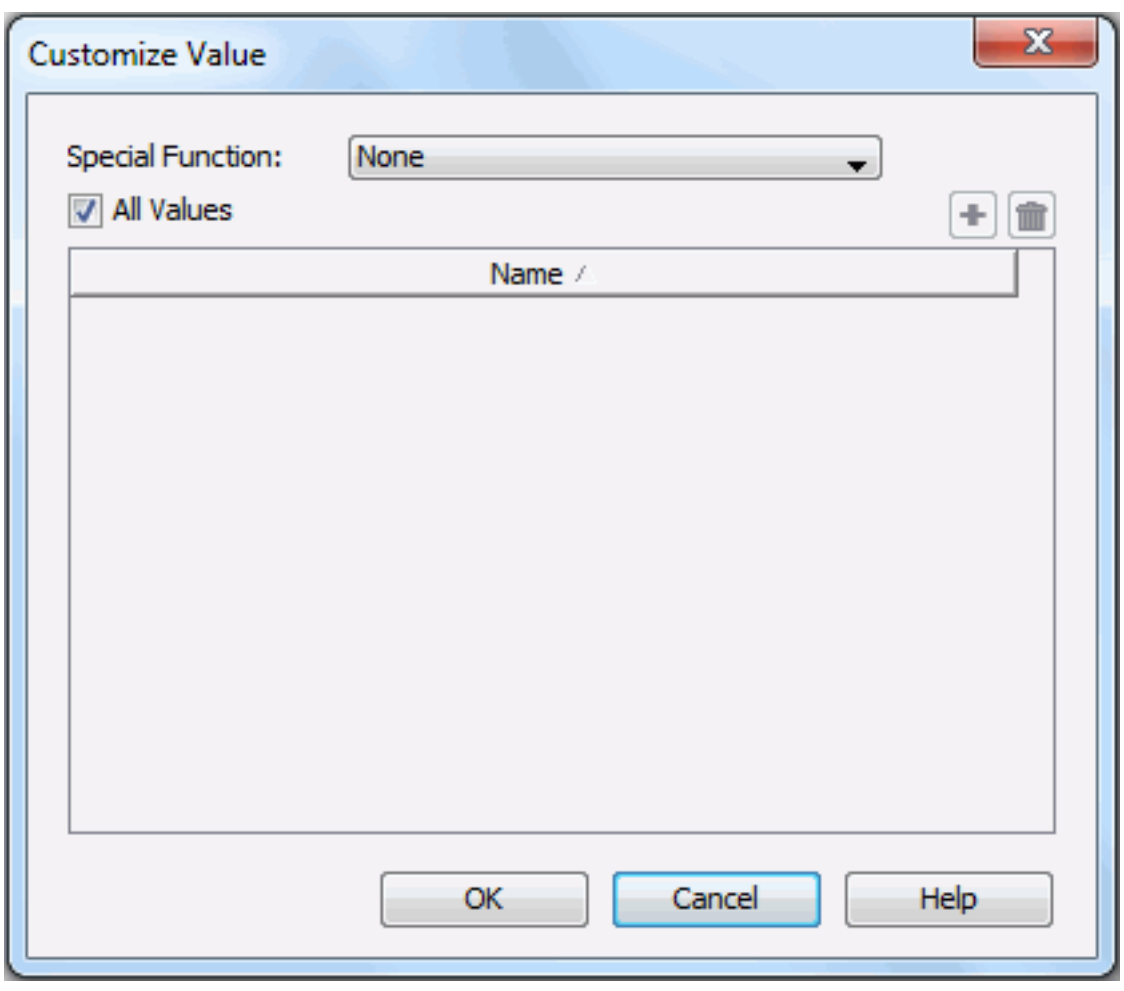
Cancel

Cancels the operation and closes the dialog.

Help

Displays the help document about this feature.

If the slider type is Single Value, options in the dialog are as follows.



Special Function

Specifies a special function for the selected fields. Available only when the selected fields are of Date/Time type.

All Values

Specifies whether or not to use all values in the selected fields for the slider.

Name

Lists the values you select. You can click on the Name header to sort the selected values.

Adds a value line to the Name box. To specify a value, you can either type it into the text box directly or select from the drop-down list.

Removes the selected value line.

OK

Applies the values to the slider and closes the dialog.

Cancel

Cancels the operation and closes the dialog.

Help

Displays the help document about this feature.

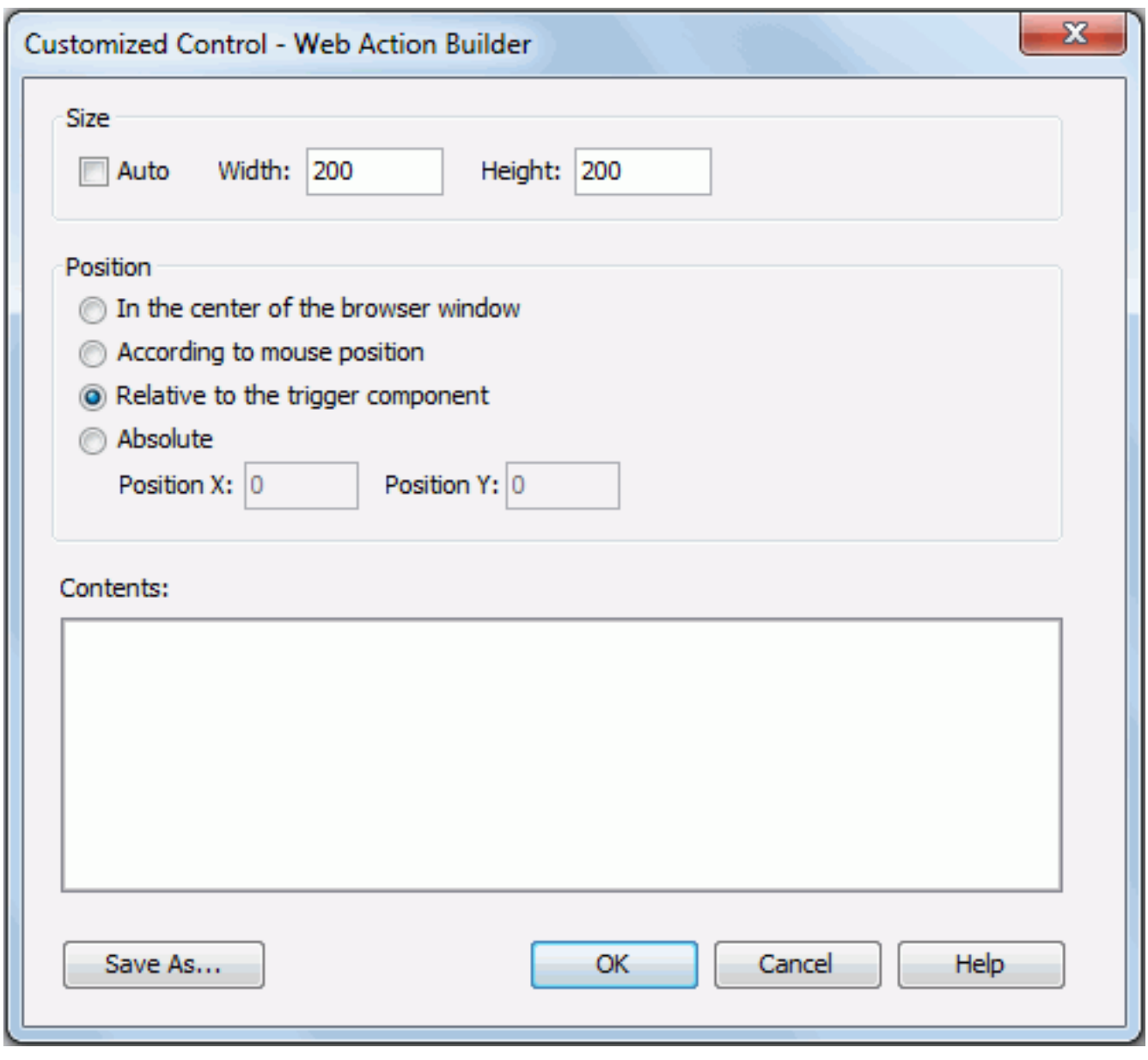
Related topics:

- Using slider or filter control to filter data

Customized Control •Web Action Builder dialog

The dialog helps you to bind, create, or edit a customized control. It is displayed when you do either of the following:

- Select Customized Control and then click OK in the Web Action List dialog.
- In the Customized Control Manager dialog, click . Then in the New Customized Control File dialog, provide a file name and click OK.
- In the Customized Control Manager dialog, right-click a customized control file and then select Edit from the shortcut menu.



The following are details about options in this dialog:

Size

Specifies the size of the customized control shown in the browser window at runtime.

- **Auto**
  Specifies whether the control is automatically sized according to its contents.
- **Width**
  Specifies the width of the control in pixels. Enabled when Auto is unchecked.
- **Height**
  Specifies the height of the control in pixels. Enabled when Auto is unchecked.

Position

Specifies the position of the customized control in the browser window at runtime:

- **In the center of the browser window**
  The customized control is placed in the center of the browser window at runtime.
- **According to mouse position**
  The customized control is placed where the mouse is clicked. The option is useful when the event is concerned with mouse action.
- **Relative to the trigger component**
  The position of the customized control is placed where the trigger object is.
- **Absolute**
  If the option is checked, you can further specify the following two options:
  - **Position X**
    Specifies the absolute X position in pixels.
  - **Position Y**
    Specifies the absolute Y position in pixels.

Contents

Input HTML fragment with JavaScript code. The HTML fragment should be plain text that is children DOM element of HTML Body.

JavaScript in HTML fragment can use web action APIs. For details, refer to index.html in the `<server_install_root>\public_html\webos\doc` folder. There is a sample of the contents provided in Sample of customized control contents.

Save As

Displays the New Customized Control File dialog to save the customized control properties into a file.

OK

- Saves the customized control properties into the web action and exits the current dialog if the dialog is opened after you select Customized Control and then click OK in the Web Action List dialog.
- Saves the customized control properties into the specified file and exits the current dialog if the dialog is opened after you click in the Customized Control Manger dialog and then provide a file name and click OK in the New Customized Control File dialog.
- Saves the changes to the customized control file if the dialog is opened after you right-click a customized control file and then select Edit from the shortcut menu in the Customized Control Manger dialog.

Cancel

Cancels the settings and exits the dialog.

Help

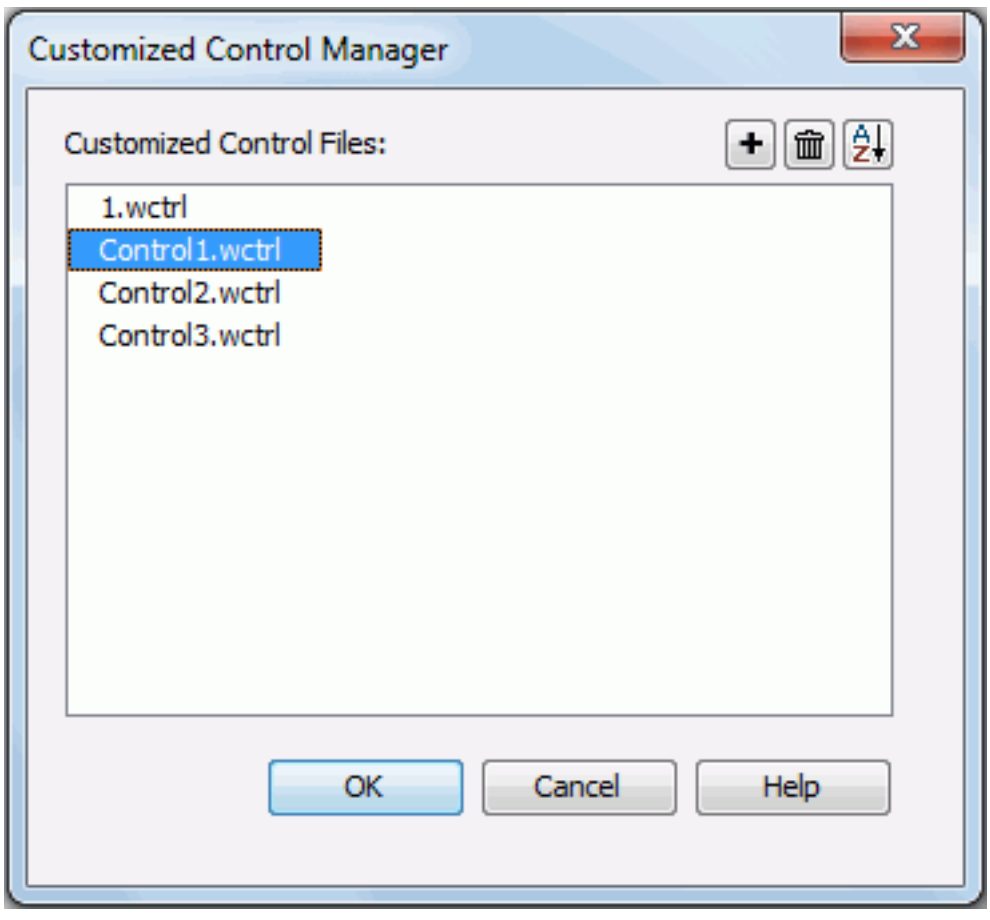
Displays the help document about this feature.

Related topics:

- Defining customized control in JReport Designer

Customized Control Manager dialog

This dialog appears when you click Edit > Customized Control Manager on the menu bar, or select Customized Control from Lib and click OK in the Web Action List dialog. It helps you to manage customized controls in the library.



The following are details about options in the dialog:

Customized Control Files

Lists all the customized control files in the customized control library.

Right-click a customized control file and the following options are available on the shortcut menu:

- **Duplicate**
  Creates a copy of the selected file.
- **Edit**
  Edit the customized control in the Customized Control - Web Action Builder dialog.
- **Delete**
  Removes the file from the library.
- **Rename**
  Modify the name of the file and then click outside of the name area to accept the change.

Displays the New Customized Control File dialog for creating a new customized control file by naming the file and then defining the customized control.

Removes the selected customized control file from the library.

Sorts the files according to their names ascendingly or descendingly. The descending order will not be remembered and whenever you access the dialog it is always the ascending order by default.

OK

- Loads the customized control properties from the selected file and exits the dialog if you are defining customized control. The button is enabled when a file is selected.
- Exits the dialog if you are managing customized controls.

Cancel

- Cancels selecting a customized control file and exits the dialog if you are defining customized control.
- Exits the dialog if you are managing customized controls.

Help

Displays the help document about this feature.

Related topics:

- Managing customized controls

Customized Function dialog

This dialog appears when you click the Customize item in the Special Function drop-down list in the New Summary dialog, Edit Summary dialog, or the Group screen of the report wizard. Options in the dialog vary according to the data type of the selected group by field.

OK

Accepts all changes and closes the dialog.

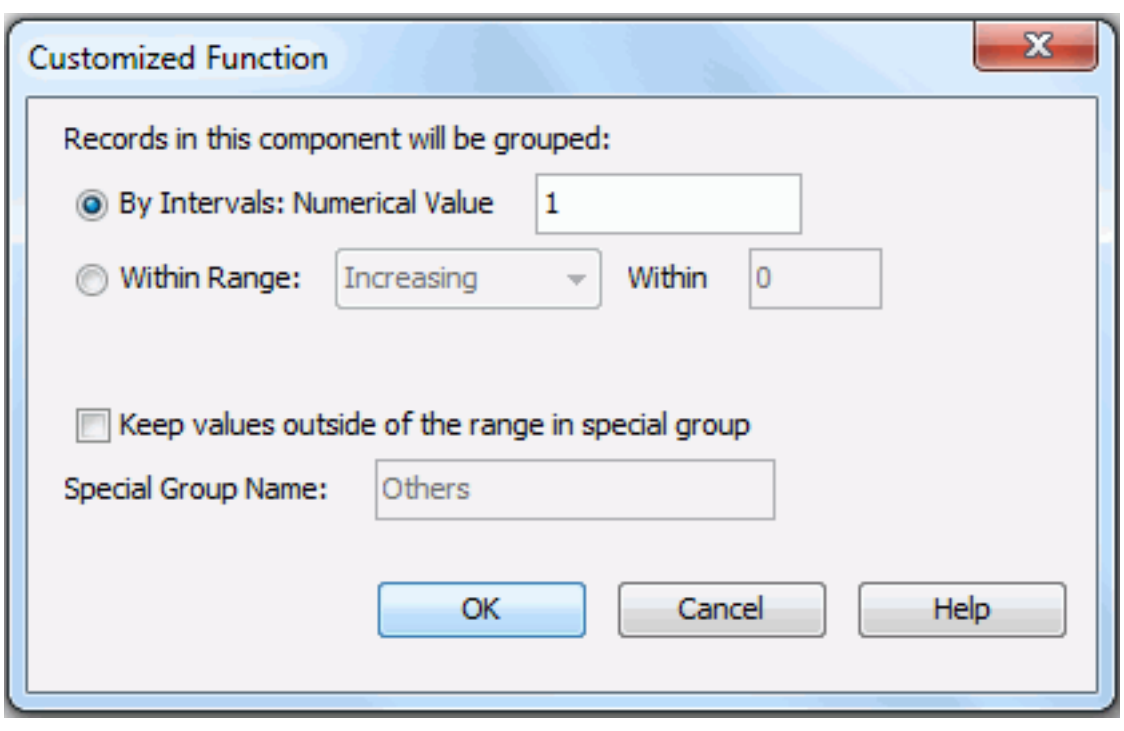
Cancel

Does not retain any changes and closes the dialog.

Help

Displays the help document about this feature.

If the group by field is of Numeric type, the dialog appears



By Intervals

Specifies to group the data by intervals.

- **Numerical Value**
  Specifies the intervals of the group.

Within Range

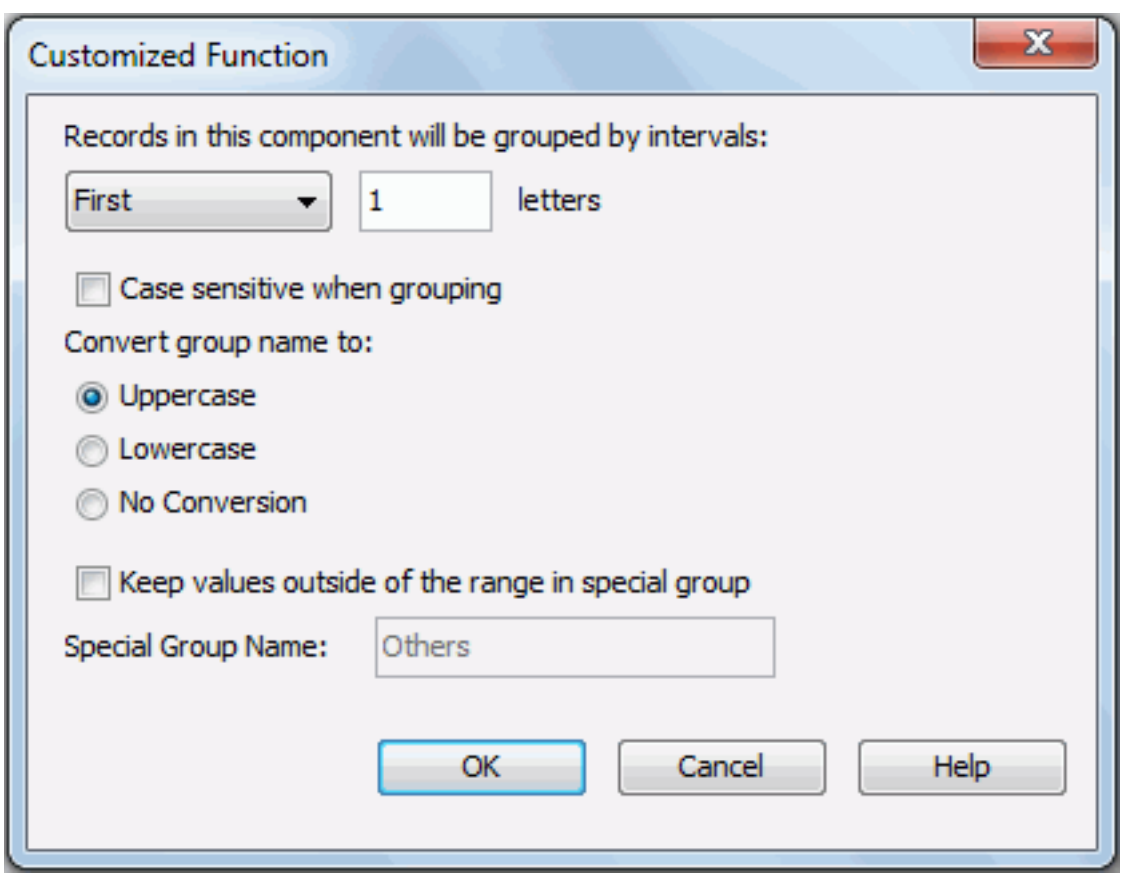
Specifies to group data within certain range. If this option is checked, you need to enter a value as the range in the Within text box and specify how to apply the range, to increasing data or decreasing data, by selecting the item from the drop-down list.

Keep values outside of the range in special group

Specifies to put values that do not fall within the defined interval or range in a new special group.

- **Special Group Name**
  Specifies the name for the special group. By default, the group name is Others. You can double-click in the text field to rename it.

If the group by field is of String type, the dialog appears



First/Last N letters

Specifies the intervals with which to group the report data. N should be an integer no larger than 255.

Case sensitive when grouping

If checked, JReport will distinguish between uppercase and lowercase characters when grouping.

Convert group name to

Specifies how to convert the group name. If Case sensitive when grouping is selected, this option will be disabled.

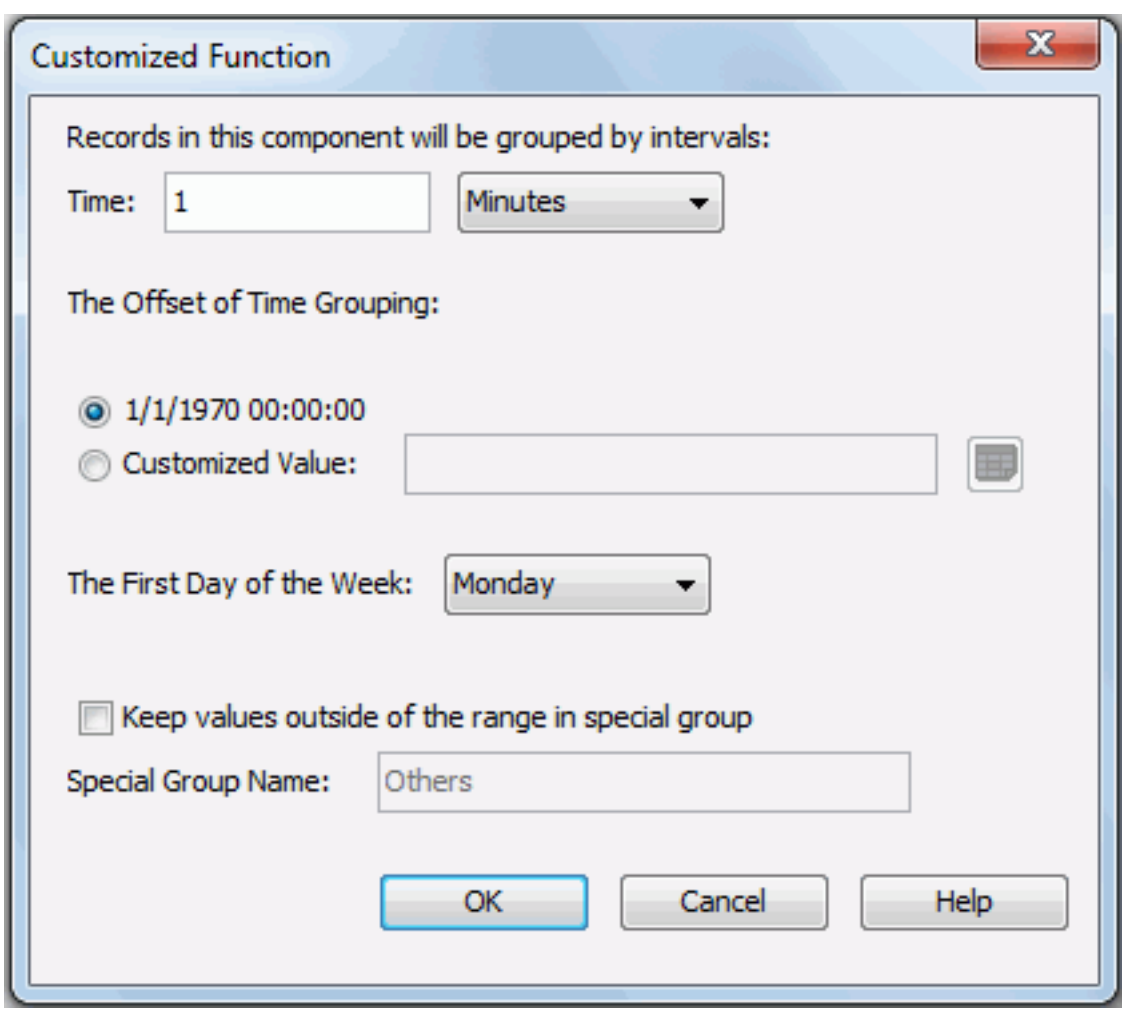
- **Uppercase**
  Specifies to convert the group names to uppercase.
- **Lowercase**
  Specifies to convert the group names to lowercase.
- **No Conversion**
  Specifies not to convert any group names.

Keep values outside of the range in special group

Specifies to put values that do not fall within the defined interval in a new special group.

- **Special Group Name**
  Specifies the name for the special group. By default, the group name is Others. You can double-click in the text field to rename it.

If the group by field is of Date/Time type, the dialog appears



Time

Specifies the time which will be applied as the intervals to group data. Select the unit of time from the drop-down list to the right of the text field. It could be second, minute, hour, day, week, month, quarter, half year and year.

The Offset of Time Grouping

Specifies the offset used to group the data.

- **1/1/1970 00:00:00**
  The default offset defined by JReport.
- **Customized Value**
  If checked, you can define your required offset by clicking the calendar button.

The First Day of the Week

Specifies which day will be regarded the first day of a week.

Keep values outside of the range in special group

Specifies to put values that do not fall within the defined interval in a new special group.

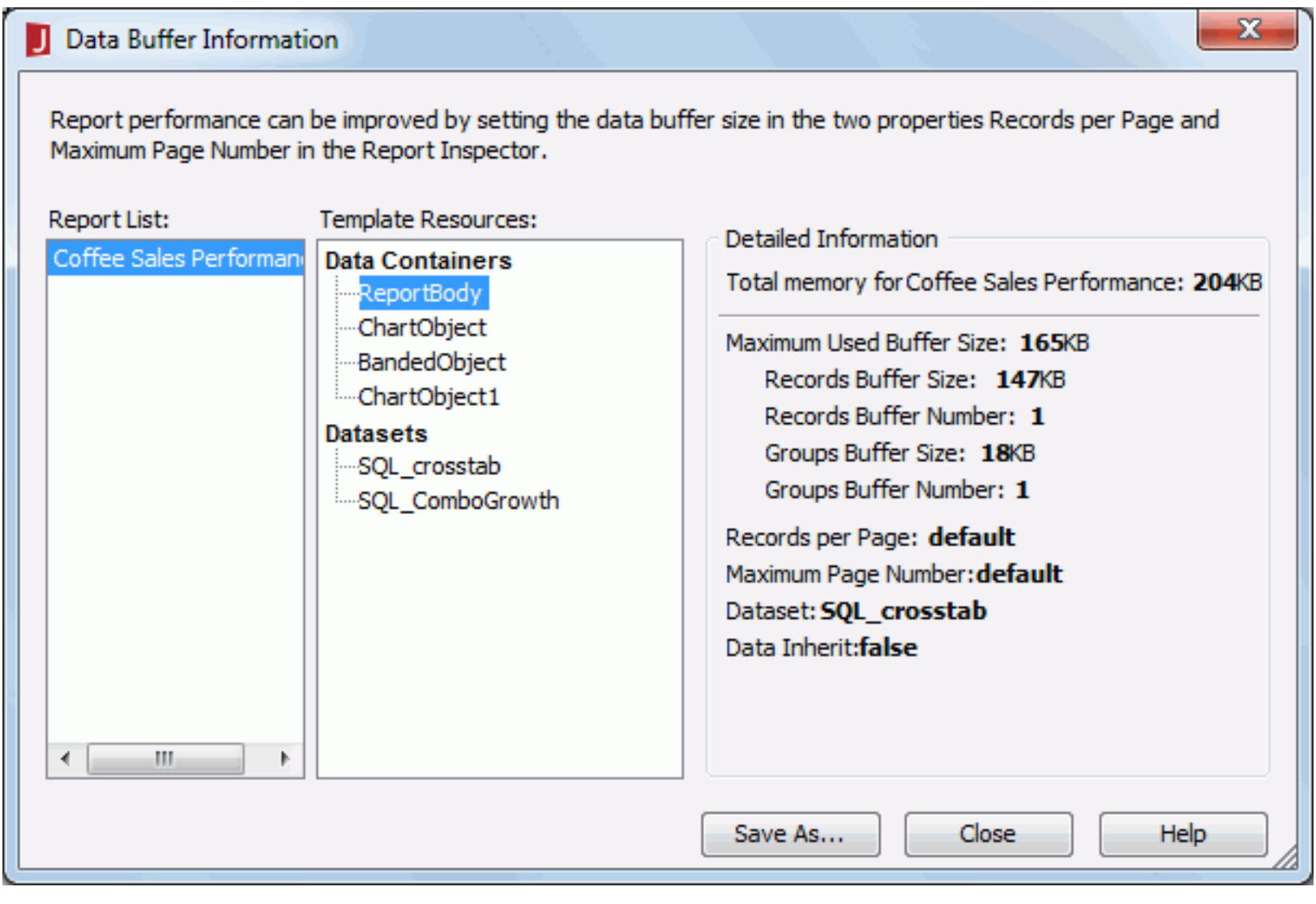
- **Special Group Name**
  Specifies the name for the special group. By default, the group name is Others. You can double-click in the text field to rename it.

Related topics:

- Specifying special function for group by field

Data Buffer Information dialog

This dialog appears when you click View > Data Buffer Information on the menu bar. It helps you to view the information of data buffer in the current report and its subreports (make sure the report is viewed successfully once at least). You can adjust the performance and footprint for the reports according to the data buffer information.



The following are details about options in the dialog:

Report List

Lists the current report and its subreports' instance names.

Template Resources

Displays the data containers and datasets of the report and its subreports.

Detailed Information (for data container)

- **Total memory for**
  Displays the total buffer size in the current report and its subreports. It is the sum of the Maximum Used Buffer Size for all of data containers in the current report and its subreports.
- **Maximum Used Buffer Size**
  Displays the maximum buffer size which the specified data container occupies. It is the sum of Records Buffer Size and Groups Buffer Size.
- **Records Buffer Size**
  Displays the maximum usage size for all the records•buffers in running memory.
- **Records Buffer Number**
  Displays the number of all the records•buffers after running memory.
- **Groups Buffer Size**
  Displays the maximum usage size for all the groups•buffers in running memory.
- **Groups Buffer Number**
  Displays the number of all the groups•buffers after running memory.
- **Records per Page**
  Displays the number of records in each page in the data buffer. You can set it in the Report Inspector to improve the report performance.
- **Maximum Page Number**
  Displays the maximum number of pages in the data buffer. You can set it in the Report Inspector to improve the report performance.
- **Dataset**
  Displays the name of the dataset which the specified data container uses.
- **Data Inherit**
  Indicates whether the datasets for this component is inherited from another component.

Detailed Information (for dataset)

- **Total memory for**
  Displays the total buffer size in the current report and its subreports. It is the sum of the Maximum Used Buffer Size for all of data containers in the current report and its subreports.
- **Maximum Used Buffer Size**
  Displays the maximum buffer size which the specified dataset occupies. It is the sum of Records Buffer Size and Groups Buffer Size.
- **Records Buffer Size**
  Displays the maximum usage size for all the records' buffers in running memory.
- **Records Buffer Number**
  Displays the number of all the records' buffers after running memory.
- **Groups Buffer Size**
  Displays the maximum usage size for all the groups' buffers in running memory.
- **Groups Buffer Number**
  Displays the number of all the groups' buffers after running memory.
- **Records per Page**
  Displays the number of records in each page in the data buffer. You can set it in the Report Inspector to improve the report performance.
- **Maximum Page Number**
  Displays the maximum number of pages in the data buffer. You can set it in the Report Inspector to improve the report performance.

Save As

Saves the data buffer information in the specified directory with XML format.

Close

Exits the dialog.

Help

Displays the help document about this feature.

Notes:

- The value of buffer size shows information of a report's latest successful running.
- The values of Records per Page and Maximum Page Number are default if they are not set.

Data Format dialog

This dialog appears when you click to specify the data format for a chart in the Report Inspector. The options in the dialog varies according to different sources it is opened from.

When you open the dialog to specify the data format for the category axis (the X axis) in a chart, it consists of two tabs: major label tab and minor label tab.

OK

Applies the changes and closes the dialog.

Cancel

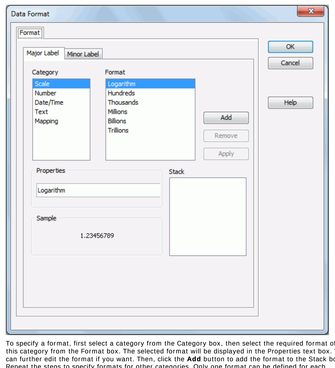
Does not retain any changes and closes the dialog.

Help

Displays the help document about this feature.

Major Label tab

Specifies the data format of the major tick mark labels on the axis.



To specify a format, first select a category from the Category box, then select the required format of this category from the Format box. The selected format will be displayed in the Properties text box. You can further edit the format if you want. Then, click the **Add** button to add the format to the Stack box. Repeat the steps to specify formats for other categories. Only one format can be defined for each category.

The following are details about options in the tab:

Category

Specifies different formats from the following categories to display values on the chart.

- **Scale**
  Divides the value by hundreds, thousands, and so on.
- **Number**
  Re-formats the number value (Original example: 123456).
- **Date/Time**
  Re-formats the date/time value (Example, Wednesday, December 25, 00:00:00 GMT-08:00 2002).
- **Text**
  Specifies the length for the value.
- **Mapping**
  Maps the new value to one or more values.

Format

Lists all the formats of the selected category. For details about each format, see Data Format.

Properties

Specifies the properties for the selected format.

Sample

Displays the selected format effects.

Stack

Lists all the formats that you selected from different categories.

Add

Adds a format to the Stack box.

Remove

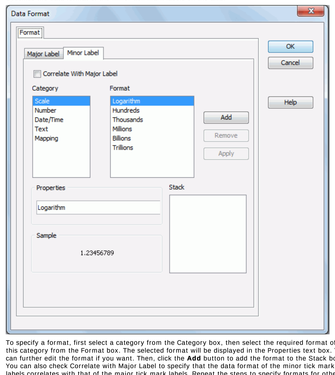
Removes a format from the Stack box.

Apply

Applies the specified format to values on the chart.

Minor Label tab

Specifies the data format of the minor tick mark labels on the axis.



To specify a format, first select a category from the Category box, then select the required format of this category from the Format box. The selected format will be displayed in the Properties text box. You can further edit the format if you want. Then, click the **Add** button to add the format to the Stack box. You can also check Correlate with Major Label to specify that the data format of the minor tick mark labels correlates with that of the major tick mark labels. Repeat the steps to specify formats for other categories. Only one format can be defined for each category.

The following are details about options in the tab:

Correlate with Major Label

If checked, the data format of the minor tick mark labels will correlate with that of the major tick mark labels automatically. Only when it is unchecked can the format properties of the minor tick mark labels take effect.

Category

Specifies different formats from the following categories to display values on the chart.

- **Scale**
  Divides the value by hundreds, thousands, and so on.
- **Number**
  Re-formats the number value (Original example: 123456).
- **Date/Time**
  Re-formats the date/time value (Example, Wednesday, December 25, 00:00:00 GMT-08:00 2002).
- **Text**
  Specifies the length for the value.
- **Mapping**
  Maps the new value to one or more values.

Format

Lists all the formats of the selected category. For details about each format, see Data Format.

Properties

Specifies the properties for the selected format.

Sample

Displays the selected format effects.

Stack

Lists all the formats that you selected from different categories.

Add

Adds a format to the Stack box.

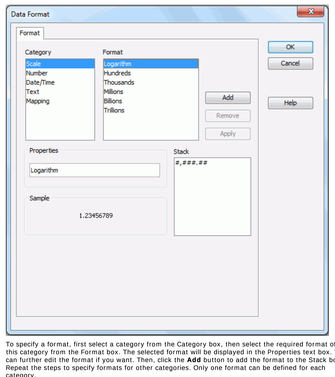
Remove

Removes a format from the Stack box.

Apply

Applies the specified format to values on the chart.

When you open the dialog to specify the data format for objects other than the category axis (the X axis) in a chart, the options in the dialog are as follows.



To specify a format, first select a category from the Category box, then select the required format of this category from the Format box. The selected format will be displayed in the Properties text box. You can further edit the format if you want. Then, click the **Add** button to add the format to the Stack box. Repeat the steps to specify formats for other categories. Only one format can be defined for each category.

The following are details about options in the dialog:

Category

Specifies different formats from the following categories to display values on the chart.

- **Scale**
  Divides the value by hundreds, thousands, and so on.
- **Number**
  Re-formats the number value (Original example: 123456).
- **Date/Time**
  Re-formats the date/time value (Example, Wednesday, December 25, 00:00:00 GMT-08:00 2002).
- **Text**
  Specifies the length for the value.
- **Mapping**
  Maps the new value to one or more values.

Format

Lists all the formats of the selected category. For details about each format, see Data Format.

Properties

Specifies the properties for the selected format.

Sample

Displays the selected format effects.

Stack

Lists all the formats that you selected from different categories.

Add

Adds a format to the Stack box.

Remove

Removes a format from the Stack box.

Apply

Applies the specified format to values on the chart.

OK

Applies the changes and closes the dialog.

Cancel

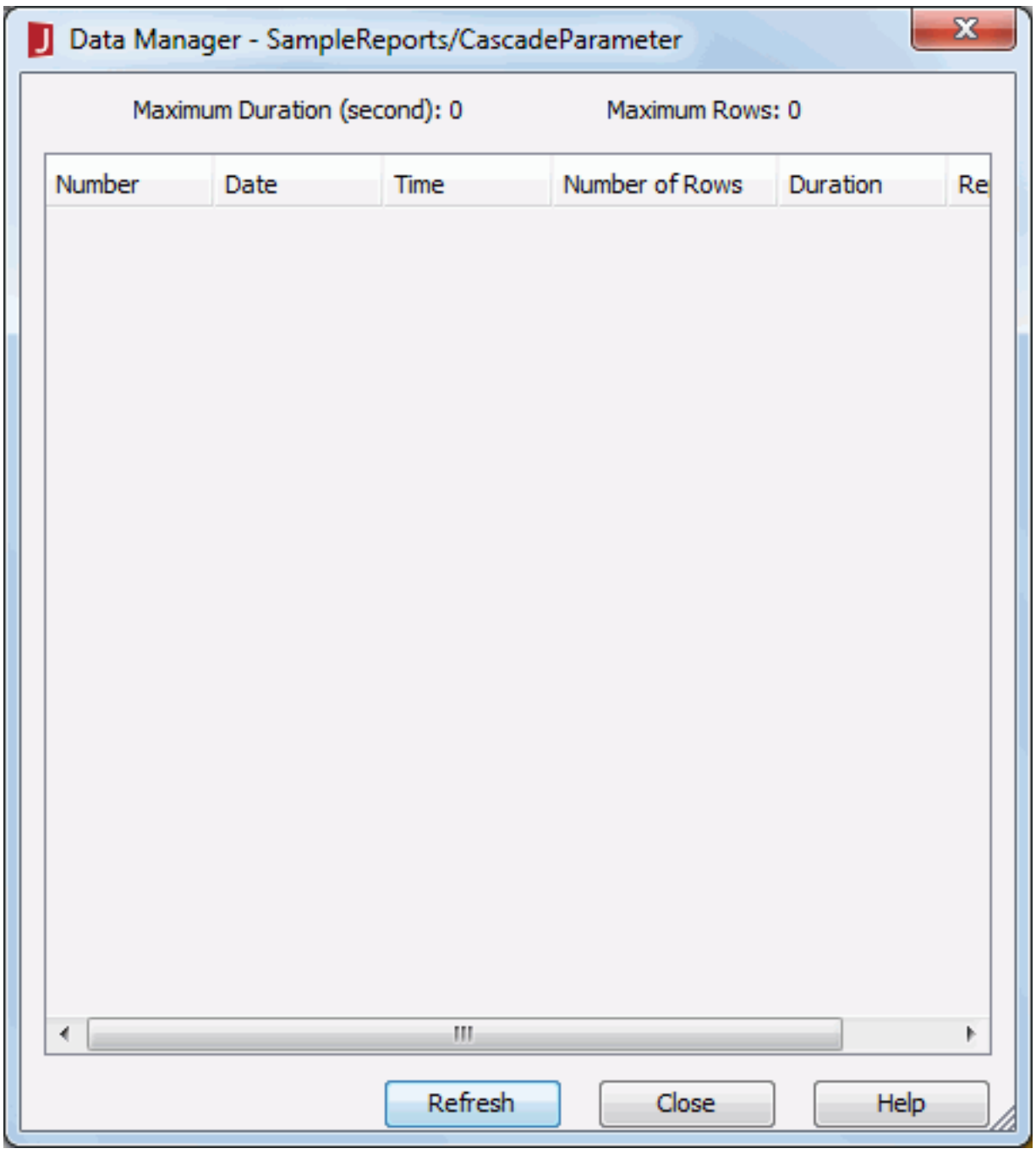
Does not retain any changes and closes the dialog.

Help

Displays the help document about this feature.

Data Manager dialog

This dialog appears when you right-click a data object which functions as a query in the Catalog Browser and select Data Manager from the shortcut menu.



The following are details about options in this dialog:

Maximum Duration (second)

Displays the value of the Maximum Duration specified on the data object.

Maximum Rows

Displays the value of Maximum Rows specified on the data object.

Number

Displays the number of the data object running record listed in the Data Manager.

Date

Displays the day on which the report is run.

Time

Displays the time when the report is run.

Number of Rows

Displays the number of rows displayed in the report.

Duration

Displays the running duration of the report.

Report

Displays the name of the report that uses the data object.

Refresh

Refreshes the record displayed in the Data Manager.

Close

Closes the Data Manager dialog.

Help

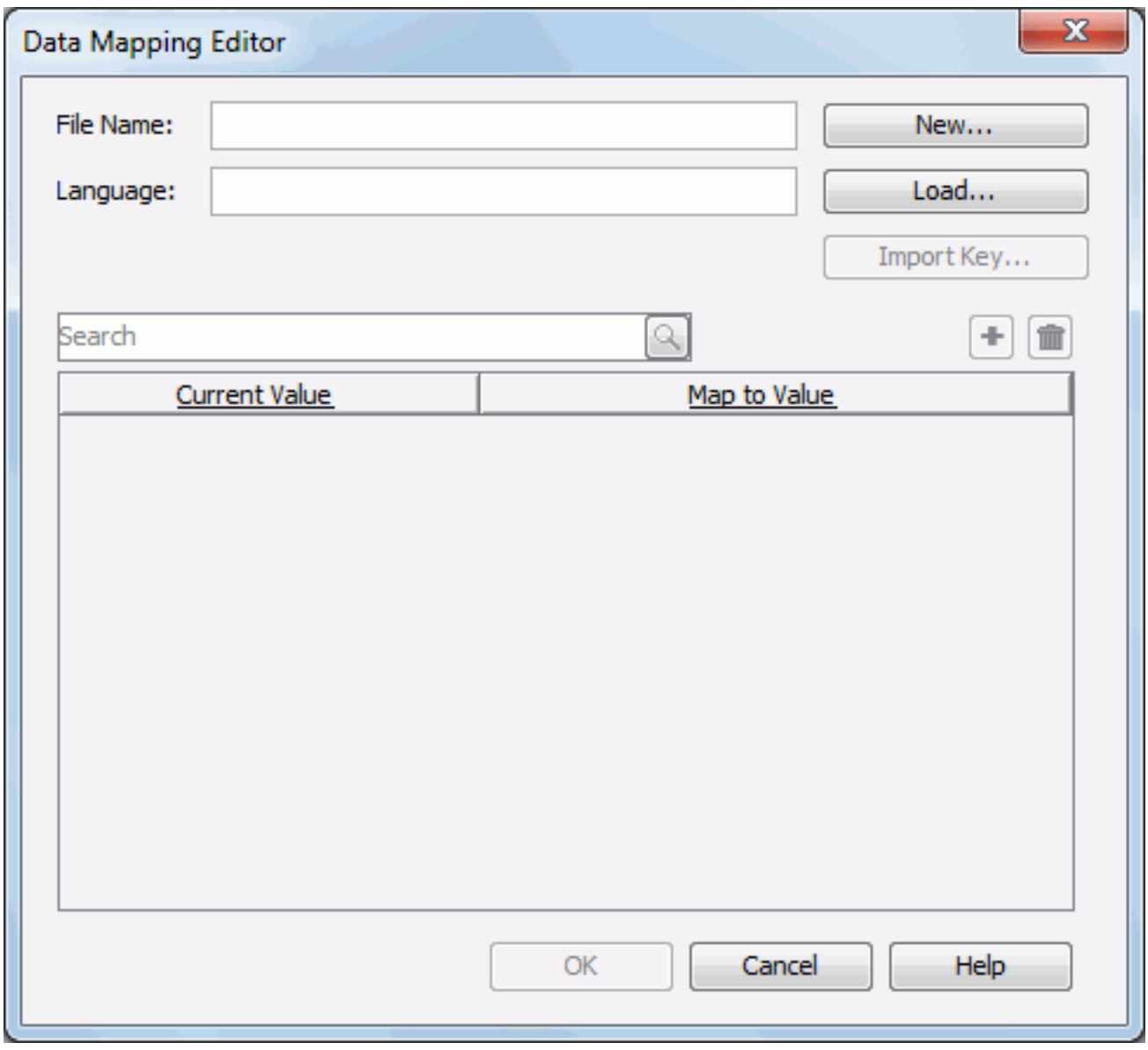
Displays the help document about this feature.

Related topics:

- Data Manager

Data Mapping Editor dialog

This dialog appears when you click the Data Mapping Editor button on the Catalog Browser toolbar. It helps you to edit information of a specific language saved in a data mapping file.



The following are details about options in this dialog.

File Name

Displays the file name of the current data mapping file that will be edited.

Language

Displays the language that is selected for the current data mapping file.

New

Opens the Create Data Mapping File dialog to generate a new data mapping file to save language information.

Load

Loads an existing data mapping file from a specific directory.

Import Key

Opens the Import Key dialog to add values into this data mapping file from catalog resources.

- **Select resource to be imported**
  Select one or more fields to be imported into the data mapping file.
- **OK**
  Imports the selected fields to the data mapping file and closes the dialog.
- **Cancel**
  Cancels importing resources and closes the dialog.

Quick search toolbar

Searches for the values that contain the specified text. After typing the text, click the search button on the toolbar or press Enter on the keyboard to start searching.

Adds a row to specify a value and its mapped value in another language.

Removes the selected value.

Current Value

Specifies the value that will be mapped according to the language of the data mapping file. You can click the column header to sort the items by current value in an ascending or descending order.

Map to Value

Specifies the value that the current value will be mapped to according to the language of the data mapping file. You can click the column header to sort the items by mapped value in an ascending or descending order.

OK

Applies the changes and closes the dialog.

Cancel

Cancels the changes and exits the dialog.

Help

Displays the help document about this feature.

Related topics:

- Mapping data values to different languages

Dataset Editor dialog

This dialog appears when you check the Existing radio button, select an existing dataset and click the Modify button in the Data screen of the report wizard. It helps you to edit the selected dataset and consists of the following tabs:

- Data tab
- Filter tab

OK

Applies the changes and closes the dialog.

Cancel

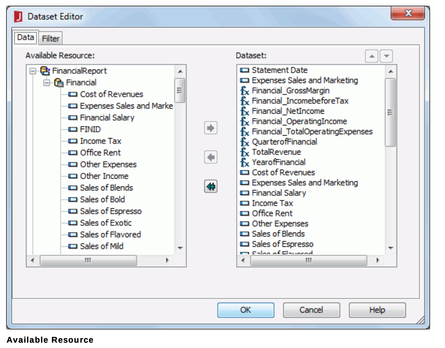
Cancels the changes and exits the dialog.

Help

Displays the help document about this feature.

Data tab

Adds some extra data fields to the dataset or removes some existing data fields from the dataset.



Available Resource

Lists all the available data fields contained in the data source on which the dataset is created.

Dataset

Lists all the fields that are included in the dataset.

Moves the select data field up a step.

Moves the select data field down a step.

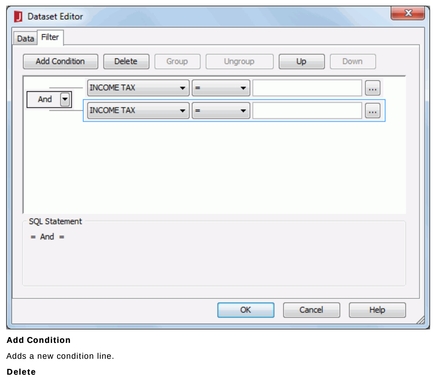
Adds the select data field to the dataset.

Removes the select data field from the dataset. Only the fields which are not used by any data component created on the dataset, either directly or indirectly, can be removed.

Removes all the data fields from the dataset. When you click the button, only the unused data fields will be removed actually. When you enter the dialog the next time, you will find that the data fields used by data components created on the dataset will still display in the Dataset box.

Filter tab

Sets some conditions to filter the data of the dataset.



Add Condition

Adds a new condition line.

Delete

Deletes the selected condition line.

Group

Makes the selected conditions in one group. Conditions can also be added to an existing group by selecting the conditions and the group while holding the Ctrl button, and then clicking the Group button.

Ungroup

Makes the selected condition ungrouped.

Up

Moves the selected condition up to a higher level.

Down

Moves the selected condition or group down to a lower level.

Logic

Lists the logic operator.

- **And**
  Logic operator And which is applied to this and the next line.
- **Or**
  Logic operator Or which is applied to this and the next line.
- **And Not**
  Logic operator And Not which is applied to this and the next line.
- **Or Not**
  Logic operator Or Not which is applied to this and the next line.

Field

Specifies the field to be filtered.

Operator

Specifies the operator to compose the filter expression.

- **=**
  Equal to
- **>**
  Greater than
- **>=**
  Greater than or equal to
- **<**
  Less than
- **<=**
  Less than or equal to
- **!=**
  Not equal to
- **[not]in**
  Causes an enumerated list of values which will appear in the WHERE clause predicate to be evaluated for a true condition.
- **[not] like**
  Like string pattern matching operator is used to compare the first expression string value to the pattern string (the second expression). If you want to use wildcard character in the pattern string, only "_" and "%" are supported.
- **[not] between**
  Allows the system to evaluate whether or not data values are located between a range of values indicated in the predicate. For operator 'between' or 'not between', there are two value input boxes for inputting the same type of values.
- **is [not] null**
  It is used in WHERE clause predicates to match null values occurring in a specified data field. For operator 'is null' or 'is not null', the value input box is hidden.

Value

Specifies the value of how to filter the field. You can either type in the value manually or click ... next to the text box to specify the value.

When you type in the value, if multiple values are required, they should be separated with ",", and if "," or "\" is contained in the values, write it as "\," or "\\".

When you click ... to specify the value, the Expressions dialog is displayed. The following are options in the dialog:

- **Real Name tab**
  Lists the tables added in the query on which the dataset is created with all their columns, also the formulas and parameters in the catalog. Double-click the required field to add it as the value.

  The following symbols at the bottom of the Expressions dialog allow you to modify the expression according to your requirements.

  - **+**
    Adds numbers or fields together.
  - **-**
    Subtracts numbers or fields together.
  - *****
    Multiplies numbers or fields together.
  - **/**
    Divides numbers or fields together.
  - **=**
    Equates fields together.
  - **"**
    Places quotations on long character strings or name that have blanks in them. For example, you should place quotes on values such as "New York" or "Washington DC".
  - **||**
    Places fields together in the same expression. For example, "New York" || "Washington DC".
  - **( )**
    Places the fields in parentheses.
- **Field Value tab**
  Lists all column names of the related table, in which the field to be filtered is. Check the column name of the field to be filtered, click OK, then the values of the field will be displayed. Choose the required value and click Set to use it to filter the field.

SQL Statement

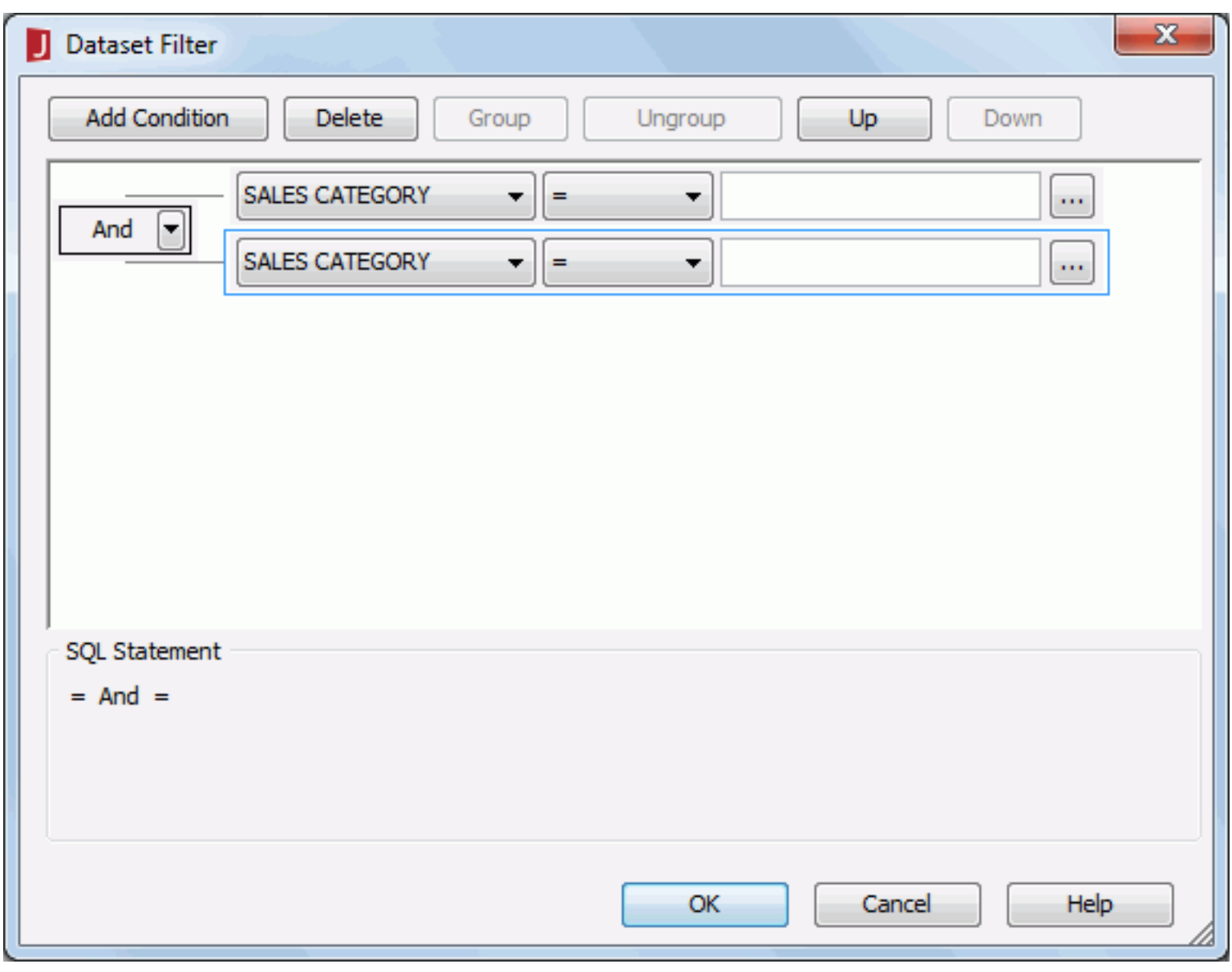
Displays the SQL statement of the filter condition.

Related topics:

- Datasets

Dataset Filter dialog

This dialog appears when you click the Dataset Filter button on the toolbar of the Resource View panel. It helps you to set filter conditions for a dataset.



The following are details about options in this dialog:

Add Condition

Adds a new condition line.

Delete

Deletes the selected condition line.

Group

Makes the selected conditions in one group. Conditions can also be added to an existing group by selecting the conditions and the group while holding the Ctrl button, and then clicking the Group button.

Ungroup

Makes the selected condition ungrouped.

Up

Moves the selected condition up to a higher level.

Down

Moves the selected condition or group down to a lower level.

Logic

Lists the logic operator.

- **And**
  Logic operator And which is applied to this and the next line.
- **Or**
  Logic operator Or which is applied to this and the next line.
- **And Not**
  Logic operator And Not which is applied to this and the next line.
- **Or Not**
  Logic operator Or Not which is applied to this and the next line.

Field

Specifies the field to be filtered.

Operator

Specifies the operator to compose the filter expression.

- **=**
  Equal to
- **>**
  Greater than
- **>=**
  Greater than or equal to
- **<**
  Less than
- **<=**
  Less than or equal to
- **!=**
  Not equal to
- **[not] in**
  Causes an enumerated list of values which will appear in the WHERE clause predicate to be evaluated for a true condition. The [not] in operator supports multi-value parameters as the value.
- **[not] like**
  Like string pattern matching operator is used to compare the first expression string value to the pattern string (the second expression). If you want to use wildcard character in the pattern string, only "_" and "%" are supported.
- **[not] between**
  Allows the system to evaluate whether or not data values are located between a range of values indicated in the predicate. For operator 'between' or 'not between', there are two value input boxes for inputting the same type of values.
- **is [not] null**
  It is used in WHERE clause predicates to match null values occurring in a specified data field. For operator 'is null' or 'is not null', the value input box is hidden.

Value

Specifies the value of how to filter the field. You can either type in the value manually or click ... next to the text box to specify the value.

When you type in the value, if multiple values are required, they should be separated with ",", and if "," or "\" is contained in the values, write it as "\," or "\\".

When you click ... to specify the value, the Expressions dialog is displayed. The following are options in the dialog:

- **Real Name tab**
  Lists the tables added in the query on which the dataset is created with all their columns, also the formulas and parameters in the catalog. Double-click the required field to add it as the value.

  The following symbols at the bottom of the Expressions dialog allow you to modify the expression according to your requirements.

  - **+**
    Adds numbers or fields together.
  - **-**
    Subtracts numbers or fields together.
  - *****
    Multiplies numbers or fields together.
  - **/**
    Divides numbers or fields together.
  - **=**
    Equates fields together.
  - **"**
    Places quotations on long character strings or name that have blanks in them. For example, you should place quotes on values such as "New York" or "Washington DC".
  - **||**
    Places fields together in the same expression. For example, "New York" || "Washington DC".
  - **( )**
    Places the fields in parentheses.
- **Field Value tab**
  Lists all column names of the related table, in which the field to be filtered is. Check the column name of the field to be filtered, click OK, then the values of the field will be displayed. Choose the required value and click Set to use it to filter the field.

SQL Statement

Displays the SQL statement of the filter condition.

OK

Applies the changes and closes the dialog.

Cancel

Cancels the changes and exits the dialog.

Help

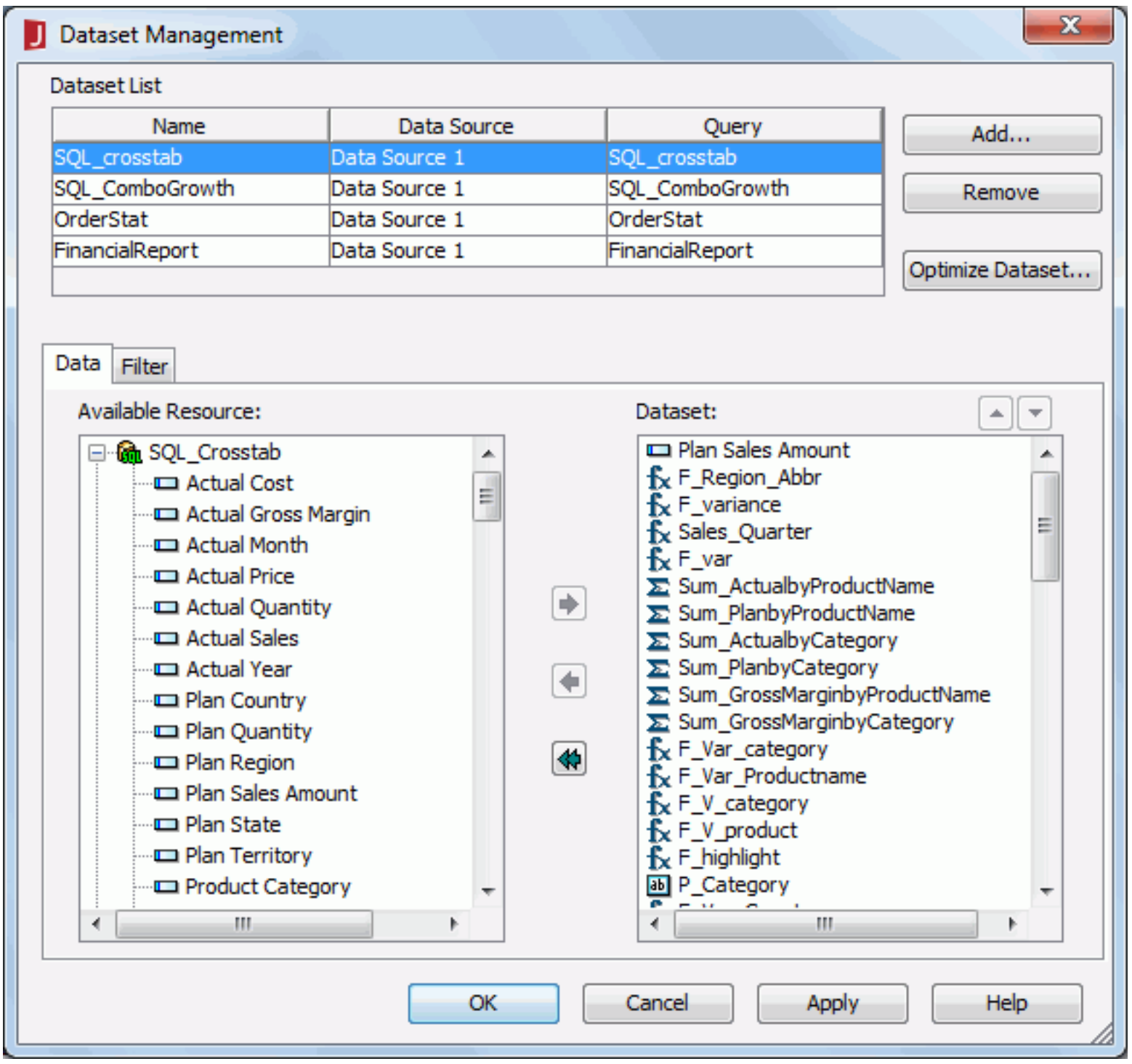
Displays the help document about this feature.

Related topics:

- Datasets

Dataset Management dialog

This dialog appears when you click Report > Dataset Management on the menu bar. It helps you to manage the datasets created for the open report.



The following are details about options in this dialog:

Dataset List

Lists all the datasets that you created for the open report.

- **Name**
  Shows the name of the dataset.
- **Data Source**
  Shows the data source in which the dataset is located.
- **Query**
  Shows the query that the dataset is based on.

Add

Opens the New Dataset dialog to add a new dataset to the report.

Remove

Removes the specified dataset from the report.

Optimize Dataset

Opens the Optimize Dataset dialog to optimize the selected dataset.

Data tab

Modifies fields in a specified dataset.

Filter tab

Sets filter conditions for data in a dataset. For details about options in the tab, see Dataset Filter dialog.

OK

Applies the changes and closes the dialog.

Cancel

Cancels the changes and exits the dialog.

Apply

Applies the changes and leaves the dialog open.

Help

Displays the help document about this feature.

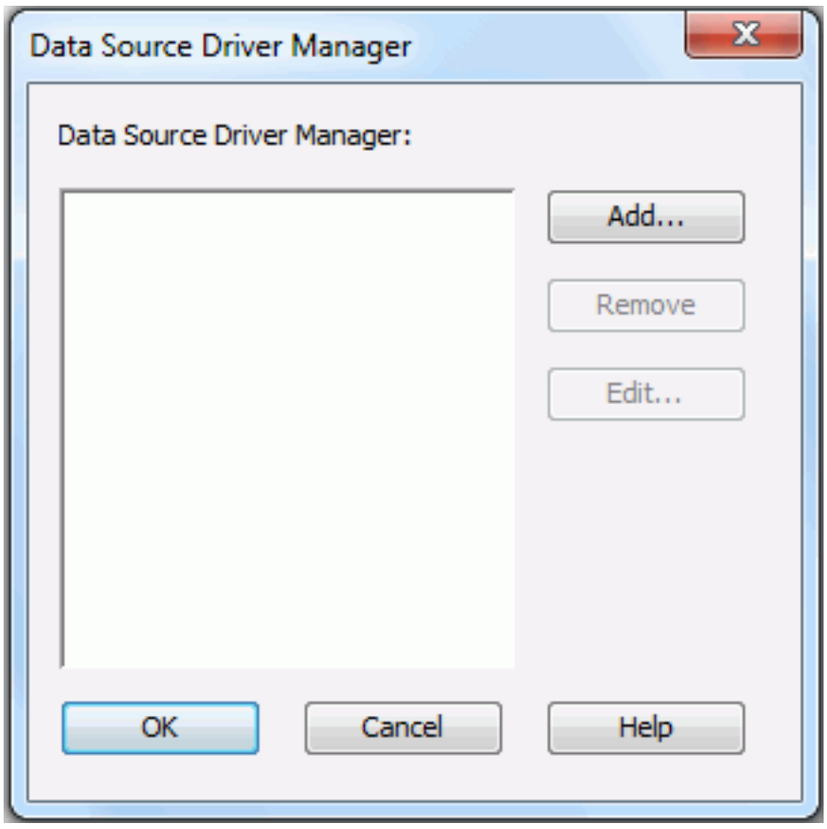
Related topics:

- Managing datasets

Data Source Driver Manager dialog

This dialog appears when you click the Data Source Driver Manager button on the Catalog Browser toolbar. It helps you to add or remove a user data source driver into or from a JReport catalog.

The following are details about options in this dialog.



Data Source Driver Manager

Lists all the available data source driver names.

Add

Opens the Data Source Editor dialog to add a new data source driver.

Remove

Removes the selected data source driver.

Edit

Edits the selected data source driver.

OK

Accepts changes and closes the dialog.

Cancel

Cancels the changes and closes the dialog.

Help

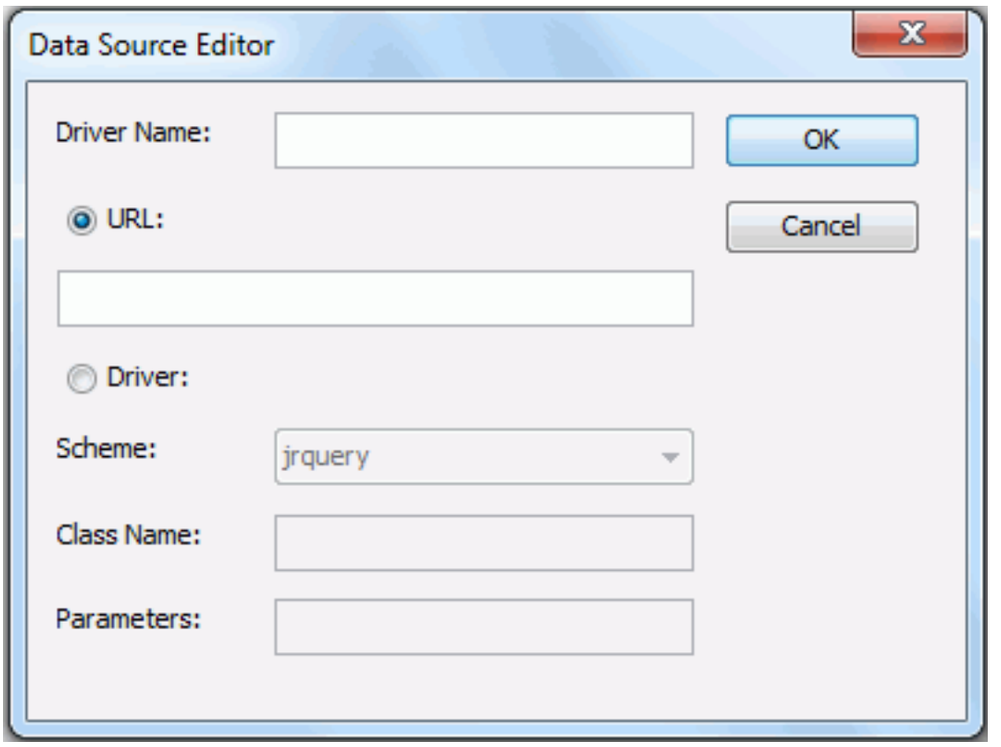
Displays the help document about this feature.

Related topics:

- Managing cached query results

Data Source Editor dialog

This dialog appears when you click the Add button, or select a data source driver and then click Edit in the Data Source Driver Manager dialog. It helps you to add a new data source driver or edit the specified data source driver in a JReport catalog.



The following are details about options in this dialog.

Driver Name

Specifies the name of the data source driver.

URL

Specifies the URL of the driver definition.

The URL is composed of three parts: scheme, qualified class name and arguments. If the Data Source Driver does not use argument, the argument part will not appear on the URL. Its syntax is: URL = "jrquery" ":" "/" qualified_classname[";" params]

Driver

Selects to specify the driver configuration.

- **Scheme**
  Specifies the scheme of the driver. Presently only "jrquery" is available.
- **Class Name**
  Specifies the qualified class name.
- **Parameters**
  Specifies the arguments for the Data Source Driver. In the parameters field, the arguments should be separated by semicolons (";").

OK

Accepts the changes and closes the dialog.

Cancel

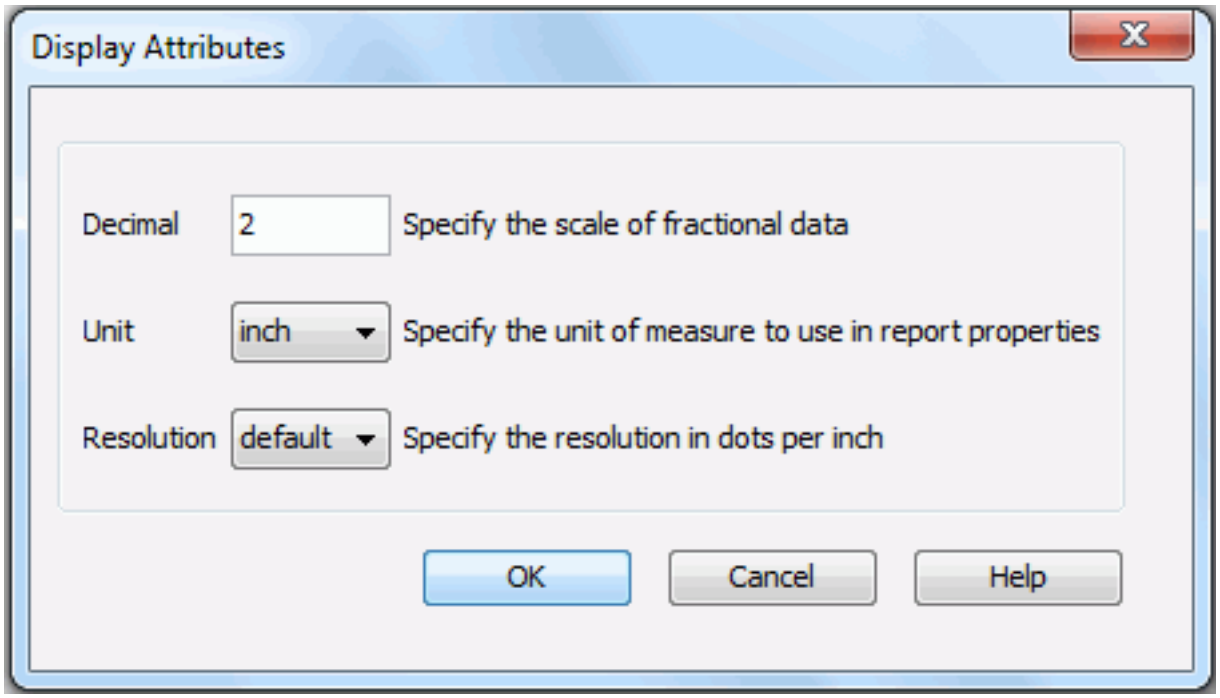
Does not retain changes and closes the dialog.

Related topics:

- Managing cached query results

Display Attributes dialog

This dialog appears when you click View > Display Attributes on the menu bar. It helps you to define certain attributes for the display values.



The following are details about options in this dialog.

Decimal

Specifies the scale of the fractional data in the report. For instance, if you set Decimal to 3, then all the property values of the objects in the Report Inspector will be rounded to the thousandth.

Unit

Specifies the unit of the measure to use in the report properties. It can either be centimeter or inch.

Resolution

Specifies the resolution in dots per inch. When it is Default, it will be the resolution of your monitor.

OK

Accepts all changes and closes the dialog.

Cancel

Does not retain any changes and closes the dialog.

Help

Displays the help document about this feature.

Display Type dialog

This dialog appears when you right-click a field and select Display Type from the shortcut menu. It helps you to define the display type of the field.

The following are all the display types that you can choose from:

- Text
- Image
- Barcode
- Rank
- Text Field
- Hidden Filed
- Text Area
- Checkbox
- Radio button
- Image Button
- Button
- List
- Drop-down List

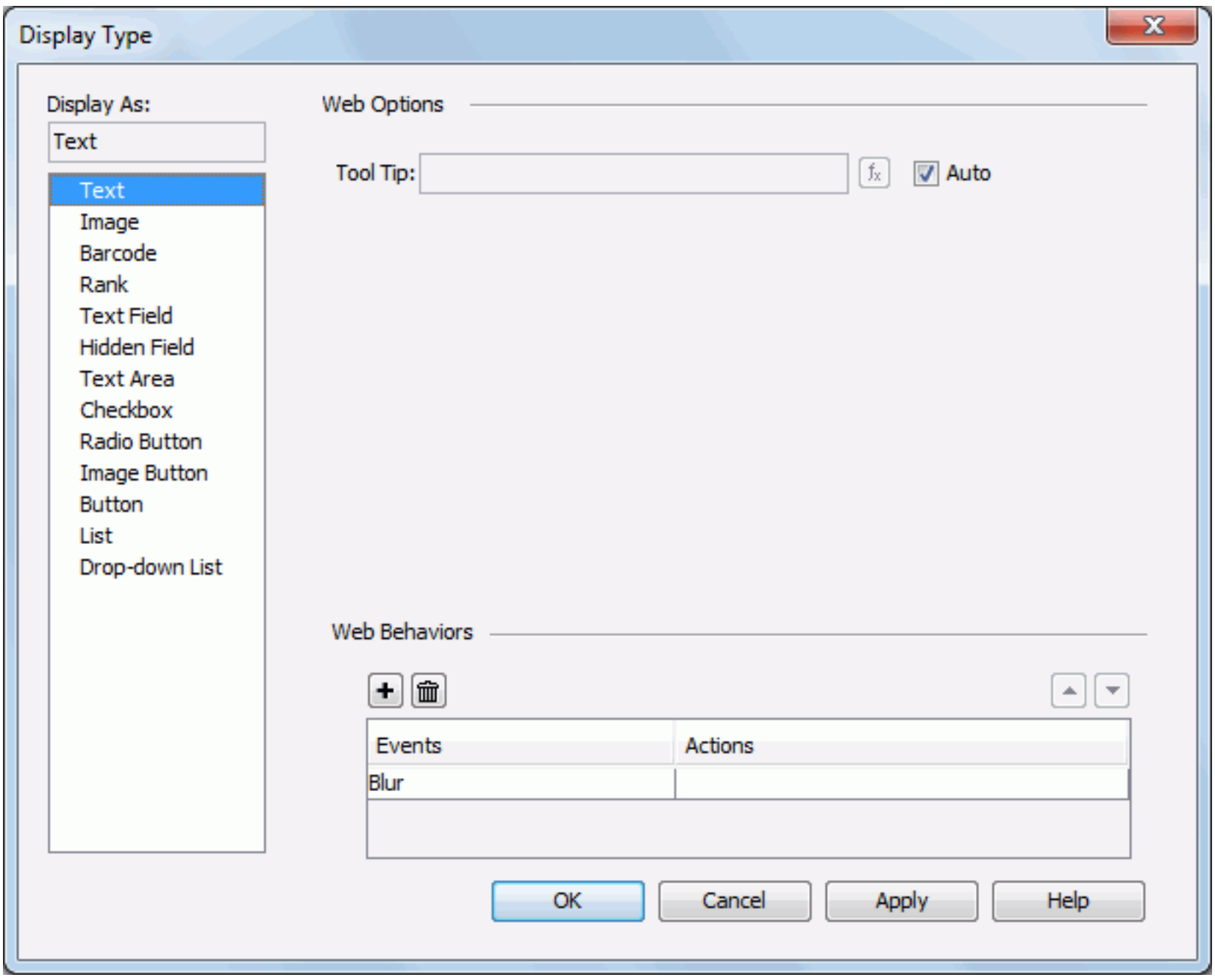
Note: The two display types - List and Drop-down List are only available when the field is a parameter field.

Related topics:

- Changing the display type of a label

Text

The following are details about options for this display type.



Tool Tip

Specifies the tooltip of the text, which will be shown when you hover the mouse over the text in HTML report result or at server runtime. If you want to use a formula to control the tooltip, click and select a formula from the drop-down list. If the given formulas cannot meet your requirements, click <CREATE FORMULA> to create one. If the text is in a crosstab, you can also use a crosstab formula created in the crosstab to control the tooltip, or click <CREATE CROSSTAB FORMULA> to create one.

- **Auto**
  When the Auto option is checked, the tooltip cannot be customized. Then at server runtime or in HTML results, when the text cannot be fully displayed, you can hover the mouse over the text to get its full content in the tooltip.

Web Behaviors

Specifies some web behaviors to the text.

- Adds a new web behavior line.
- Removes the selected web behavior.
- Moves the selected web behavior up a step.
- Moves the selected web behavior down a step.
- **Events**
  Specifies the trigger event.
- **Actions**
  Specifies the web action you want the event to trigger.
  - Opens the Web Action List dialog to bind some web action to the event.

OK

Applies the changes and closes the dialog.

Cancel

Does not retain any changes and closes the dialog.

Apply

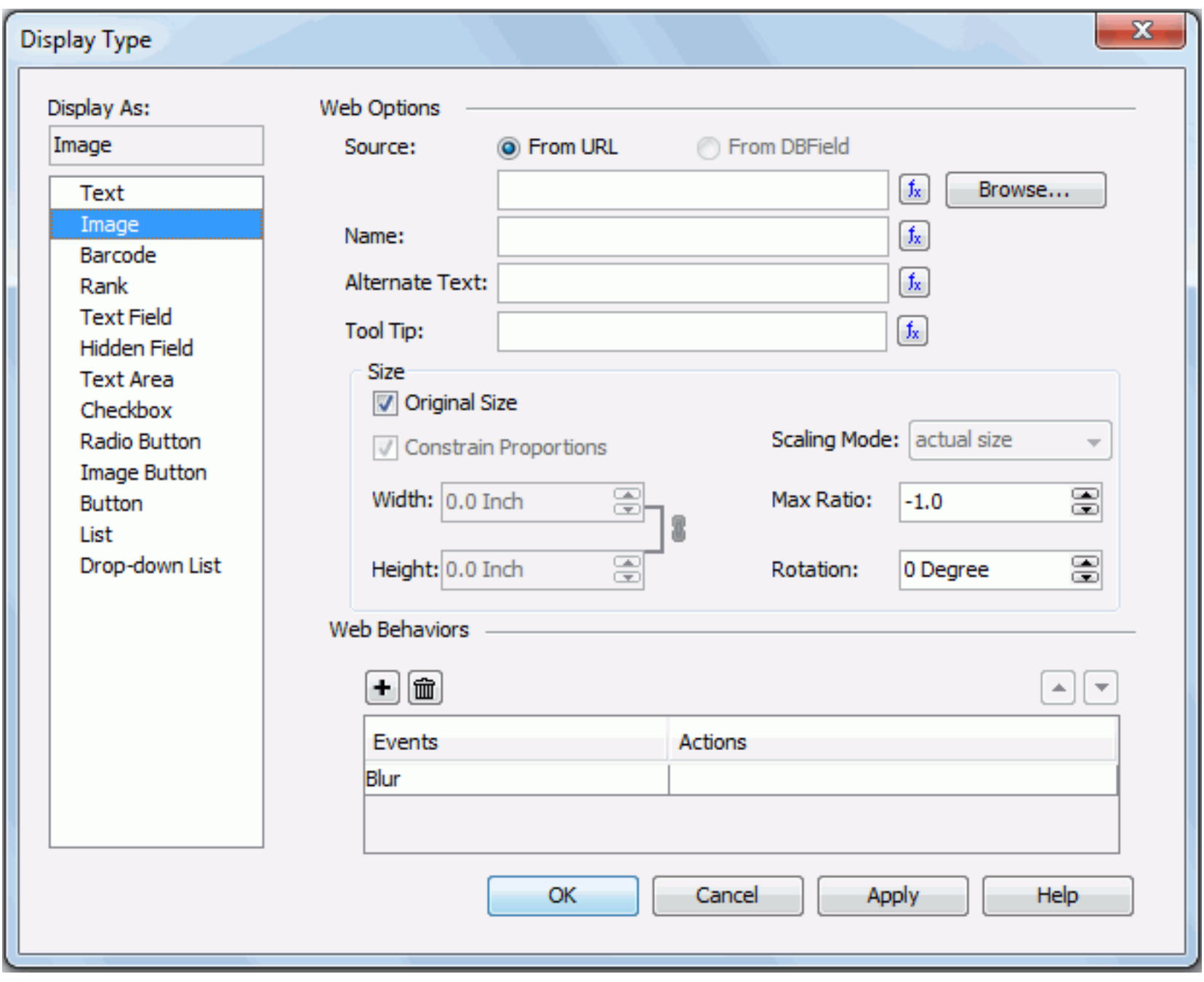
Applies all changes and leaves the dialog open.

Help

Displays the help document about this feature.

Image

The following are details about options for this display type.



Source

Specifies where the image you want to use comes from.

- **From URL**
  Specifies to use an image from a URL. You can click the Browse button to specify the path or input the URL directly in the text field. These types of images are supported: GIF, BMP, JPG, and PNG.
- **From DBField**
  For a DBField, formula, or a summary, you can check this option to make the value of the DBField/formula/summary the image source.
  - **Decode Type**
    Specifies the type for decoding the image. It can be GIF or JPG, BMP, PNG, AUTODETECT (this is to detect an image type saved in the database automatically).

Name

Specifies the name of the image. You can input the name directly in the text field or use a formula to control it.

Alternate Text

Specifies the alternate text of the image, which will be shown when the image cannot be displayed.

Tool Tip

Specifies the tooltip of the image, which will be shown when you hover the mouse over the image when the report runs in HTML format or in Page Report Studio.

Indicates the value of the option can be controlled by a formula. Click this button and select a formula from the drop-down list. If the given formulas cannot meet your requirements, click <CREATE FORMULA> to create one. If the image is in a crosstab, you can also use a crosstab formula created in the crosstab to control the tooltip, or click <CREATE CROSSTAB FORMULA> to create one.

Size

Specifies the size of the image.

- **Original Size**
  Specifies whether to use the original size of the image.
- **Constrain Proportions**
  If checked, when you set the display width or height of the image, the other will be adjusted accordingly based on the image's width-to-height ratio. Available only when the Original Size option is unchecked.
- **Width**
  Specifies the new display width of the image in inches. Available only when the Original Size option is unchecked.
- **Height**
  Specifies the new display height of the image in inches. Available only when the Original Size option is unchecked.
- **Scaling Mode**
  Specifies the scaling mode of the image. Available only when the Original Size option is unchecked. There are five types available for you:
  - **actual size**
    If selected, the image will be shown in its actual size.
  - **fit image**
    If selected, the image will be scaled to be wholly shown.
  - **fit width**
    If selected, the image will be scaled to fit the width of the image viewer.
  - **fit height**
    If selected, the image will be scaled to fit the height of the image viewer.
  - **customize**
    If selected, the image will be scaled according to the width and height that you specify in the Width and Height fields.
- **Max Ratio**
  Specifies the maximum scaling ratio of the image. By default the scaling ratio of the image is not limited. If it is set to any value greater than 0, the actual scaling ratio will be less than or equal to it.
- **Rotation**
  Specifies the angle, at which to rotate the image, in degrees. The following is the meaning of different values:
  - 0 - No rotation.
  - Positive value - Rotates the image clockwise.
  - Negative value - Rotates the image anticlockwise.

    Note: When rotating an image, the rectangle that holds the image maintains its original size, which may result in that the image exceeds the field border and therefore the parts that extend outside of the border will be cut off.

Web Behaviors

Specifies some web behaviors to the image.

- Adds a new web behavior line.
- Removes the selected web behavior.
- Moves the selected web behavior up a step.
- Moves the selected web behavior down a step.
- **Events**
  Specifies the trigger event.
- **Actions**
  Specifies the web action you want the event to trigger.
  - Opens the Web Action List dialog to bind some web action to the event.

OK

Applies the changes and closes the dialog.

Cancel

Does not retain any changes and closes the dialog.

Apply

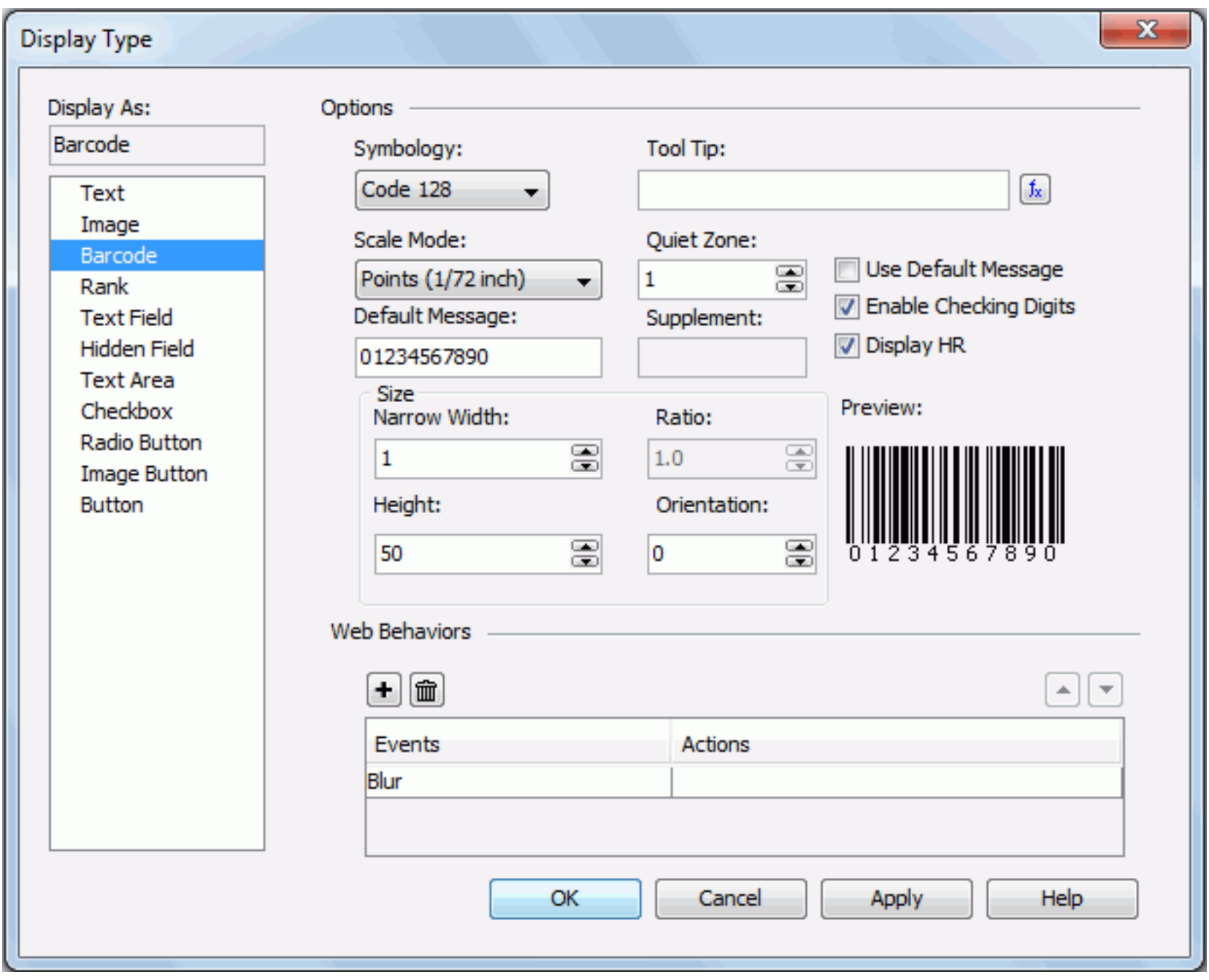
Applies all changes and leaves the dialog open.

Help

Displays the help document about this feature.

Barcode

The following are details about options for this display type.



Symbology

Specifies the type of the barcode. It could be UPC A/E, EAN 13/8, CODE 39, CODE 128, CODabar, and Code 128A/128B/128C.

Tool Tip

Specifies the tooltip of the barcode, which will be shown when you hover the mouse over the barcode when the report runs in HTML format or in Page Report Studio.

If you want to use a formula to control the tooltip, click and select a formula from the drop-down list. If the given formulas cannot meet your requirements, click <CREATE FORMULA> to create one. If the barcode is in a crosstab, you can also use a crosstab formula created in the crosstab to control the tooltip, or click <CREATE CROSSTAB FORMULA> to create one.

Scale Mode

Specifies the unit for the values of Quiet Zone, Narrow Width, Supplement, Height and Ratio.

Quiet Zone

Specifies the space around the barcode.

Default Message

Specifies the default value for the barcode, which will be used for the barcode in design mode.

Supplement

Specifies the supplement for the barcode. The barcode types, Code 39, Code 128/128A/128B/128C and Codabar, don't have supplements.

Use Default Massage

If checked, the value you specify in the Default Message field will be used as the barcode value when you view the report result. Otherwise, the value will only be used in design mode.

Enable Checking Digits

Specifies whether to include check digits in the barcode.

Display HR

Specifies whether to display the barcode numbers together with the barcode. Enabled only for Code 39, Code 128/128A/128B/128C and Codabar.

Size

Specifies the size of the barcode.

- **Narrow Width**
  Specifies the width between the barcode bars.
- **Ratio**
  Specifies the width ratio of the thick bar to the thin bar in the CODE39/Codabar barcode. The Ratio box can have only 2 effective values, 2.0 and 3.0, any ratio values are not equal to 2.0 or 3.0 will be granted as 2.0. Enabled only for CODE39 and Codabar.
- **Height**
  Specifies the height for the bars of the barcode.
- **Orientation**
  Specifies the orientation for the barcode.

Preview

Displays the format of the selected symbology.

Web Behaviors

Specifies some web behaviors to the barcode.

- Adds a new web behavior line.
- Removes the selected web behavior.
- Moves the selected web behavior up a step.
- Moves the selected web behavior down a step.
- **Events**
  Specifies the trigger event.
- **Actions**
  Specifies the web action you want the event to trigger.
  - Opens the Web Action List dialog to bind some web action to the event.

OK

Applies the changes and closes the dialog.

Cancel

Does not retain any changes and closes the dialog.

Apply

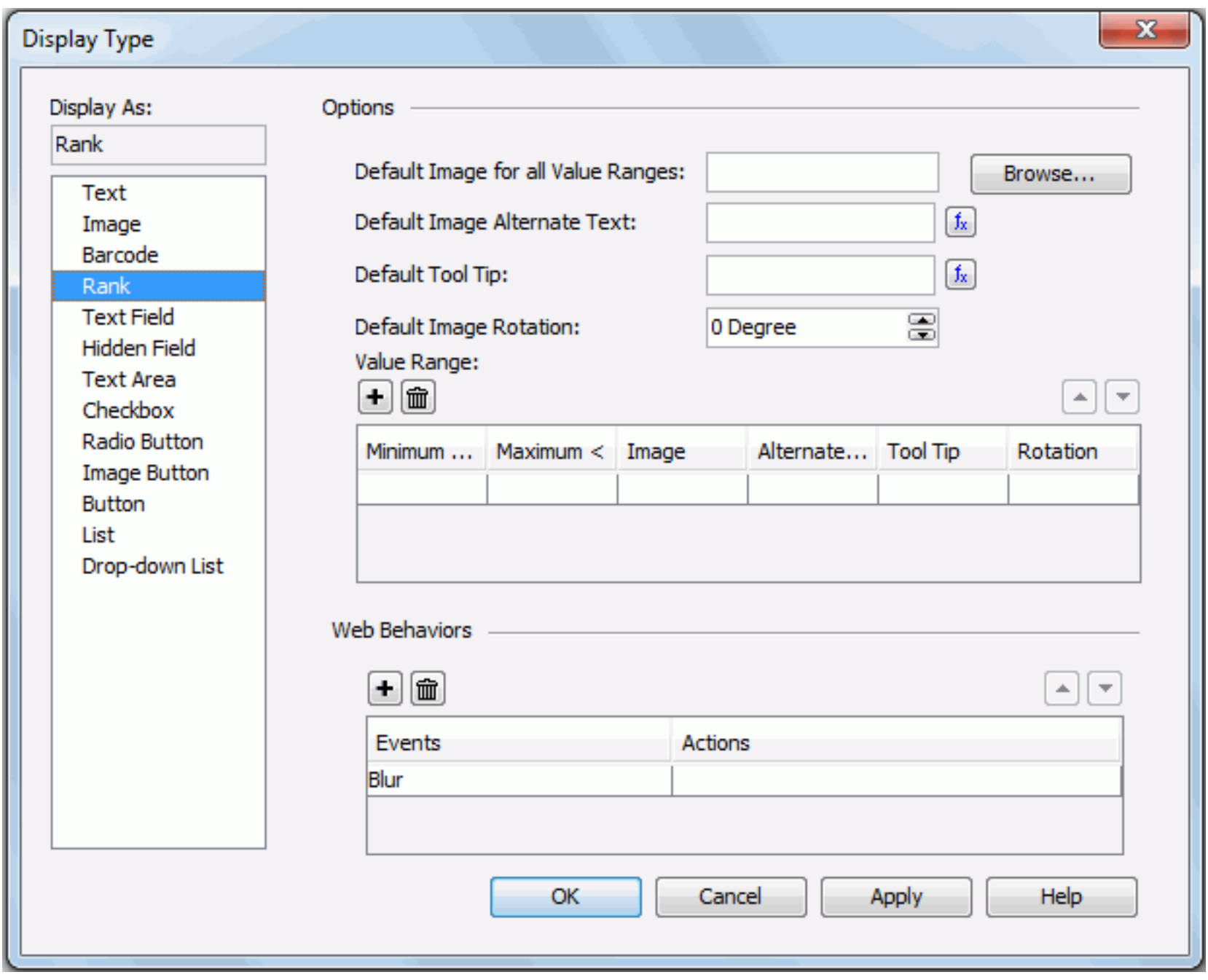
Applies all changes and leaves the dialog open.

Help

Displays the help document about this feature.

Rank

The following are details about options for this display type.



Default Image for All Value Ranges

Specifies the default image for all value ranges, which will be applied when the value of a field does not fall into any of the defined ranges. You can click the Browse button to specify the image path or input the path and file name of the image directly in the text field.

Default Image Alternate Text

Specifies the alternate text for the default image, which will be shown when the default image cannot be displayed.

Default Tool Tip

Specifies the tooltip of the default image, which will be shown when you hover the mouse over the image when the report runs in HTML format or in Page Report Studio.

Indicates the value of the option can be controlled by a formula. Click this button and select a formula from the drop-down list. If the given formulas cannot meet your requirements, click <CREATE FORMULA> to create one. If the rank is in a crosstab, you can also use a crosstab formula created in the crosstab to control the tooltip, or click <CREATE CROSSTAB FORMULA> to create one.

Default Image Rotation

Specifies the default image's rotation angle, in degrees. The following is the meaning of different values:

- 0 - No rotation.
- Positive value - Rotates the image clockwise.
- Negative value - Rotates the image anticlockwise.

Note: When rotating an image, the rectangle that holds the image maintains its original size, which may result in that the image exceeds the field border and therefore the parts that extend outside of the border will be cut off.

Value Range

Specifies the value ranges.

- Adds a new value range.
- Removes the selected value range.
- Moves the selected value range up a step.
- Moves the selected value range down a step.
- **Minimum (>=)**
  Specifies the minimum value for this range.
- **Maximum (<)**
  Specifies the maximum value for this range.
- **Image**
  Specifies the image of this value range, then when a field's value falls into this range, this image will be displayed.
- **Alternate Text**
  Specifies the alternate text for this value range's image, which will be shown when the image cannot be displayed.
- **Tool Tip**
  Specifies the tooltip of this value range's image, which will be shown when you hover the mouse over the image when the report runs in HTML format or in Page Report Studio.
- **Rotation**
  Specifies the rotation angle of this value range's image, in degrees.

Web Behaviors

Specifies some web behaviors to the rank.

- Adds a new web behavior line.
- Removes the selected web behavior.
- Moves the selected web behavior up a step.
- Moves the selected web behavior down a step.
- **Events**
  Specifies the trigger event.
- **Actions**
  Specifies the web action you want the event to trigger.
  - Opens the Web Action List dialog to bind some web action to the event.

OK

Applies the changes and closes the dialog.

Cancel

Does not retain any changes and closes the dialog.

Apply

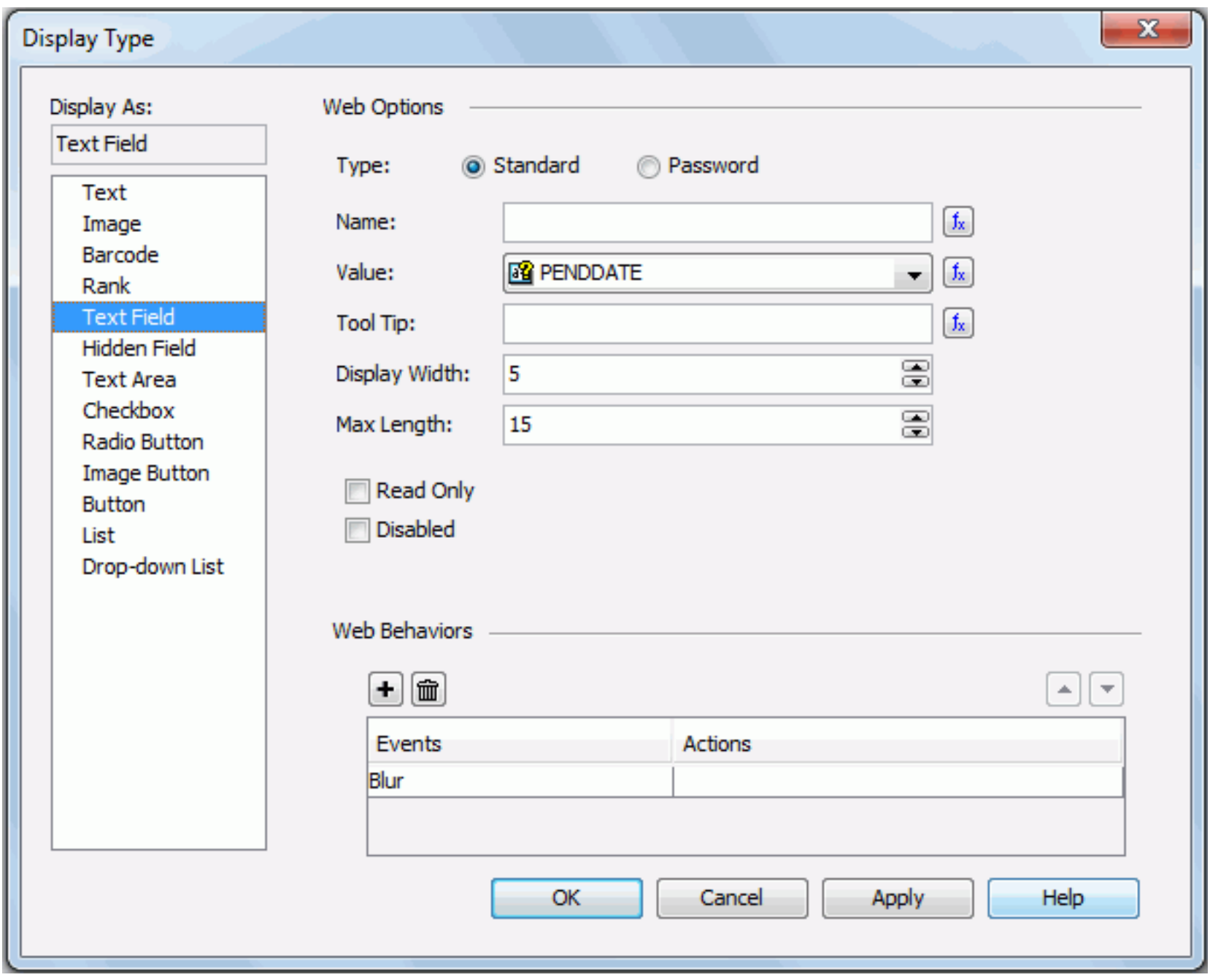
Applies all changes and leaves the dialog open.

Help

Displays the help document about this feature.

Text Field

The following are details about options for this display type.



Type

Specifies the type of the text field.

- **Standard**
  Specifies that the text field is a standard text field.
- **Password**
  Specifies that the text field is a password text field, which means that the value input in the text field will be hidden using asterisks.

Name

Specifies the name of the text field.

Value

Specifies the value of the text field.

Tool Tip

Specifies the tooltip of the text field, which will be shown when you hover the mouse over the text field when the report runs in HTML format or in Page Report Studio.

Indicates the value of the option can be controlled by a formula. Click this button and select a formula from the drop-down list. If the given formulas cannot meet your requirements, click <CREATE FORMULA> to create one. If the text field is in a crosstab, you can also use a crosstab formula created in the crosstab to control the tooltip, or click <CREATE CROSSTAB FORMULA> to create one.

Display Width

Specifies the display width of the text field.

Max Length

Specifies the maximum length of the string that is allowed in the text field.

Read Only

Specifies whether to make the field read-only to others.

Disabled

Specifies whether the text field is disabled or enabled.

Web Behaviors

Specifies some web behaviors to the text field.

- Adds a new web behavior line.
- Removes the selected web behavior.
- Moves the selected web behavior up a step.
- Moves the selected web behavior down a step.
- **Events**
  Specifies the trigger event.
- **Actions**
  Specifies the web action you want the event to trigger.
  - Opens the Web Action List dialog to bind some web action to the event.

OK

Applies the changes and closes the dialog.

Cancel

Does not retain any changes and closes the dialog.

Apply

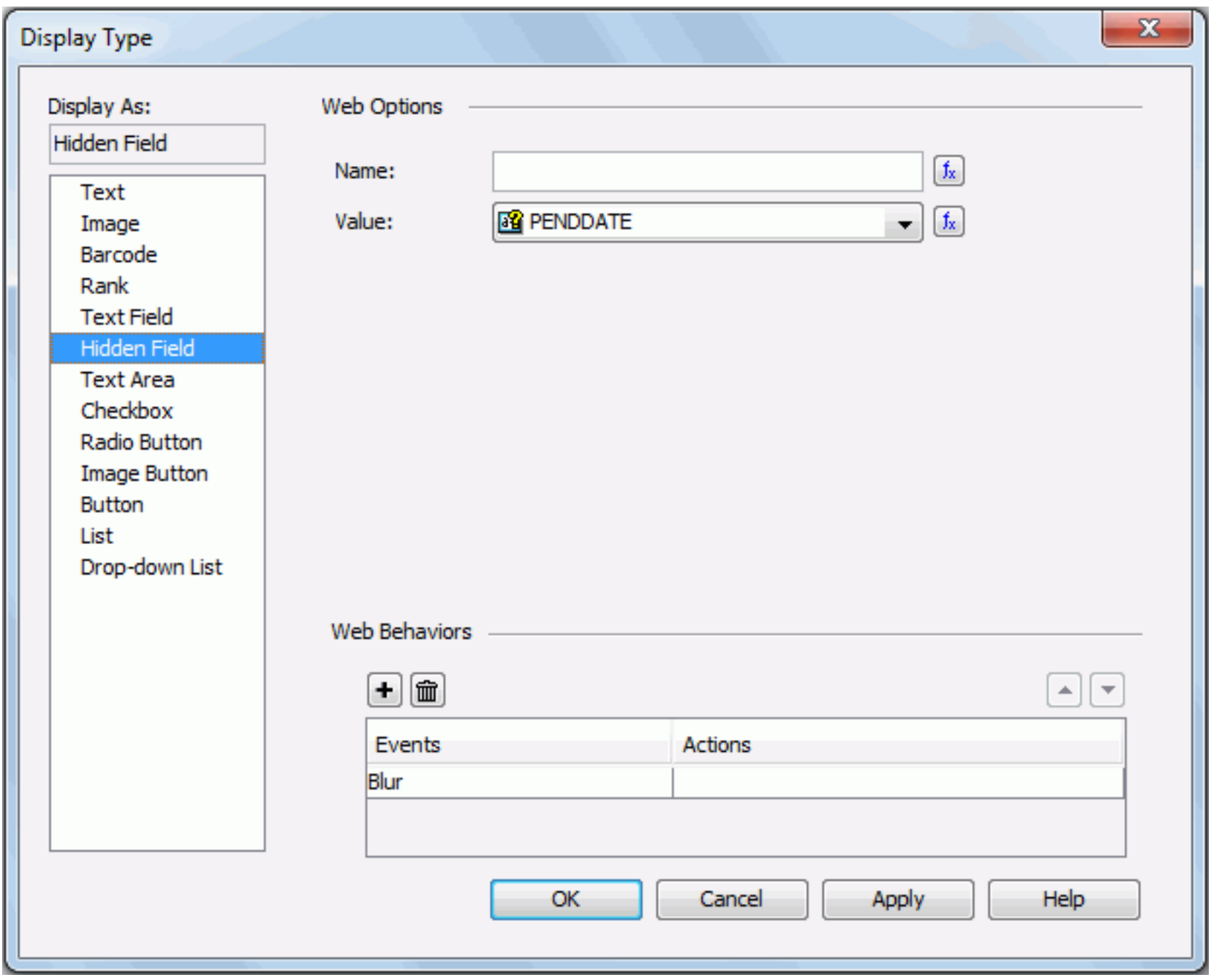
Applies all changes and leaves the dialog open.

Help

Displays the help document about this feature.

Hidden Field

The following are details about options for this display type.



Name

Specifies the name of the hidden field.

Value

Specifies the value of the hidden field.

Indicates the value of the option can be controlled by a formula. Click this button and select a formula from the drop-down list. If the given formulas cannot meet your requirements, click <CREATE FORMULA> to create one. If the hidden field is in a crosstab, you can also use a crosstab formula created in the crosstab to control the tooltip, or click <CREATE CROSSTAB FORMULA> to create one.

Web Behaviors

Specifies some web behaviors to the hidden field.

- Adds a new web behavior line.
- Removes the selected web behavior.
- Moves the selected web behavior up a step.
- Moves the selected web behavior down a step.
- **Events**
  Specifies the trigger event.
- **Actions**
  Specifies the web action you want the event to trigger.
  - Opens the Web Action List dialog to bind some web action to the event.

OK

Applies the changes and closes the dialog.

Cancel

Does not retain any changes and closes the dialog.

Apply

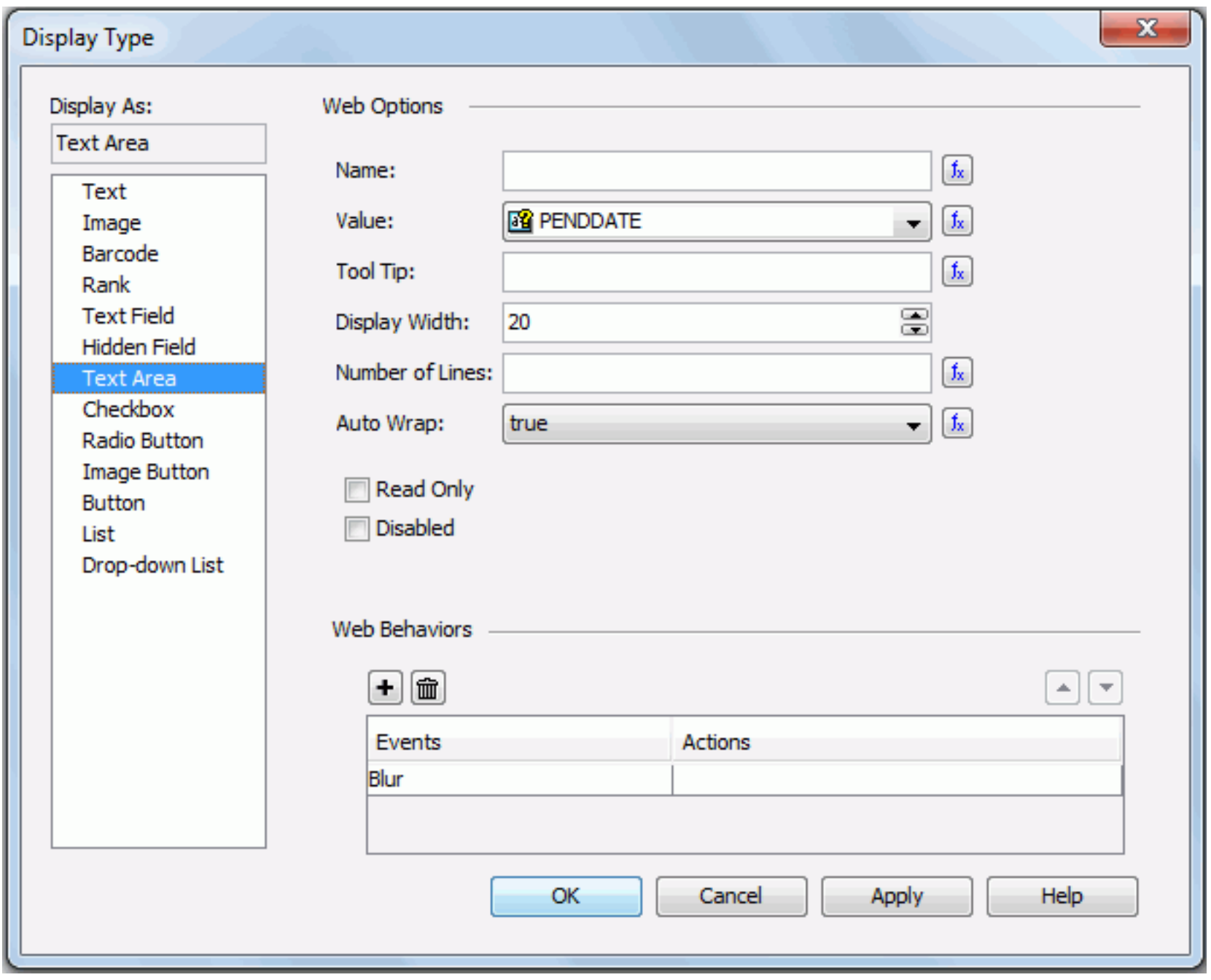
Applies all changes and leaves the dialog open.

Help

Displays the help document about this feature.

Text Area

The following are details about options for this display type.



Name

Specifies the name of the text area.

Value

Specifies the value of the text area.

Tool Tip

Specifies the tooltip of the text area, which will be shown when you hover the mouse over the text area when the report runs in HTML format or in Page Report Studio.

Display Width

Specifies the display width of the text area.

Number of Lines

Specifies the number of lines that is allowed in the text area.

Auto Wrap

Specifies whether to wrap the text in the text area.

Indicates the value of the option can be controlled by a formula. Click this button and select a formula from the drop-down list. If the given formulas cannot meet your requirements, click <CREATE FORMULA> to create one. If the text area is in a crosstab, you can also use a crosstab formula created in the crosstab to control the tooltip, or click <CREATE CROSSTAB FORMULA> to create one.

Read Only

Specifies whether to make the text area read-only to others.

Disabled

Specifies whether the text area is disabled or enabled.

Web Behaviors

Specifies some web behaviors to the text area.

- Adds a new web behavior line.
- Removes the selected web behavior.
- Moves the selected web behavior up a step.
- Moves the selected web behavior down a step.
- **Events**
  Specifies the trigger event.
- **Actions**
  Specifies the web action you want the event to trigger.
  - Opens the Web Action List dialog to bind some web action to the event.

OK

Applies the changes and closes the dialog.

Cancel

Does not retain any changes and closes the dialog.

Apply

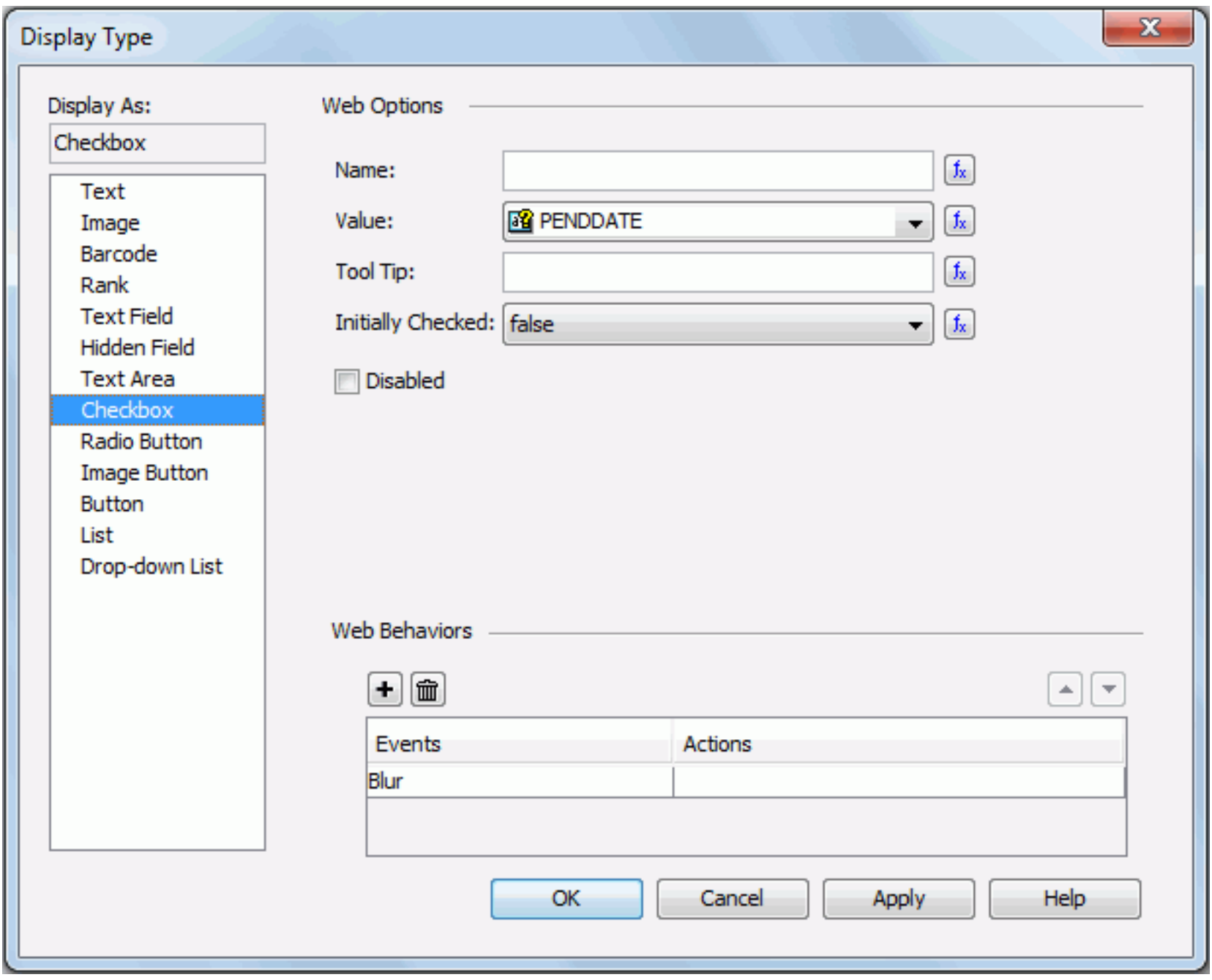
Applies all changes and leaves the dialog open.

Help

Displays the help document about this feature.

Checkbox

The following are details about options for this display type.



Name

Specifies the name of checkbox.

Value

Specifies the value of the checkbox.

Tool Tip

Specifies the tooltip of the checkbox, which will be shown when you hover the mouse over the checkbox when the report runs in HTML format or in Page Report Studio.

Initially Checked

Specifies whether or not the checkbox is selected by default.

Indicates the value of the option can be controlled by a formula. Click this button and select a formula from the drop-down list. If the given formulas cannot meet your requirements, click <CREATE FORMULA> to create one. If the checkbox is in a crosstab, you can also use a crosstab formula created in the crosstab to control the tooltip, or click <CREATE CROSSTAB FORMULA> to create one.

Disabled

Specifies whether the checkbox is enabled or disabled.

Web Behaviors

Specifies some web behaviors to the checkbox.

- Adds a new web behavior line.
- Removes the selected web behavior.
- Moves the selected web behavior up a step.
- Moves the selected web behavior down a step.
- **Events**
  Specifies the trigger event.
- **Actions**
  Specifies the web action you want the event to trigger.
  - Opens the Web Action List dialog to bind some web action to the event.

OK

Applies the changes and closes the dialog.

Cancel

Does not retain any changes and closes the dialog.

Apply

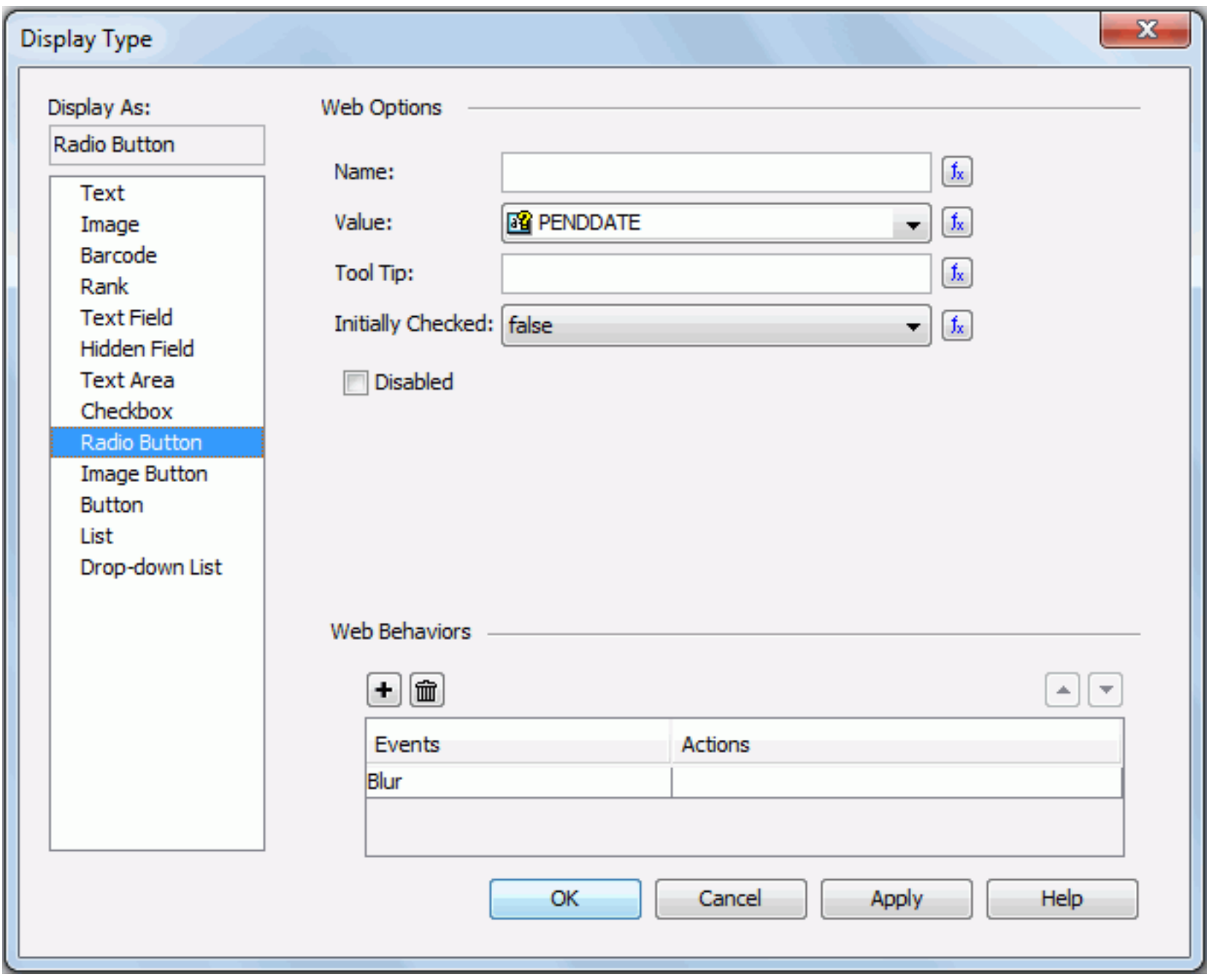
Applies all changes and leaves the dialog open.

Help

Displays the help document about this feature.

Radio Button

The following are details about options for this display type.



Name

Specifies the name of radio button.

Value

Specifies the value of the radio button.

Tool Tip

Specifies the tooltip of the radio button, which will be shown when you hover the mouse over the radio button when the report runs in HTML format or in Page Report Studio.

Initially Checked

Specifies whether or not the radio button is selected by default.

Indicates the value of the option can be controlled by a formula. Click this button and select a formula from the drop-down list. If the given formulas cannot meet your requirements, click <CREATE FORMULA> to create one. If the radio button is in a crosstab, you can also use a crosstab formula created in the crosstab to control the tooltip, or click <CREATE CROSSTAB FORMULA> to create one.

Disabled

Specifies whether the radio button is enabled or disabled.

Web Behaviors

Specifies some web behaviors to the radio button.

- Adds a new web behavior line.
- Removes the selected web behavior.
- Moves the selected web behavior up a step.
- Moves the selected web behavior down a step.
- **Events**
  Specifies the trigger event.
- **Actions**
  Specifies the web action you want the event to trigger.
  - Opens the Web Action List dialog to bind some web action to the event.

OK

Applies the changes and closes the dialog.

Cancel

Does not retain any changes and closes the dialog.

Apply

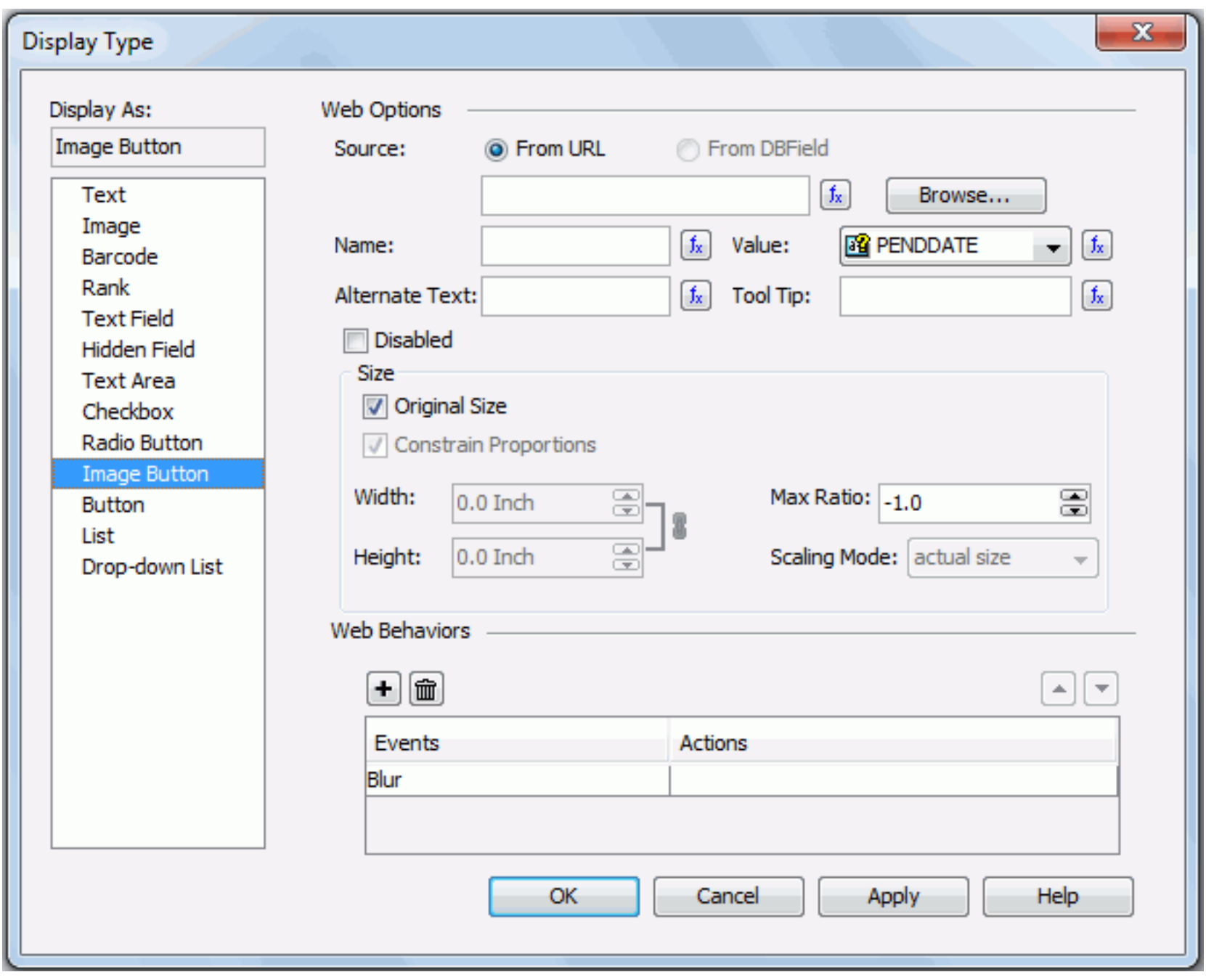
Applies all changes and leaves the dialog open.

Help

Displays the help document about this feature.

Image Button

The following are details about options for this display type.



Source

Specifies where the image button you want to use comes from.

- **From URL**
  Specifies to use an image from a URL. You can click the Browse button to specify the path or input the URL directly in the text field. These types of images are supported: GIF, BMP, JPG, and PNG.
- **From DBField**
  For a DBField, formula, or a summary, you can check this option to make the value of the DBField/formula/summary the image source.
  - **Decode Type**
    Specifies the type for decoding the image. It can be GIF or JPG, BITMAP, AUTODETECT (this is to detect an image type saved in the database automatically).

Name

Specifies the name of the image button.

Value

Specifies the value of the image button.

Alternate Text

Specifies the alternate text of the image, which will be shown when the image cannot be displayed.

Tool Tip

Specifies the tooltip of the image button, which will be shown when you hover the mouse over the image button when the report runs in HTML format or in Page Report Studio.

Indicates the value of the option can be controlled by a formula. Click this button and select a formula from the drop-down list. If the given formulas cannot meet your requirements, click <CREATE FORMULA> to create one. If the image button is in a crosstab, you can also use a crosstab formula created in the crosstab to control the tooltip, or click <CREATE CROSSTAB FORMULA> to create one.

Disabled

Specifies whether the image button is disabled or enabled.

Size

Specifies the size of the image.

- **Original Size**
  Specifies whether to use the original size of the image.
- **Constrain Proportions**
  If checked, when you set the width or height of the image, the other will be adjusted accordingly based on the image's width-to-height ratio. Available only when the Original Size option is unchecked.
- **Width**
  Specifies the new display width of the image in inches. Available only when the Original Size option is unchecked.
- **Height**
  Specifies the new display height of the image in inches. Available only when the Original Size option is unchecked.
- **Max Ratio**
  Specifies the maximum scaling ratio of the image. By default the scaling ratio of the image is not limited. If it is set to any value greater than 0, the actual scaling ratio will be less than or equal to it.
- **Scaling Mode**
  Specifies the scaling mode of the image. Available only when the Original Size option is unchecked. There are five types available for you:
  - **actual size**
    If selected, the image will be shown in its actual size.
  - **fit image**
    If selected, the image will be scaled to be wholly shown.
  - **fit width**
    If selected, the image will be scaled to fit the width of the image viewer.
  - **fit height**
    If selected, the image will be scaled to fit the height of the image viewer.
  - **customize**
    If selected, the image will be scaled according to the width and height that you specify in the Width and Height fields.

Web Behaviors

Specifies some web behaviors to the image button.

- Adds a new web behavior line.
- Removes the selected web behavior.
- Moves the selected web behavior up a step.
- Moves the selected web behavior down a step.
- **Events**
  Specifies the trigger event.
- **Actions**
  Specifies the web action you want the event to trigger.
  - Opens the Web Action List dialog to bind some web action to the event.

OK

Applies the changes and closes the dialog.

Cancel

Does not retain any changes and closes the dialog.

Apply

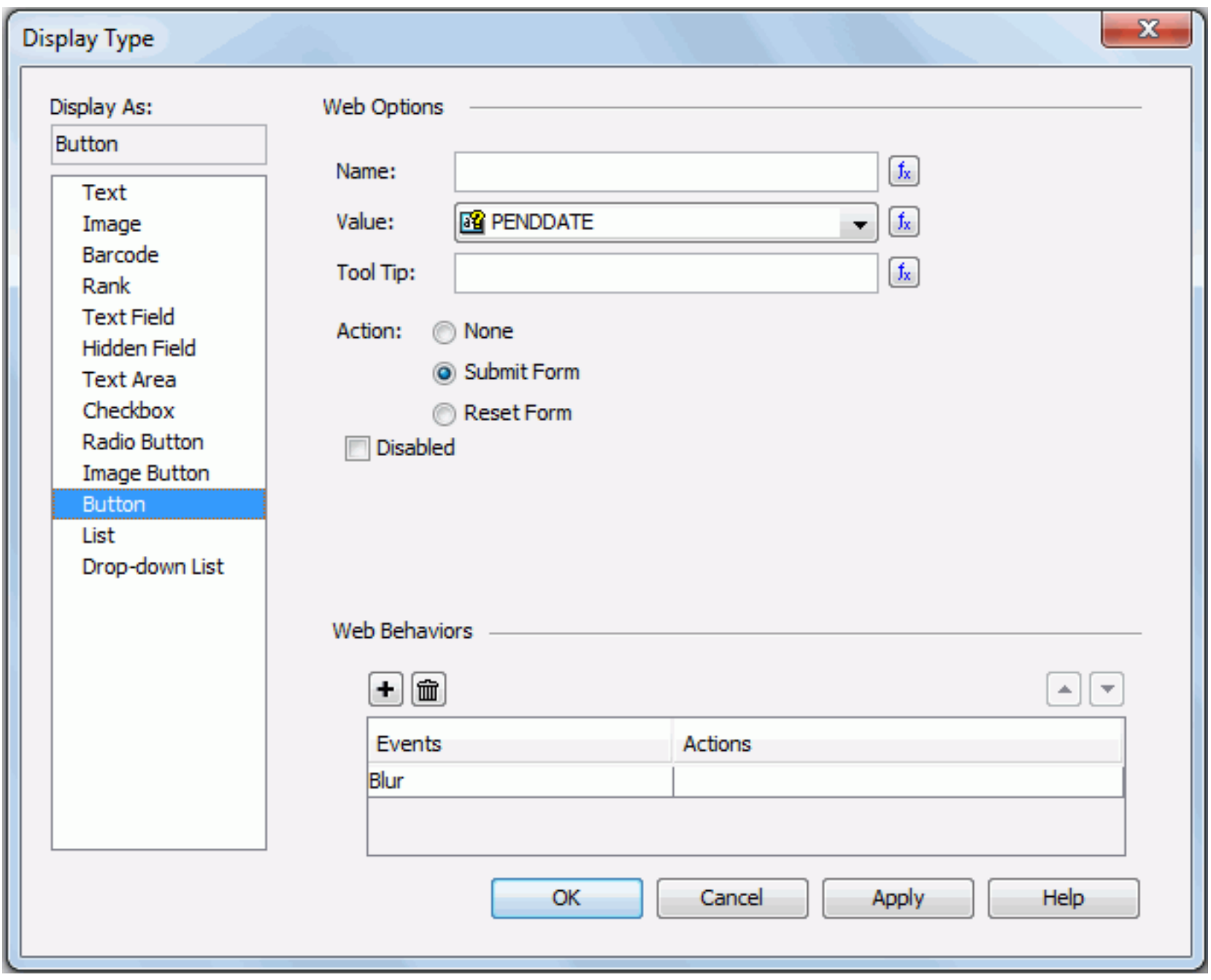
Applies all changes and leaves the dialog open.

Help

Displays the help document about this feature.

Button

The following are details about options for this display type.



Name

Specifies the name of the button.

Value

Specifies the value of the button.

Tool Tip

Specifies the tooltip of the button, which will be shown when you hover the mouse over the button when the report runs in HTML format or in Page Report Studio.

Indicates the value of the option can be controlled by a formula. Click this button and select a formula from the drop-down list. If the given formulas cannot meet your requirements, click <CREATE FORMULA> to create one. If the button is in a crosstab, you can also use a crosstab formula created in the crosstab to control the tooltip, or click <CREATE CROSSTAB FORMULA> to create one.

Action

Specifies the default action that will happen when the button is clicked.

- **None**
  Specifies that no default action happens when the button is clicked.
- **Submit Form**
  Specifies that the default action of the button is to submit a form.
- **Reset Form**
  Specifies that the default action of the button is to reset a form.

Disabled

Specifies whether the button is disabled or enabled.

Web Behaviors

Specifies some web behaviors to the button.

- Adds a new web behavior line.
- Removes the selected web behavior.
- Moves the selected web behavior up a step.
- Moves the selected web behavior down a step.
- **Events**
  Specifies the trigger event.
- **Actions**
  Specifies the web action you want the event to trigger.
  - Opens the Web Action List dialog to bind some web action to the event.

OK

Applies the changes and closes the dialog.

Cancel

Does not retain any changes and closes the dialog.

Apply

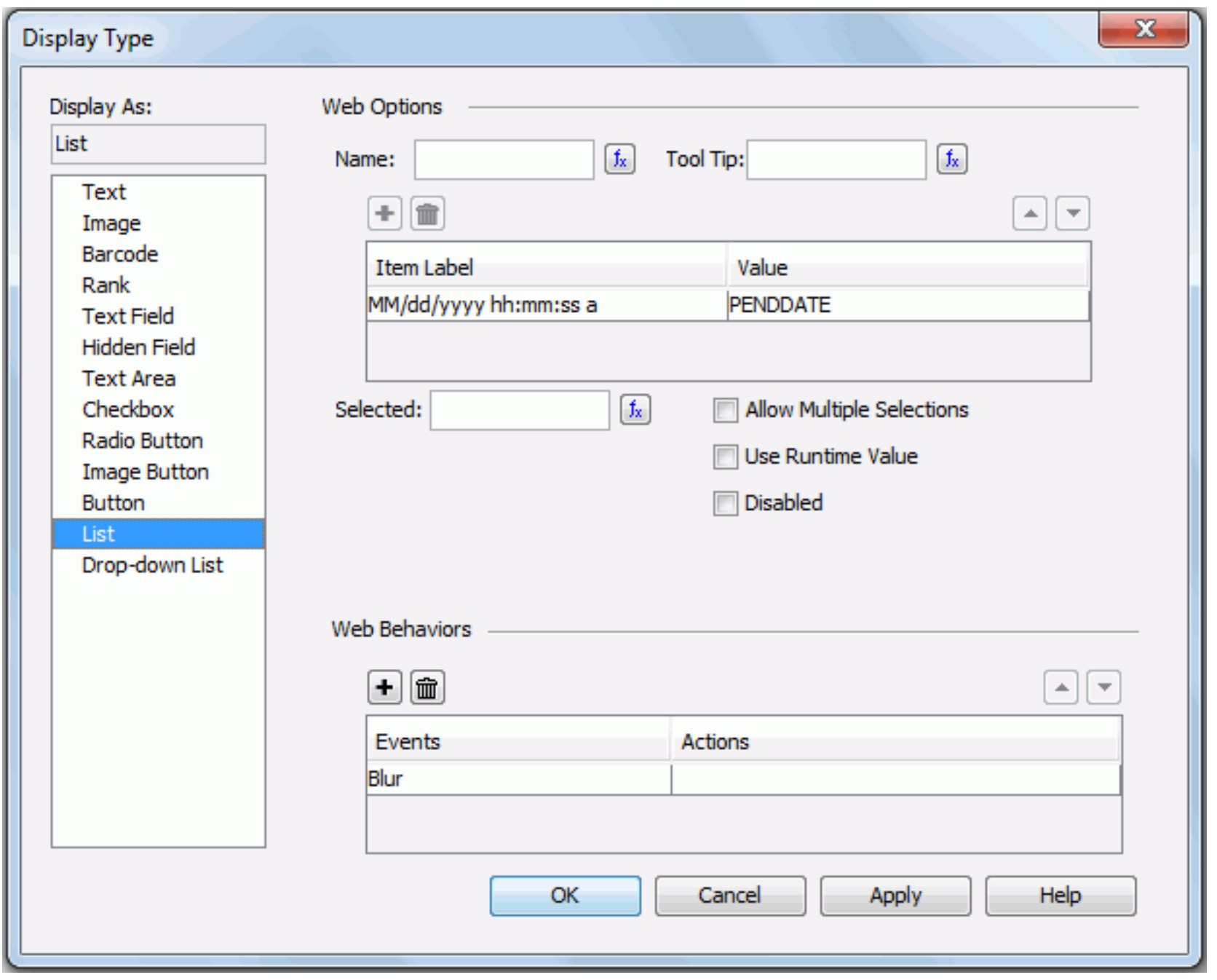
Applies all changes and leaves the dialog open.

Help

Displays the help document about this feature.

List

The following are details about options for this display type.



Name

Specifies the name of the list.

Tool Tip

Specifies the tooltip of the list, which will be shown when you hover the mouse over the list when the report runs in HTML format or in Page Report Studio.

Adds a new item in the list.

Removes the selected item.

Moves the selected item up a step.

Moves the selected item down a step.

Item Label

Specifies the display text or format of the item value. Select the desired one from the drop-down list.

Value

Specifies the item value.

- Opens the Insert Fields dialog to add field to control the value.

Selected

Specifies the value that will be selected in the list by default.

Indicates the value of the option can be controlled by a formula. Click this button and select a formula from the drop-down list. If the given formulas cannot meet your requirements, click <CREATE FORMULA> to create one.

Allow Multiple Selections

Specifies whether to allow selecting multiple items in the list at the same time.

Use Runtime Value

When changing the display type of a parameter to list, you can check this option to use the runtime value as the selected value of the list.

Disabled

Specifies whether the list is disabled or enabled.

Web Behaviors

Specifies some web behaviors to the list.

- Adds a new web behavior line.
- Removes the selected web behavior.
- Moves the selected web behavior up a step.
- Moves the selected web behavior down a step.
- **Events**
  Specifies the trigger event.
- **Actions**
  Specifies the web action you want the event to trigger.
  - Opens the Web Action List dialog to bind some web action to the event.

OK

Applies the changes and closes the dialog.

Cancel

Does not retain any changes and closes the dialog.

Apply

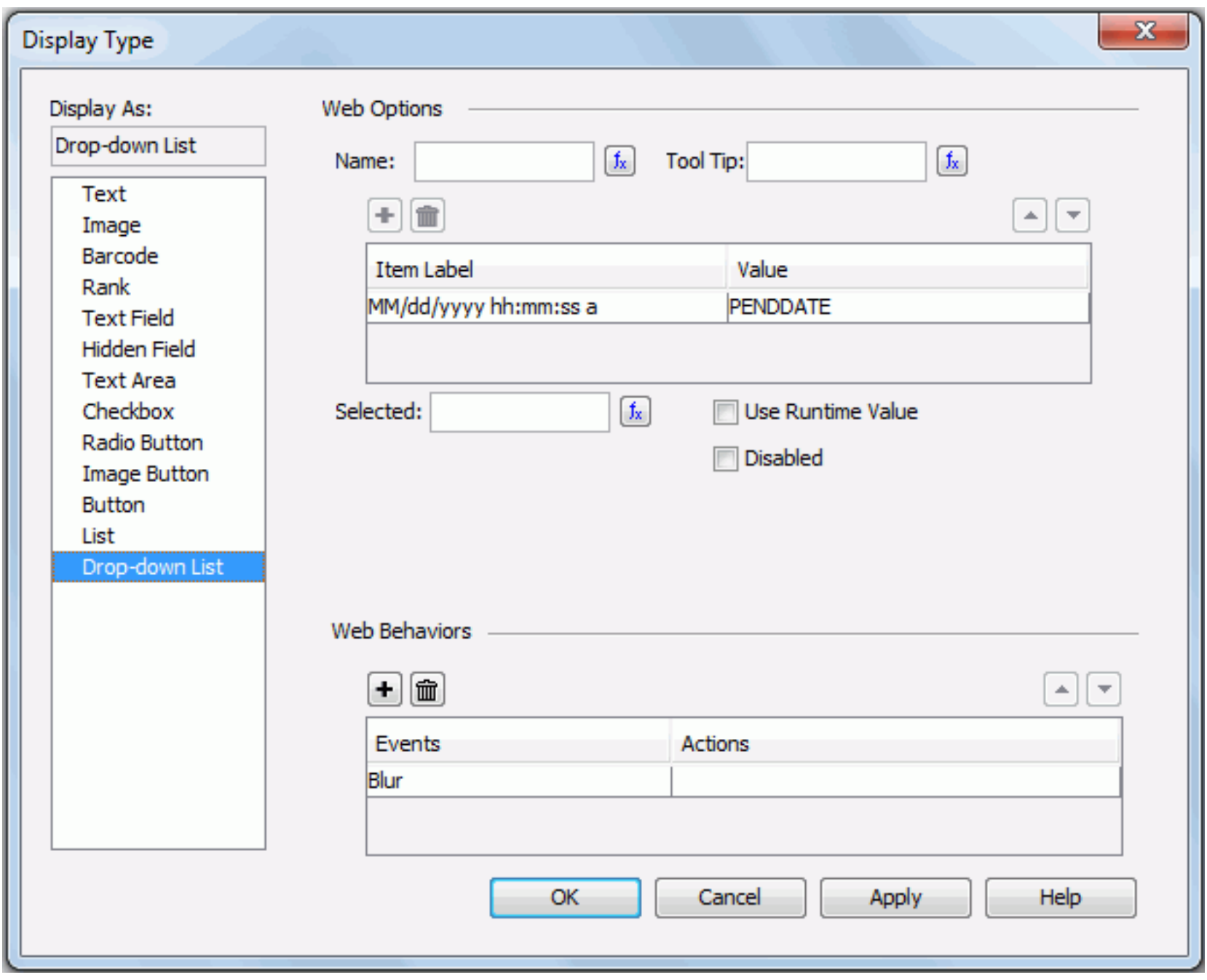
Applies all changes and leaves the dialog open.

Help

Displays the help document about this feature.

Drop-down List

The following are details about options for this display type.



Name

Specifies the name of the drop-down list.

Tool Tip

Specifies the tooltip of the drop-down list, which will be shown when you hover the mouse over the drop-down list when the report runs in HTML format or in Page Report Studio.

Adds a new item in the drop-down list.

Removes the selected item.

Moves the selected item up a step.

Moves the selected item down a step.

Item Label

Specifies the display text or format of the item value. Select the desired one from the drop-down list.

Value

Specifies the item value.

- Opens the Insert Fields dialog to add field to control the value.

Selected

Specifies the value that will be selected in the drop-down list by default.

Indicates the value of the option can be controlled by a formula. Click this button and select a formula from the drop-down list. If the given formulas cannot meet your requirements, click <CREATE FORMULA> to create one.

Use Runtime Value

When changing the display type of a parameter to drop-down list, you can check this option to use the runtime value as the selected value of the drop-down list.

Disabled

Specifies whether the drop-down list is disabled or enabled.

Web Behaviors

Specifies some web behaviors to the drop-down list.

- Adds a new web behavior line.
- Removes the selected web behavior.
- Moves the selected web behavior up a step.
- Moves the selected web behavior down a step.
- **Events**
  Specifies the trigger event.
- **Actions**
  Specifies the web action you want the event to trigger.
  - Opens the Web Action List dialog to bind some web action to the event.

OK

Applies the changes and closes the dialog.

Cancel

Does not retain any changes and closes the dialog.

Apply

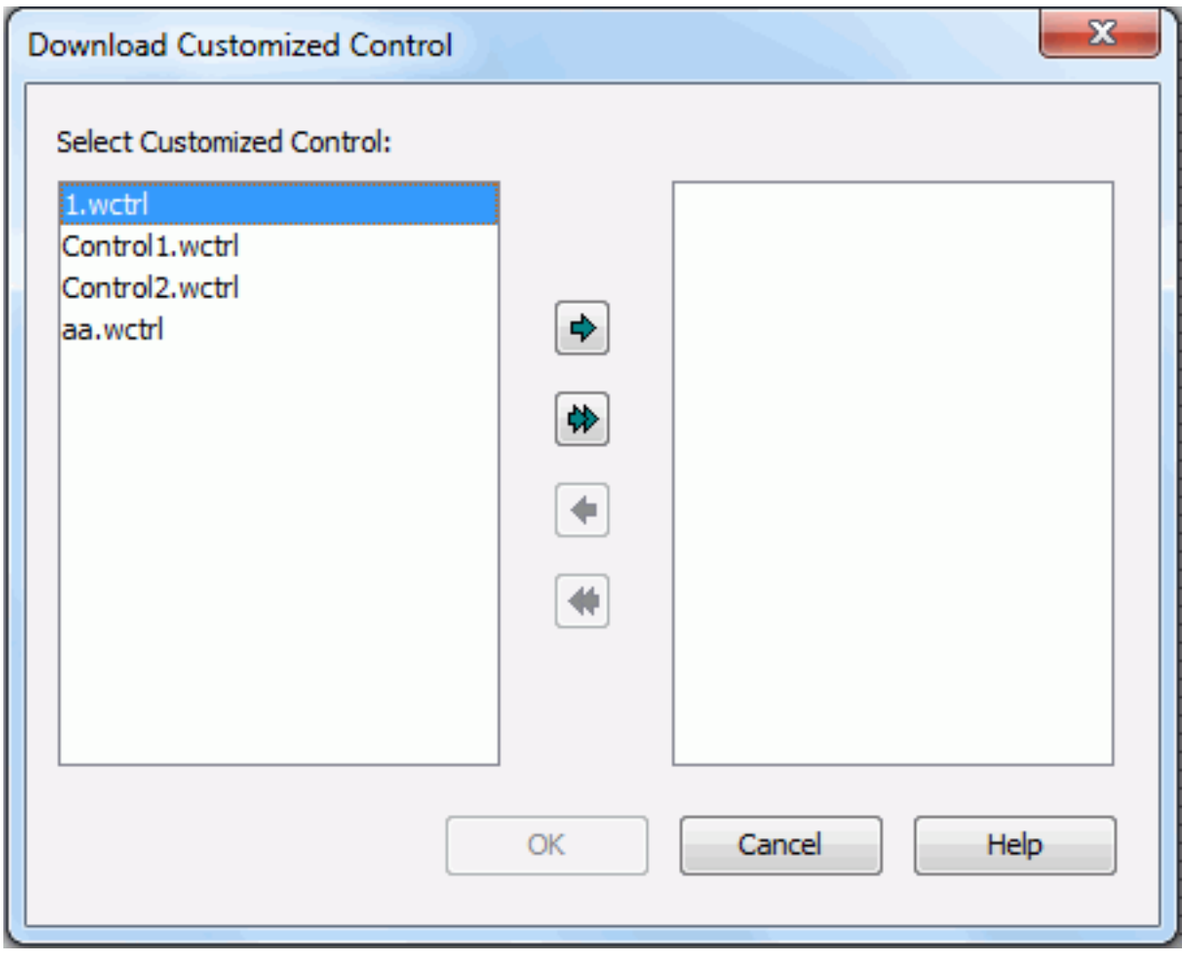
Applies all changes and leaves the dialog open.

Help

Displays the help document about this feature.

Download Customized Control dialog

This dialog is displayed when you click File > Publish and Download > Download from Server > Download Customized Control from JReport Server on the menu bar. It helps you to import customized control files from JReport Server.



All customized control files are listed for you to download.

Use the two right-headed-arrow buttons to add the selected customized control files or all of them from the left list to the right list. Use the two left-headed-arrow buttons to remove the selected customized control files or all of them from the right list.

OK

Exits the dialog and starts downloading. If a downloaded customized control file uses the same name as a customized control file at the local, the former will replace the latter directly without prompting. Make sure that you have properly backup the customized control files at the JReport Designer side before downloading from JReport Server.

Cancel

Cancels downloading customized control files from JReport Server and closes the dialog.

Help

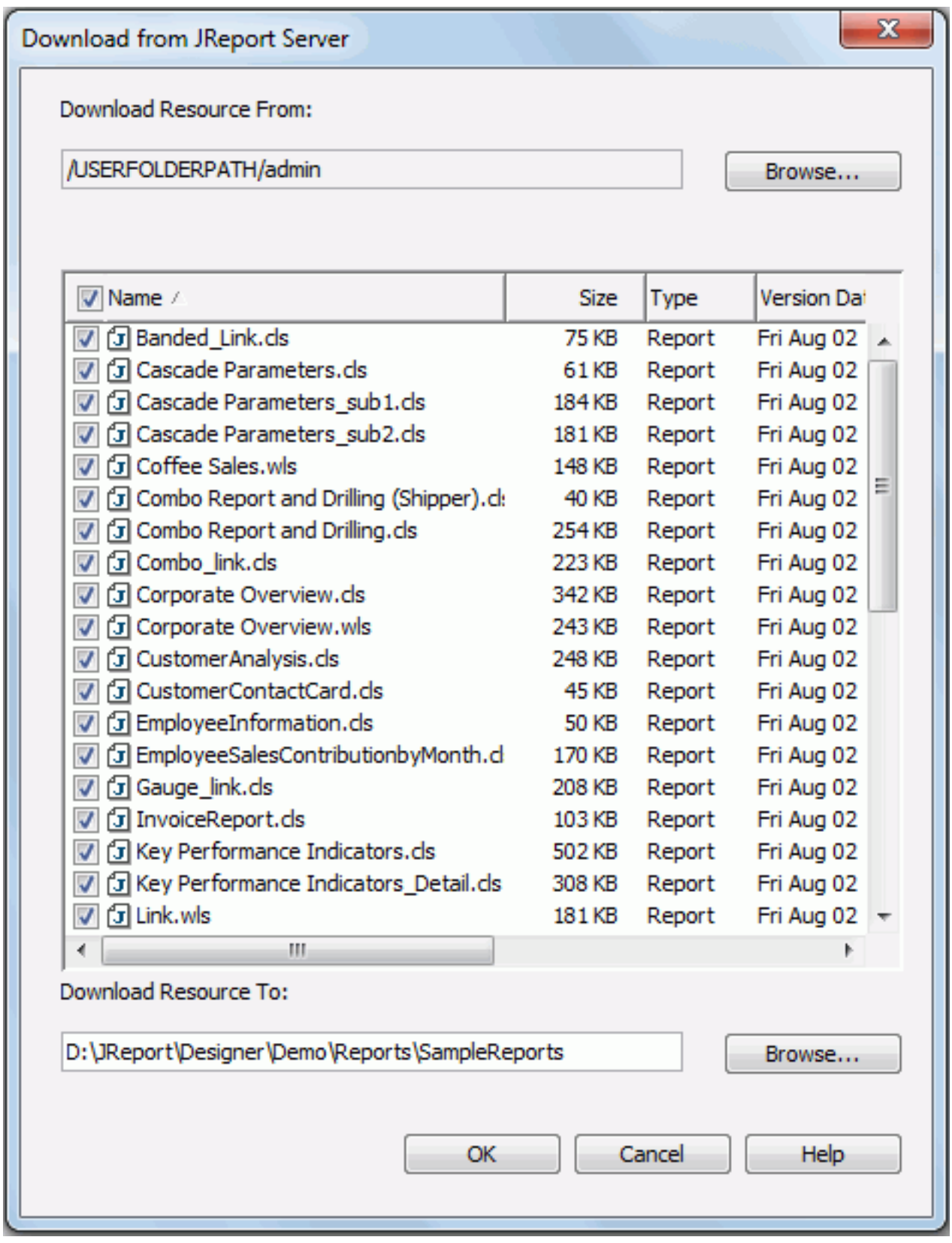
Displays the help document about this feature.

Related topics:

- Downloading customized controls

Download from JReport Server dialog

This dialog appears when you click File > Publish and Download > Download from Server > Download Report from Server/Download Component from Server on the menu bar, specify the information in the Connect to JReport Server dialog and then click Connect. It helps you to download resources from JReport Server to the specified directory on your local.



The following are details about options in the dialog:

Download Resource From

Specifies the folder on the server from which resources will be downloaded. You can click the Browse button to select the path in the Select Folder dialog.

Resource tree box

Lists the resources, including catalog files and report files, in the specified folder of JReport Server. If you specify to download resources from the Public Reports folder on the server, only the resources on which you have the Visible permission are listed in this box. Check the checkboxes ahead of the resources you want to download.

- **Name**
  Displays the resource names.
- **Size**
  Displays the file size.
- **Type**
  Displays the file type.
- **Version Date**
  Displays the latest version date of the resources.
- **Description**
  Shows other necessary descriptions of the resources if any.
- **Has Read Permission**
  Shows whether you have the Read permission on the resources. You can only download resources on which you have the Read permission from JReport Server.

Tips:

- Except for the Name column, you can specify whether or not to show the other columns in this box by right-clicking on any column header and then checking/unchecking the corresponding items on the shortcut menu.
- You can sort the resources by name, size, type and so on by clicking on the corresponding column header.

Download Resource To

Specifies the directory to which the resources will be downloaded. By default, the path of the currently open catalog is shown here. You can click the Browse button to find another directory.

OK

Applies the changes and downloads the resources from JReport Server to the specified directory.

Cancel

Closes the dialog, leaving any changes unsaved.

Help

Displays the help document about this feature.

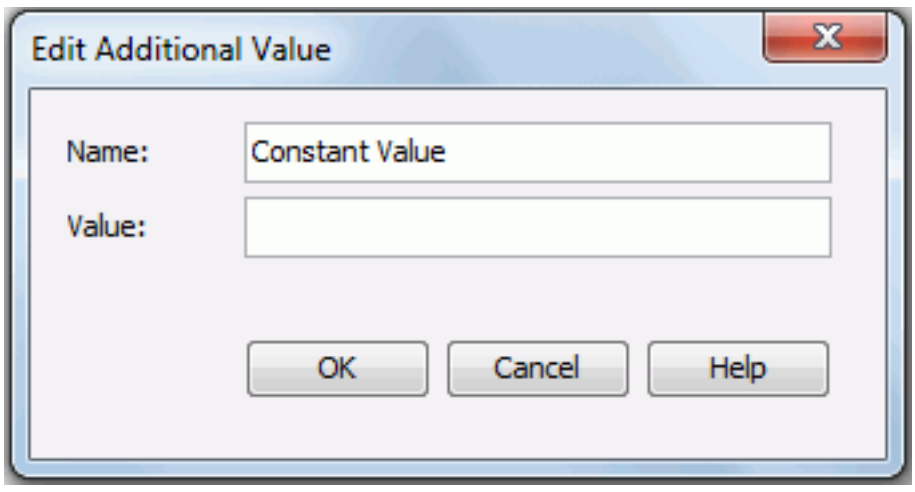
Related topics:

- Downloading resources from JReport Server

Edit Additional Value dialog

The dialog appears when you select Constant Value/Average Value in the Resources box in the Display screen of the chart wizard and click to add it to the Show Values box, or select an additional value in the Show Values box and then click when you create or modify a chart. It helps you to add additional values to be shown in the chart or modify the existing additional values. It varies according to different value types.

If it is a constant value, options in the dialog are as follows.



Name

Specifies the display name of the constant value.

Value

Specifies the constant value. Its data type must be numerical.

OK

Creates the constant value and closes the dialog.

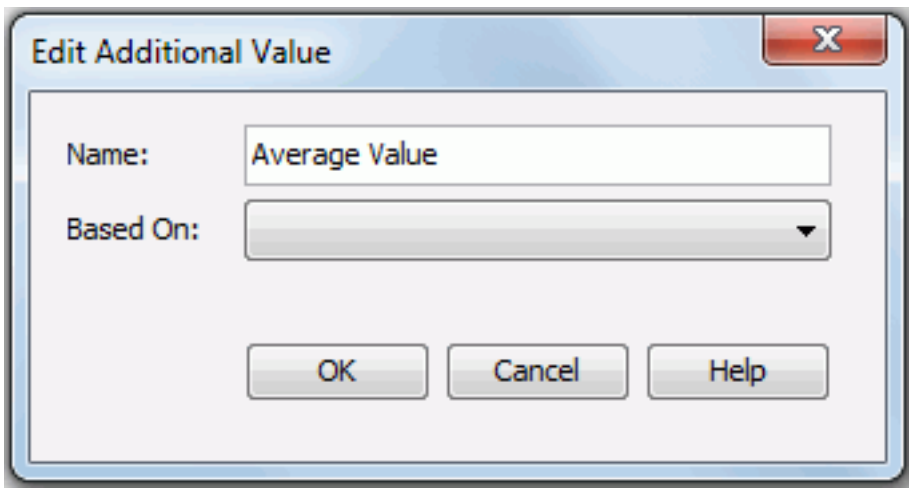
Cancel

Does not retain any changes and closes the dialog.

Help

Displays the help document about this feature.

If it is an average value, options in the dialog are as follows.



Name

Specifies the display name of the average value.

Based On

Specifies a summary or formula field based on which the average value will be calculated.

OK

Creates average value and closes the dialog.

Cancel

Does not retain any changes and closes the dialog.

Help

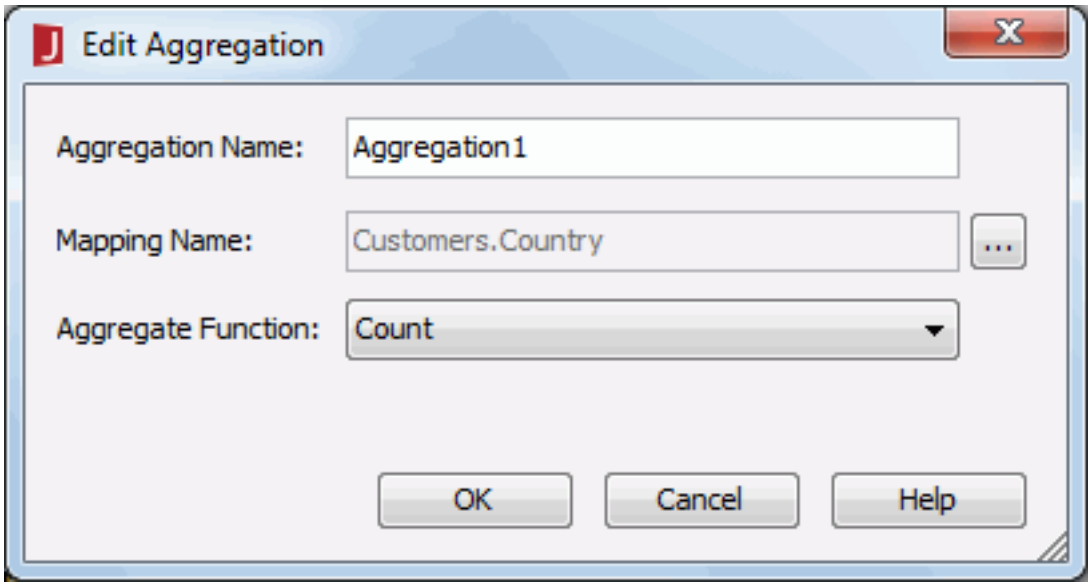
Displays the help document about this feature.

Related topics:

- Inserting a chart

Edit Aggregation dialog

The dialog appears when you right-click a dynamic aggregation and then select Edit Aggregation from the shortcut menu in the Resource View panel of the main window. It helps you to edit the selected dynamic aggregation.



Aggregation Name

Specifies the name of the aggregation. Input the name as you want in the text field.

Mapping Name

Specifies the mapping name of the field to which the aggregation is mapped. Click ... to select the field.

Aggregate Function

Specifies the function for the aggregation. For details about each function, see Math functions.

OK

Edits the aggregation and closes the dialog.

Cancel

Cancels the editing and closes the dialog.

Help

Displays the help document about this feature.

Related topics:

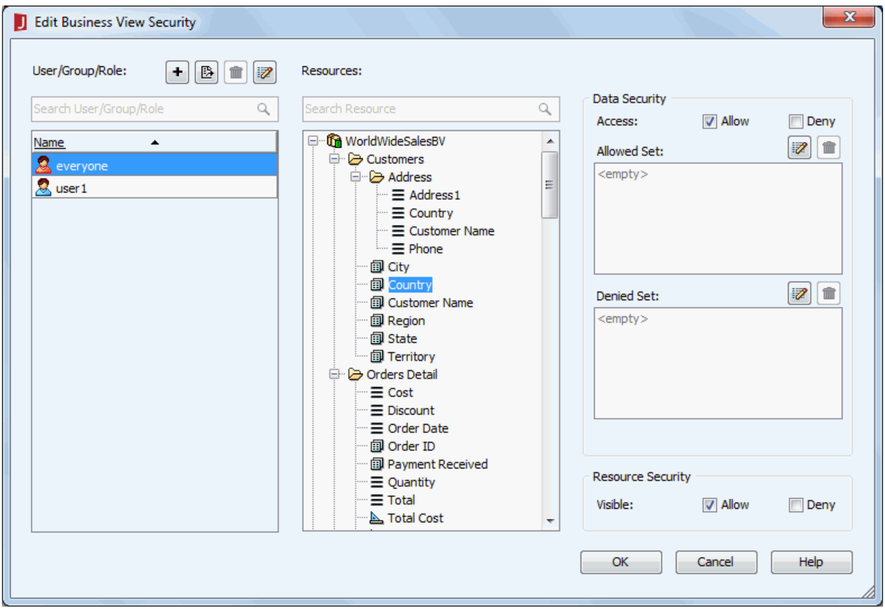
- Using dynamic resources in a library component

Edit Business View Security dialog

This dialog appears when you do one of the following:

- Click Tools > Security Configuration in the Business View Editor window.
- In the Catalog Browser, navigate to a data source > Security, right-click Business View Security, and then select Edit Business View Security from the shortcut menu.
- In the Catalog Browser, expand a business view, right-click Business View Security, and then click Edit Business View Security on the shortcut menu.

It helps you to define member-level security for business views.



The following are details about options in the dialog:

User/Group/Role

Lists all the users, roles and groups (also referred to as principals) for the security policy.

- Adds a principal. After you click the button, a drop-down menu will be shown:
  - **Add User**
    Creates a new user in the Add User dialog.
  - **Add Group**
    Creates a new group in the Add Group dialog.
  - **Add Role**
    Creates a new role in the Add Role dialog.
  - **Import from JReport Server**
    Opens the Connect to JReport Server dialog to import principals from JReport Server.
  - **Import from File**
    Imports principals from an xml file.
- Exports the principals defined in the security policy to an xml file.
- Deletes the selected principal. The everyone role cannot be deleted, and it by default has the Access and Visible permissions on all the data fields.
- Edits the selected principal in the displayed dialog.
- **Quick search toolbar**
  Type in text in the toolbar to search for principals in the list below. The names containing the matched text will be shown.

Resources

Lists all the business views in the current catalog data source.

- **Quick search toolbar**
  Searches for the resource in the list below. The names containing the matched text will be shown.

Data Security

- **Access**
  Specifies whether values of the selected field is available to the selected principal. Without the Access permission, when the principal runs reports which contain objects such as dynamic formulas and filters that reference the field, these objects will not be able to get values from the field and instead get a null value.

  The Access permission can be set to Allow, Deny, or unknown (neither Allow nor Deny).

  If a principal's Access permission on a field finally turns out to be unknown after inheriting permissions from the parent principals, the principal's Access permission on the field will be regarded as Deny.

The following options are enabled when a group field is selected.

- **Allowed Set**
  Specifies the members of the selected group field which the selected principal is allowed to access. It could be <All>, some members of the group field, a condition that retrieves certain members, or null.
  - Edits the allowed members in the Edit Values dialog.
  - Deletes the specified members or condition.
- **Denied Set**
  Specifies the members of the selected group field which the selected principal is not allowed to access. It could be <All>, some members of the group field, a condition, or null.
  - Edits the denied members in the Edit Values dialog.
  - Deletes the specified members or condition.
- **Allow Unspecified Members**
  The option is available when a user is selected. It specifies whether the unspecified members are available to the user. The unspecified members here mean the members of the group field that are not specified in its allowed/denied set for the user or in the inherited allowed/denied set from the parents.

Resource Security

- **Visible**
  Specifies whether the selected field is visible to the selected principal. Can be set to Allow, Deny, or unknown (neither Allow nor Deny).

  If a principal's Visible permission on a field finally turns out to be unknown after inheriting permissions from the parent principals, the principal's Visible permission on the field will be regarded as Deny.

OK

Applies all changes and closes the dialog.

Cancel

Does not retain any changes and closes the dialog.

Help

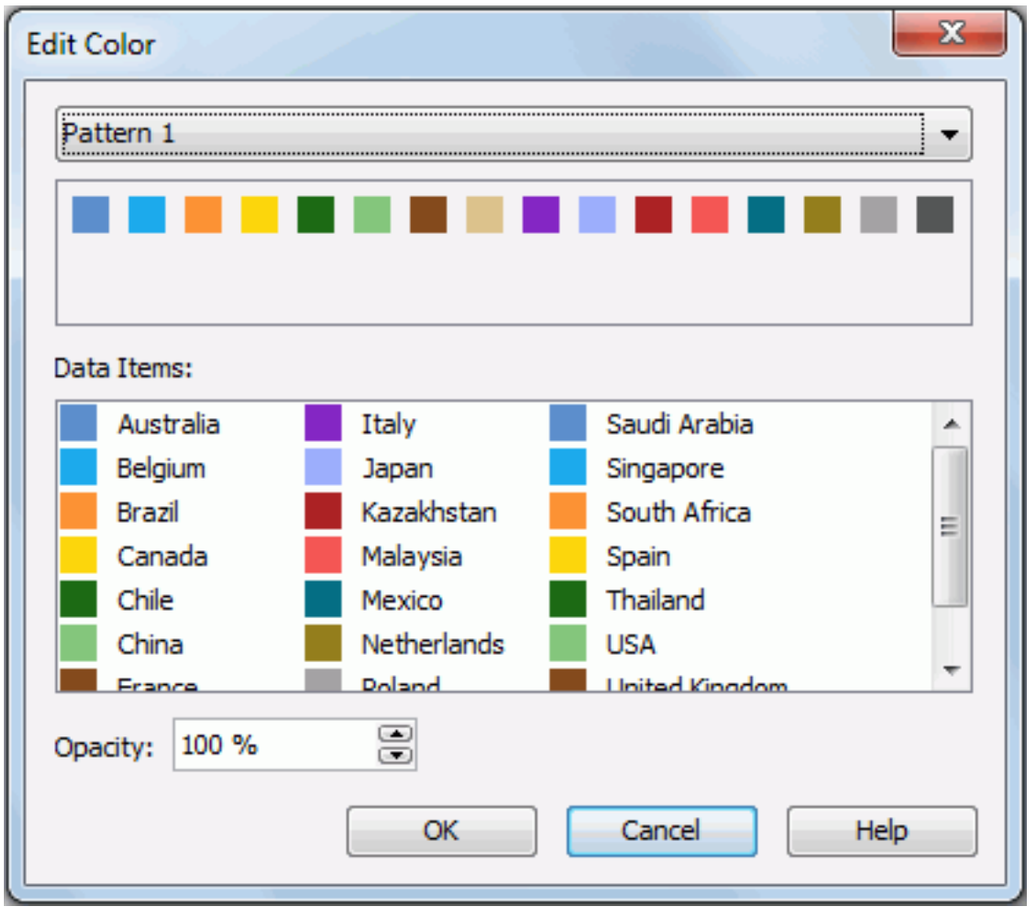
Displays the help document about this feature.

Related topics:

- Configure member-level security

Edit Color dialog

The dialog appears when you click the button next to the Color By or Fill By text box when one or more group fields have been added to the text box in the Display screen of the map wizard. It helps you to specify colors of the markers or areas in a geographic map.



The following are details about options in the dialog:

Color pattern drop-down list

Specifies the color pattern to be used.

Preview box

Displays a preview of the colors in the selected pattern.

Data Items

Lists the field values with the colors in the selected pattern. You can also change the color for each value by selecting it and then clicking a color in the preview box.

Opacity

Specifies the opacity of the colors in the selected color pattern.

OK

Applies the specified colors to the markers or areas and closes the dialog.

Cancel

Cancels the operation and closes the dialog.

Help

Displays the help document about this feature.

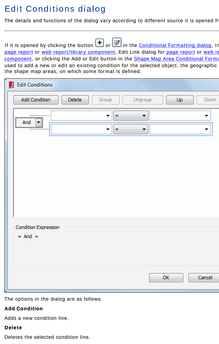
Related topics:

- Inserting a geographic map to a web report

Edit Conditions dialog

The details and functions of the dialog vary according to different source it is opened from.

If it is opened by clicking the button or in the Conditional Formatting dialog, Insert Link dialog for page report or web report/library component, Edit Link dialog for page report or web report/library component, or clicking the Add or Edit button in the Shape Map Area Conditional Formatting dialog, it is used to add a new or edit an existing condition for the selected object, the geographic map markers, or the shape map areas, on which some format is defined.



The options in the dialog are as follows.

Add Condition

Adds a new condition line.

Delete

Deletes the selected condition line.

Group

Makes the selected conditions in a group. Conditions can also be added to an existing group by selecting the conditions and the group while holding the Ctrl button, and then clicking the Group button.

Ungroup

Makes the selected condition ungrouped.

Up

Moves the selected condition or group up to a higher level.

Down

Moves the selected condition or group down to a lower level.

Logic

Lists the logic operator.

- **And**
  Logic operator And which is applied to this and the next line.
- **Or**
  Logic operator Or which is applied to this and the next line.
- **And Not**
  Logic operator And Not which is applied to this and the next line.
- **Or Not**
  Logic operator Or Not which is applied to this and the next line.

Field

Specifies the field on which the condition is based.

Operator

Specifies the operator to compose the condition expression.

- **=**
  Equal to
- **>**
  Greater than
- **>=**
  Greater than or equal to
- **<**
  Less than
- **<=**
  Less than or equal to
- **!=**
  Not equal to

Value

Specifies the value of how to build the condition.

- If you are editing conditional link or conditional formatting on a field, you can either type in the value manually or select the value from the drop-down list, where the top 50 values of the selected field are listed.
- If you are editing conditional formatting on map areas, type in the value manually or click the button next to the value text box to specify the value.

Condition Expression

Displays the condition expression.

OK

Applies the changes and closes the dialog.

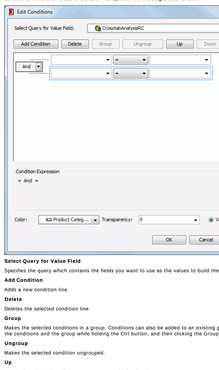
Cancel

Cancels the changes and exits the dialog.

Help

Displays the help document about this feature.

If it is opened by clicking in the Format Bar/Line/Pie/Donut dialog, it is used to add more complicated conditional fill to the values of a chart. The options in the dialog are as follows.



Select Query for Value Field

Specifies the query which contains the fields you want to use as the values to build the condition.

Add Condition

Adds a new condition line.

Delete

Deletes the selected condition line.

Group

Makes the selected conditions in a group. Conditions can also be added to an existing group by selecting the conditions and the group while holding the Ctrl button, and then clicking the Group button.

Ungroup

Makes the selected condition ungrouped.

Up

Moves the selected condition or group up to a higher level.

Down

Moves the selected condition or group down to a lower level.

Logic

Lists the logic operator.

- **And**
  Logic operator And which is applied to this and the next line.
- **Or**
  Logic operator Or which is applied to this and the next line.
- **And Not**
  Logic operator And Not which is applied to this and the next line.
- **Or Not**
  Logic operator Or Not which is applied to this and the next line.

Field

Specifies the field on which the condition is based.

Operator

Specifies the operator to compose the condition expression.

- **=**
  Equal to
- **>**
  Greater than
- **>=**
  Greater than or equal to
- **<**
  Less than
- **<=**
  Less than or equal to
- **!=**
  Not equal to

Value

Specifies the value of how to build the condition.

Condition Expression

Displays the condition expression.

Color

Specifies the color that will be applied to the chart values which meet the condition. Select a field from the drop-down list to control the color. If the given fields cannot meet your requirement, you can click <Input...> and then input the value directly in the text box.

Transparency

Specifies the transparency of the color. Select a field from the drop-down list to control the transparency. If the given fields cannot meet your requirement, you can click <Input...> and then input the value directly in the text box.

Value

Specifies to display the data in the condition expression as value. The option only applies to pie chart, 100% Stacked Bar/Bench 2-D chart, 100% Stacked Bar/Bench 3-D chart and 100% Stacked Line 2-D chart, and only when the field which the condition is based on is value field.

Percent

Specifies to display the data in the condition expression in percent. The option only applies to pie chart, 100% Stacked Bar/Bench 2-D chart, 100% Stacked Bar/Bench 3-D chart and 100% Stacked Line 2-D chart, and only when the field which the condition is based on is a value field.

OK

Applies the changes and closes the dialog.

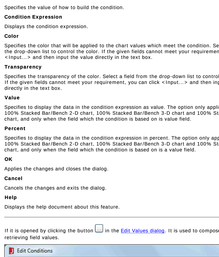
Cancel

Cancels the changes and exits the dialog.

Help

Displays the help document about this feature.

If it is opened by clicking the button in the Edit Values dialog. It is used to compose an expression for retrieving field values.



The options in the dialog are as follows.

Add Condition

Adds a new condition line.

Delete

Deletes the selected condition line.

Group

Makes the selected conditions in a group. Conditions can also be added to an existing group by selecting the conditions and the group while holding the Ctrl button, and then clicking the Group button.

Ungroup

Makes the selected condition ungrouped.

Up

Moves the selected condition or group up to a higher level.

Down

Moves the selected condition or group down to a lower level.

Logic

Lists the logic operator.

- **And**
  Logic operator And which is applied to this and the next line.
- **Or**
  Logic operator Or which is applied to this and the next line.
- **And Not**
  Logic operator And Not which is applied to this and the next line.
- **Or Not**
  Logic operator Or Not which is applied to this and the next line.

Field

The field is read only.

Operator

Specifies the operator to compose the condition expression.

- **=**
  Equal to
- **>**
  Greater than
- **>=**
  Greater than or equal to
- **<**
  Less than
- **<=**
  Less than or equal to
- **!=**
  Not equal to
- **[not] in**
  Causes an enumerated list of values to appear in the WHERE clause predicate, used for evaluating for a true condition. For operator "in" or "not in", it is allowed to input multiple values in the Value input box and separate the values with comma (,).
- **[not] like**
  Like string pattern matching operator is used to compare the first expression string value to the pattern string (the second expression). If you want to use wildcard character in the pattern string, only "_" and "%" are supported.
- **[not] between**
  Allows the system to evaluate whether or not data values are located between a range of values indicated in the predicate. For operator "between" or "not between", there are two value input boxes for inputting the same type of values.
- **is [not] null**
  It is used in WHERE clause predicates to match null values occurring in a specified data field. For operator "is null" or "is not null", the value input box is hidden.

Value

Specifies the value of how to build the condition. You can either type in the value manually or select the value from the drop-down list, where the top 50 values of the selected field are listed.

When the field is based on a formula which references a parameter, the value list will be null. Parameters are not supported here.

SQL Statement

Displays the statement expression.

OK

Applies the changes and closes the dialog.

Cancel

Cancels the changes and exits the dialog.

Help

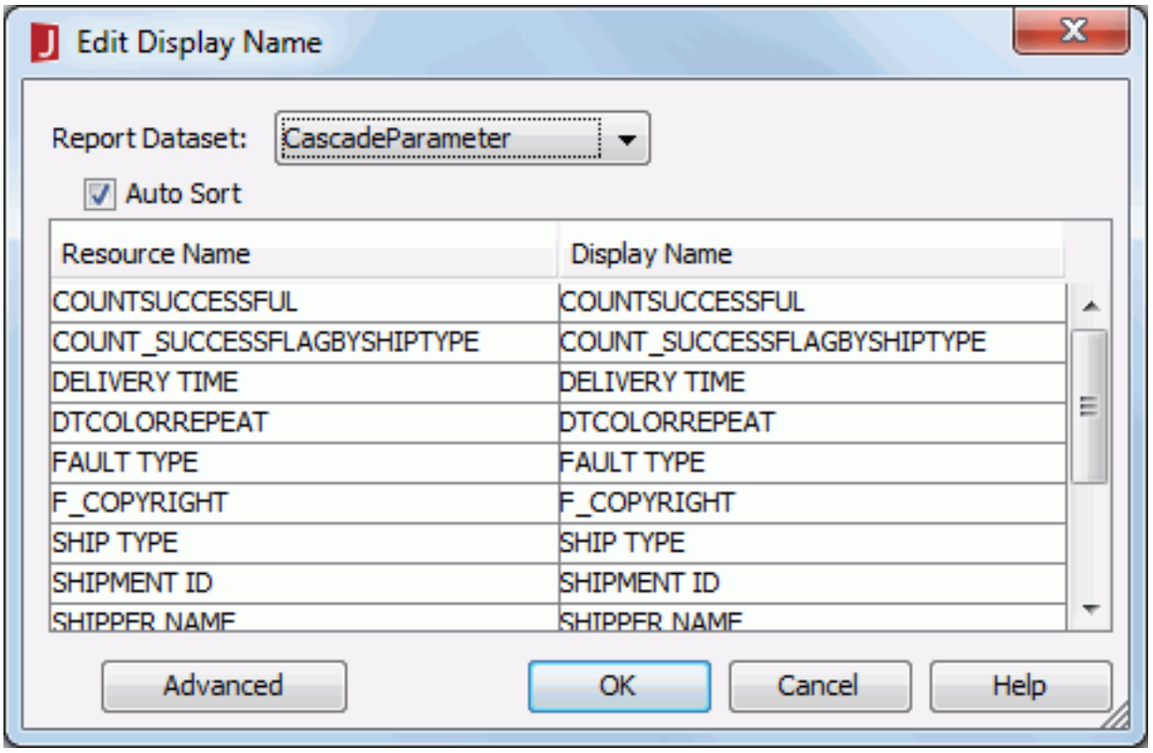
Displays the help document about this feature.

Related topics:

- Adding conditional formats to a DBfield
- Binding a link to a label
- Adding conditional formats to shape map areas
- Adding conditional color fill to a chart
- Configuring member-level security

Edit Display Name dialog

This dialog appears when you click Report > Edit Display Name on the menu bar. It helps you to customize the field names in a dataset so as to make them easier to be identified in Page Report Studio.



The following are details about options in the dialog:

Report Dataset

Lists all the datasets in the current report.

Auto Sort

Specifies to sort the fields automatically.

Resource Name

Displays the names of fields in the selected dataset.

Display Name

Specifies the display names for fields in the selected dataset.

Advanced

Opens the Edit Display Name for Component dialog to configure advanced display name settings.

OK

Applies the changes and closes the dialog.

Cancel

Cancels the changes and exits the dialog.

Help

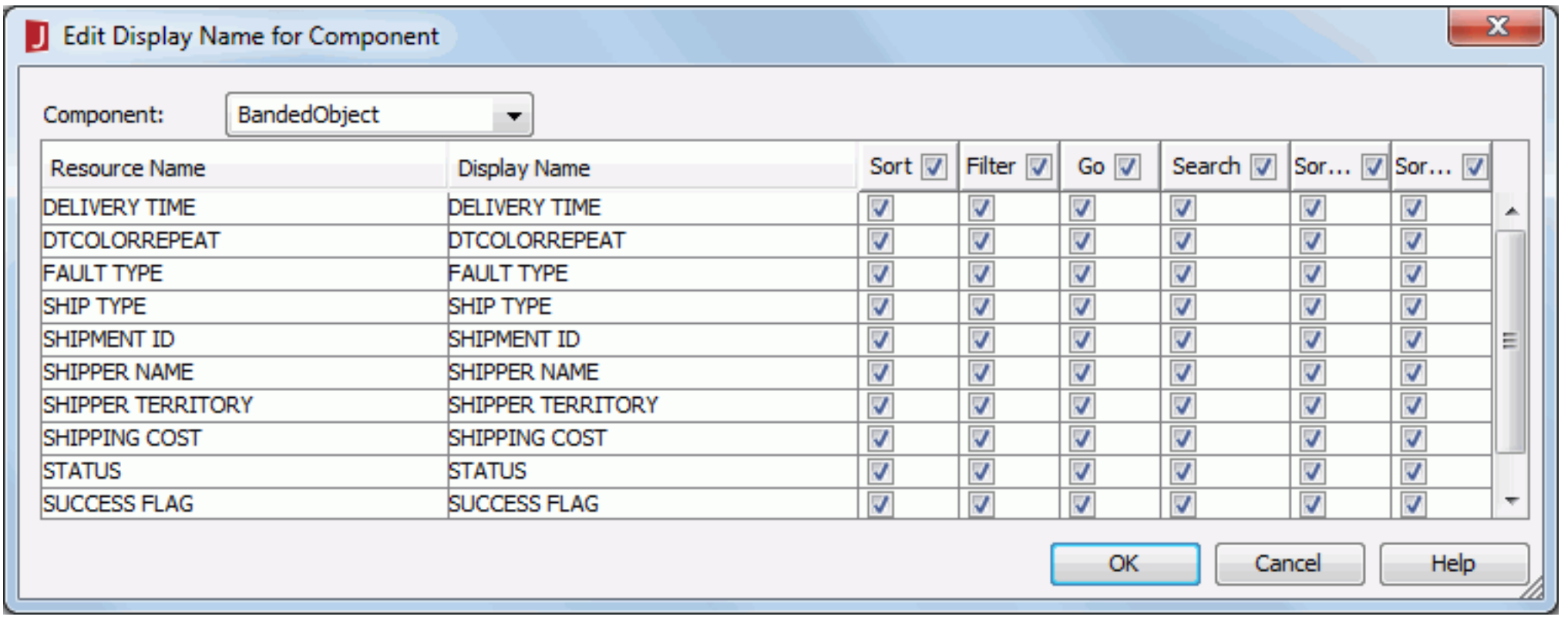
Displays the help document about this feature.

Related topics:

- Customizing the field display names

Edit Display Name for Component dialog

This dialog appears when you click the Advanced button in the Edit Display Name dialog, or right-click a data container and select Edit Display Name from the shortcut menu in the design area. It helps you to define in which actions display names of fields in the data container will take part in Page Report Studio.



The following are details about options in the dialog:

Component

- If the dialog is opened by clicking the Advanced button in the Edit Display Name dialog, in the drop-down list, all the data containers in current report will be listed.
- If the dialog is opened by clicking Edit Display Name on the shortcut menu of a data container in the design area, only the specified component will be listed in this drop-down list.

Resource Name

Displays the names of fields in the selected component.

Display Name

Shows the display names for fields in the selected component.

Action columns

Indicates whether the actions are enabled for the fields. When an action is enabled for a field, the field's display name instead of the mapping name will be shown in the corresponding dialog or submenu in Page Report Studio; when not enabled, end users will not be able to perform this action on the field in Page Report Studio. If the display name of any field is set to be blank, the field will not be available for any of the actions below. Check the box on the column header to enable the corresponding action on all fields.

- **Sort**
  Specifies whether to enable the Sort action on the fields. The Sort column is disabled when the selected component is a crosstab or chart.
- **Filter**
  Specifies whether to enable the Filter action on the fields.
- **Go**
  Specifies whether to enable the Go action on the fields. The Go column is disabled when the selected component is a crosstab.
- **Search**
  Specifies whether to enable the Search action on the fields. The Search column is disabled when the selected component is a crosstab or chart.
- **Sort in Group**
  For a grouped banded object or table, end users can also choose to sort the groups at certain group level based on a specified field in each group in Page Report Studio. Here you can decide whether to enable this Sort in Group action on the fields by checking the corresponding checkboxes.

OK

Applies the changes and closes the dialog.

Cancel

Cancels the changes and exits the dialog.

Help

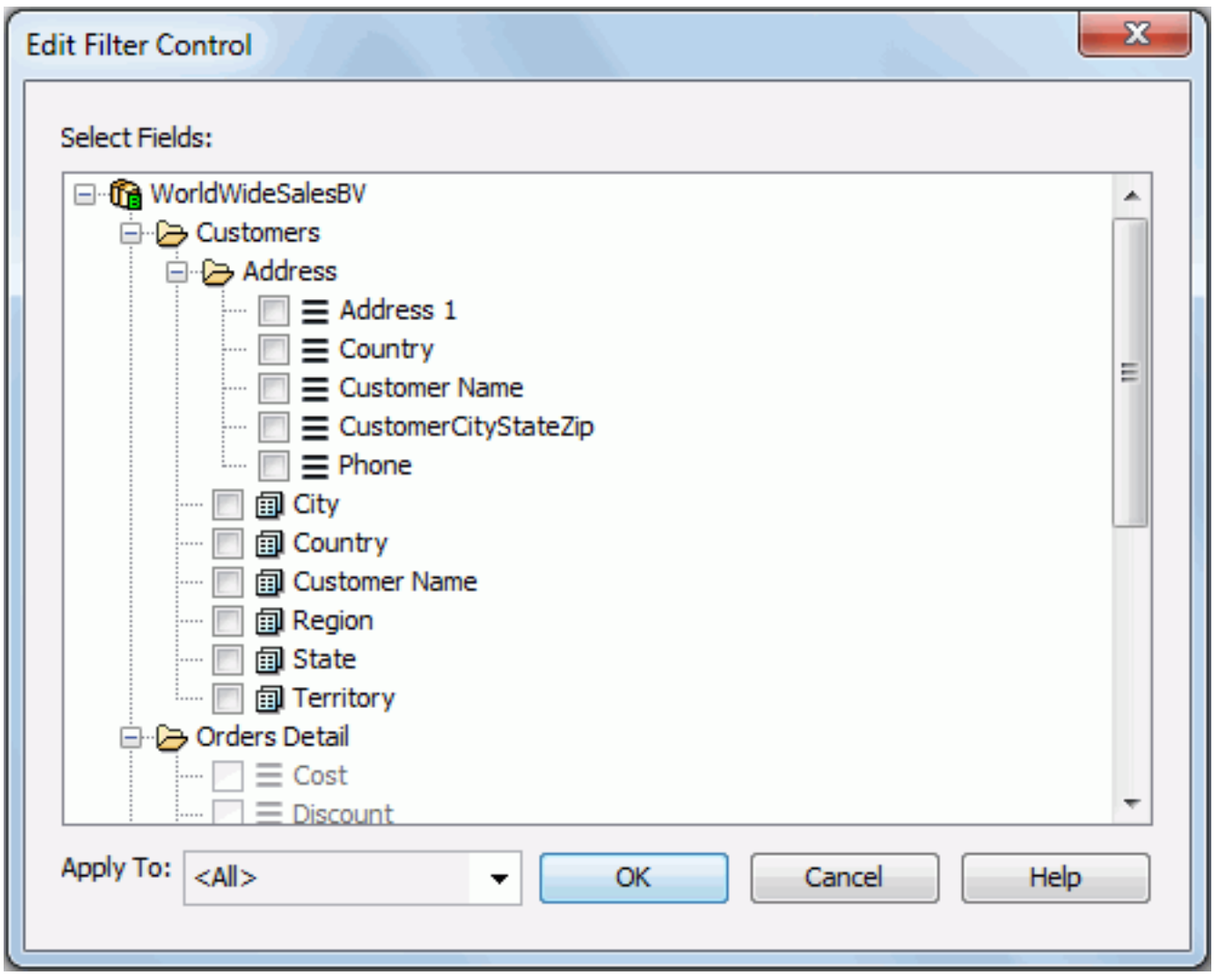
Displays the help document about this feature.

Related topics:

- Customizing the field display names

Edit Filter Control dialog

The dialog appears when you right-click a filter control and select Edit Filter Control from the shortcut menu. It helps you to edit the filter control.



The following are details about options of the dialog:

Select Fields

Specifies the fields to bind to the filter control. All the selected fields should be of the same data type. All uncomparable data type fields cannot be bound to filter controls, such as Binary, Blob, Clob, Longvarchar, Longvarbinary, and Varbinary.

The common usage is to select one field, and then based on the field to filter the data of the components created from the same data source as the field.

To filter components using different data sources, choose a common field all the data sources contain and select the field in all the data sources.

Apply To

Specifies the target data components you want to apply the filter to. The drop-down list includes all the data components in the current report that are based on the same data sources the selected fields are in.

OK

Inserts a filter control into the report and closes the dialog.

Cancel

Cancels the operation and closes the dialog.

Help

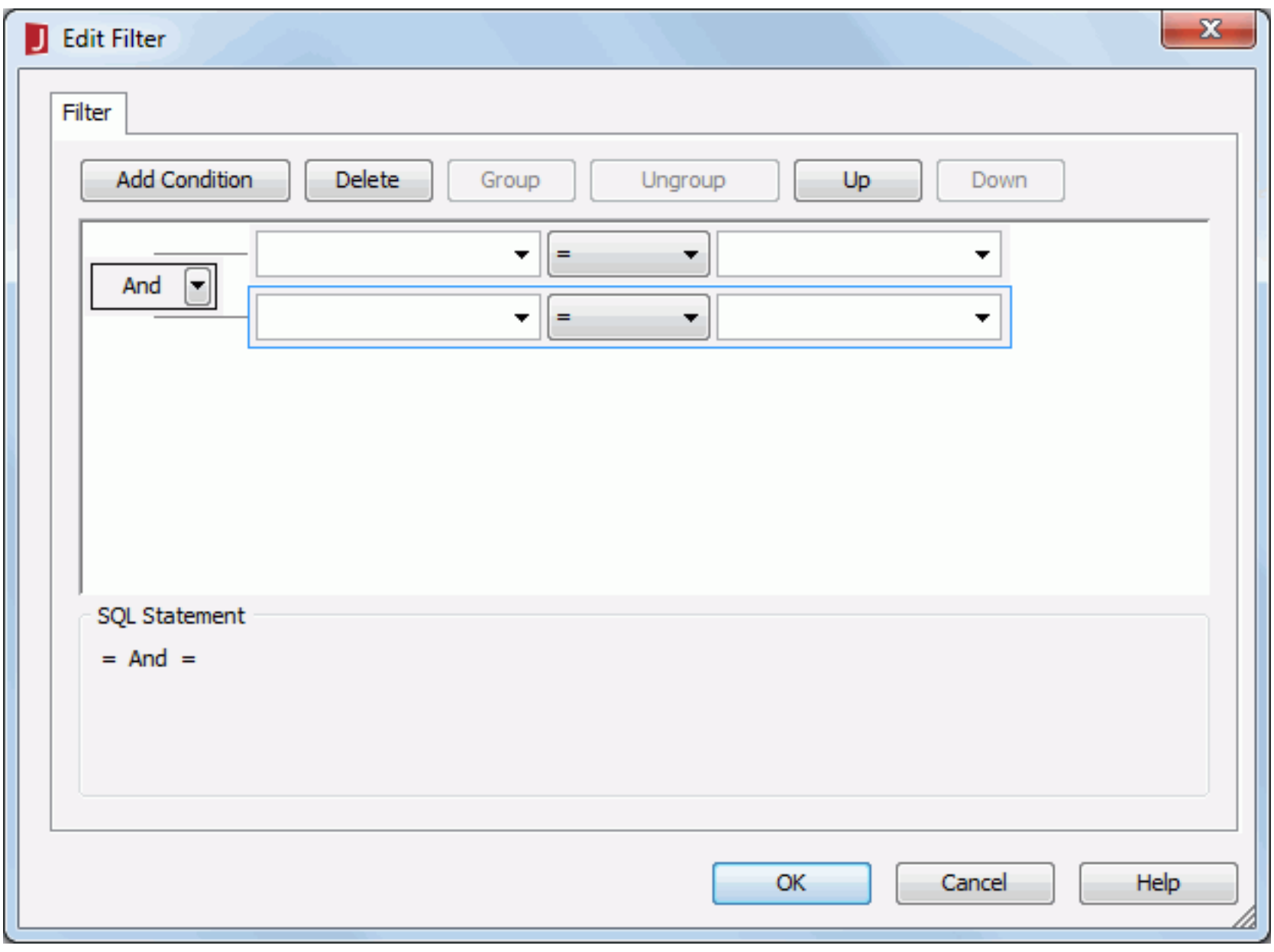
Displays the help document about this feature.

Related topics:

- Using slider or filter control to filter data

Edit Filter dialog for page report

This dialog appears when you right-click a data container in the design area in a page report and select Format Filter from the shortcut menu. It helps you to filter the data used in the data container.



The following are details about options in the dialog:

Add Condition

Adds a new condition line.

Delete

Deletes the selected condition line.

Group

Makes the selected conditions in a group. Conditions can also be added to an existing group by selecting the conditions and the group while holding the Ctrl button, and then clicking the Group button.

Ungroup

Makes the selected condition ungrouped.

Up

Moves the selected condition or group up to a higher level.

Down

Moves the selected condition or group down to a lower level.

Logic

Lists the logic operator.

- **And**
  Logic operator And which is applied to this and the next line.
- **Or**
  Logic operator Or which is applied to this and the next line.
- **And Not**
  Logic operator And Not which is applied to this and the next line.
- **Or Not**
  Logic operator Or Not which is applied to this and the next line.

Field

Specifies the field on which the condition is set.

Operator

Specifies the operator to compose the filter expression.

- **=**
  Equal to
- **>=**
  Greater than or equal to
- **>**
  Greater than
- **<**
  Less than
- **<=**
  Less than or equal to
- **!=**
  Not equal to
- **[not] in**
  Causes an enumerated list of values to appear in the WHERE clause predicate, used for evaluating for a true condition. For operator ‘in' or ‘not in', it is allowed to input multiple values in the Value input box and separate the values with comma (,).
- **[not] like**
  Like string pattern matching operator is used to compare the first expression string value to the pattern string (the second expression). If you want to use wildcard character in the pattern string, only "_" and "%" are supported.
- **[not] between**
  Allows the system to evaluate whether or not data values are located between a range of values indicated in the predicate. For operator ‘between' or ‘not between', there are two value input boxes for inputting the same type of values.
- **is [not] null**
  It is used in WHERE clause predicates to match null values occurring in a specified data field. For operator ‘is null' or ‘is not null', the value input box is hidden.

Value

Specifies the value of how to filter the field. You can type in the value manually or select the desired one from the drop-down list. When you type in the value manually, if multiple values are required, they should be separated with ",", and if "," or "\" is contained in the values, write it as "\," or "\\".

SQL Statement

Displays the SQL statement of the filter expressions.

OK

Applies the changes and closes the dialog.

Cancel

Cancels the changes and exits the dialog.

Help

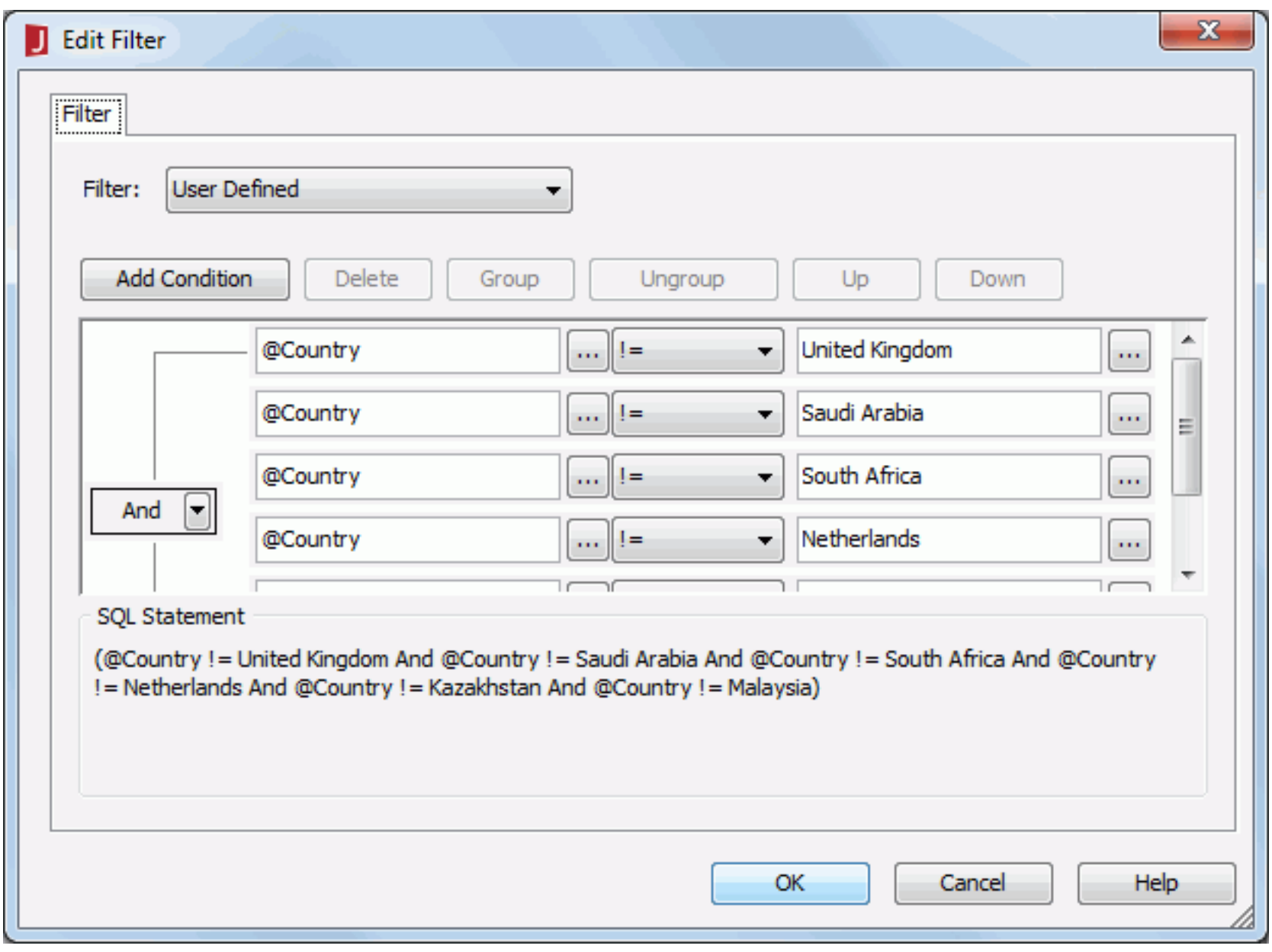
Displays the help document about this feature.

Related topics:

- Filtering the data

Edit Filter dialog for web report/library component

This dialog appears when you right-click a data container in the design area in a web report or library component and select Format Filter from the shortcut menu. It helps you to filter the data used in the data container.



The following are details about options in the dialog:

Filter

It lists all the predefined filters of the business view used by the current data container. Select one from the drop-down list to apply, or select User Defined and define a new filter according to your requirements.

Add Condition

Adds a new condition line.

Delete

Deletes the selected condition line.

Group

Makes the selected conditions in a group. Conditions can also be added to an existing group by selecting the conditions and the group while holding the Ctrl button, and then clicking the Group button.

Ungroup

Makes the selected condition ungrouped.

Up

Moves the selected condition or group up to a higher level.

Down

Moves the selected condition or group down to a lower level.

Logic

Lists the logic operator.

- **And**
  Logic operator And which is applied to this and the next line.
- **Or**
  Logic operator Or which is applied to this and the next line.
- **And Not**
  Logic operator And Not which is applied to this and the next line.
- **Or Not**
  Logic operator Or Not which is applied to this and the next line.

Field

Specifies the field on which the condition is set.

Operator

Specifies the operator to compose the filter expression.

- **=**
  Equal to
- **>=**
  Greater than or equal to
- **>**
  Greater than
- **<**
  Less than
- **<=**
  Less than or equal to
- **!=**
  Not equal to
- **[not] in**
  Causes an enumerated list of values to appear in the WHERE clause predicate, used for evaluating for a true condition. For operator 'in' or 'not in', it is allowed to input multiple values in the Value input box and separate the values with comma (,).
- **[not] like**
  Like string pattern matching operator is used to compare the first expression string value to the pattern string (the second expression). If you want to use wildcard character in the pattern string, only "_" and "%" are supported.
- **[not] between**
  Allows the system to evaluate whether or not data values are located between a range of values indicated in the predicate. For operator 'between' or 'not between', there are two value input boxes for inputting the same type of values.
- **is [not] null**
  It is used in WHERE clause predicates to match null values occurring in a specified data field. For operator 'is null' or 'is not null', the value input box is hidden.

Value

Specifies the value of how to filter the field. You can either type in the value manually or click ... next to the text field to display the Values dialog and specify the value. When you type in the value manually, if multiple values are required, they should be separated with ",", and if "," or "\" is contained in the values, write it as "\," or "\\".

SQL Statement

Displays the SQL statement of the filter expressions.

OK

Applies the changes and closes the dialog.

Cancel

Cancels the changes and exits the dialog.

Help

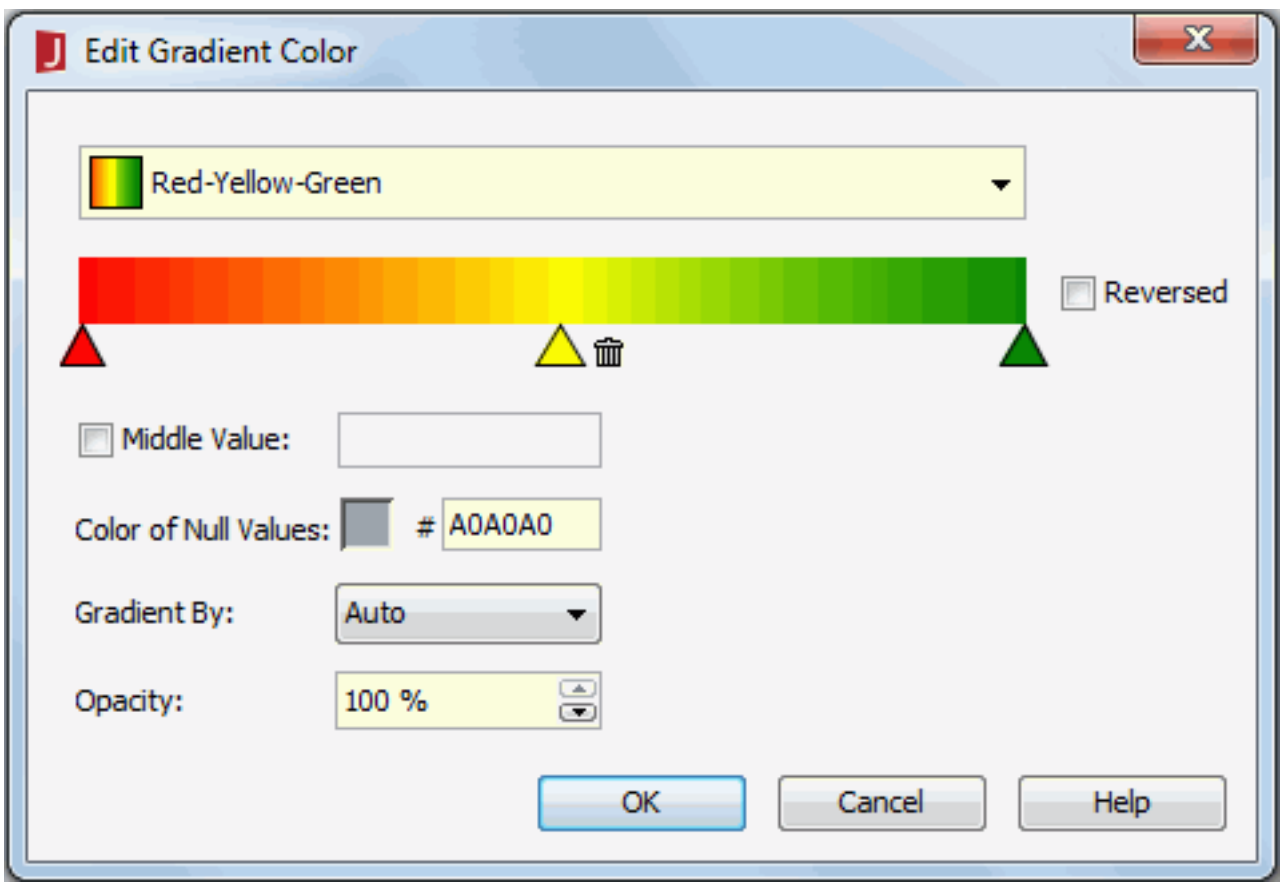
Displays the help document about this feature.

Related topics:

- Filtering the data

Edit Gradient Color dialog

The dialog appears when you click the button beside the Color By or Fill By text box when an aggregation field has been added to the text box in the Display screen of the map wizard. It helps you to specify a gradient color for the markers or areas in a geographic map.



The following are details about options in the dialog:

Gradient color drop-down list

Specifies the gradient color to be used.

Color bar

Displays the colors in the selected gradient color.

The gradient start color, middle color and end color can be customized. Click the corresponding triangle below the color bar to select a color from the color palette or click **More Colors** on the color palette to access the Pick a Color dialog in which you can select a color within a wider range. Click to delete the middle color or start color if not needed. If the start color is deleted, a default start color will be applied automatically. Click **+** to specify the start color or middle color again. The middle color can not be deleted or added when there is no start color. After you change a default gradient color, it will be saved as a customized gradient color and you will find the name changed to Customize in the color drop-down list.

Reversed

Reverses the direction of the gradient color.

Middle Value

Specifies whether to set the middle data value mapping to the middle color. It takes effect only when the middle color is not null. If unchecked, the center value between the start value and end value will be used to map to the middle color. If checked, the value you specify will be used to map to the middle color. Note that the value will be ignored if it is not between the start value and end value.

- **Auto Expand Start/End Value**
  Specifies whether to expand the start value and end value automatically. Available only when the Middle Value is set to 0. If checked, it means 0 is the separator of positive values and negative values. The negative and positive values of the maximum absolute value will be used to map to the start color and end color, and 0 will always be in the center of the data range.

Color of Null Values

Specifies the color for null values.

To change the color, click the color image button to customize the color in the color palette, or you can input the hexadecimal value of a color directly in the text field after #.

Gradient By

Specifies the gradient color algorithm between two colors.

- **Auto**
  Specifies to use the default algorithm according to different kinds of gradient colors automatically. For single color, RGB linear algorithm will be used. For start-end color, HSL linear algorithm will be used. For start-middle-end color, HSL linear algorithm will be used separately between the start-middle color and middle-end color.

- **RGB**
  Specifies to use RGB linear algorithm, which uses red, green, and blue as the algorithm parameters.

- **HSL**
  Specifies to use HSL linear algorithm, which uses hue, saturation, and lightness as the algorithm parameters.

Opacity

Specifies the opacity of the gradient color.

OK

Applies the specified gradient color to the markers or areas and closes the dialog.

Cancel

Cancels the operation and closes the dialog.

Help

Displays the help document about this feature.

Related topics:

- Inserting a geographic map to a web report

Edit Group dialog

The dialog appears when you select a group and then click in one of the following dialogs or tabs:

- Security dialog
- Edit Business View Security dialog
- Category Property dialog > Security tab
- Add View Element dialog > Security tab
- Edit View Element dialog > Security tab

It helps you to edit the group, and consists of the following tabs:

- General tab
- Relation tab

OK

Accepts all changes and closes the dialog.

Cancel

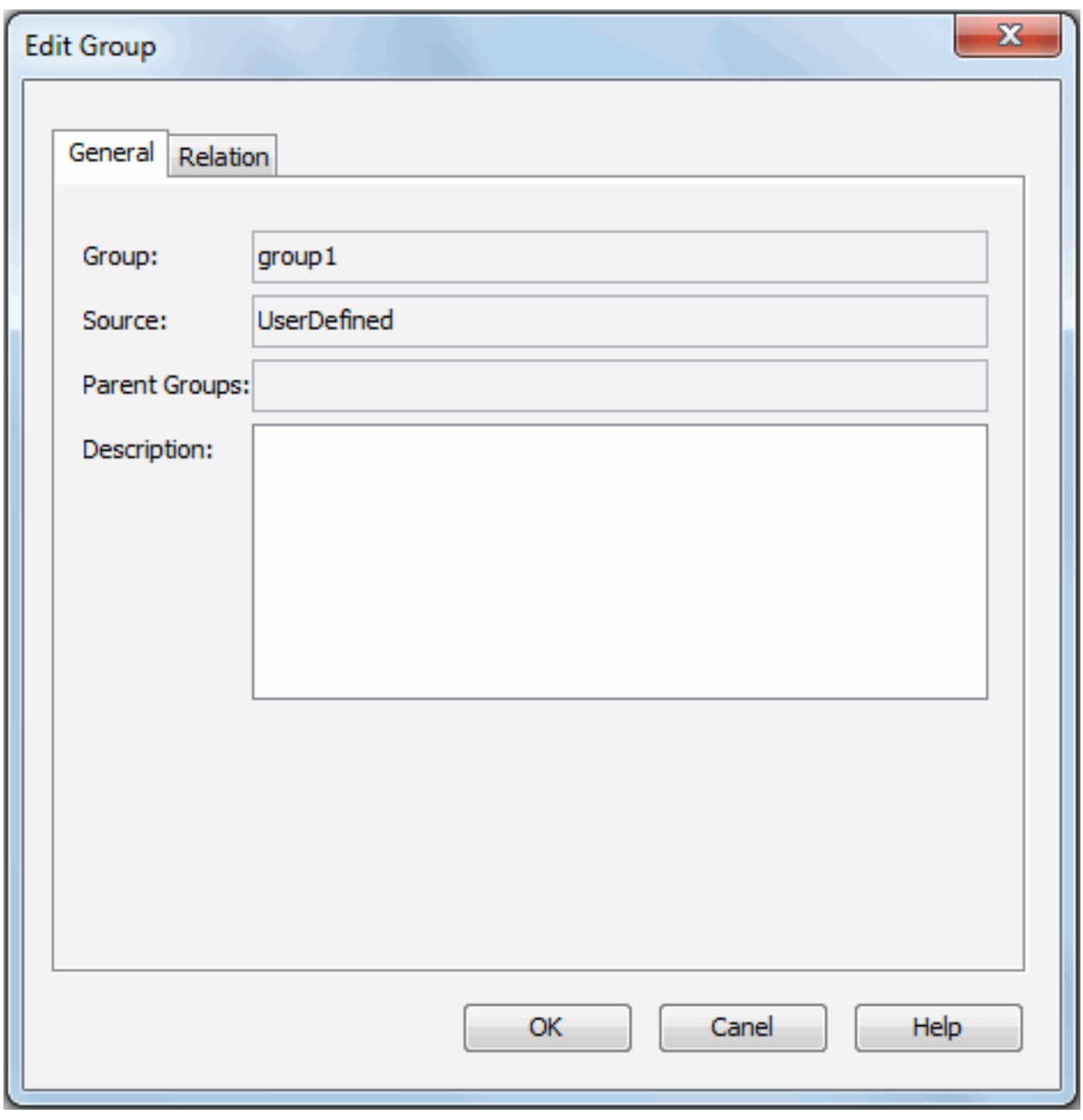
Does not retain any changes and closes the dialog.

Help

Displays the help document about this feature.

General tab

Specifies the group information.



Group

Specifies the name of the group.

Source

Displays the source of the group. It is read only.

Parent Groups

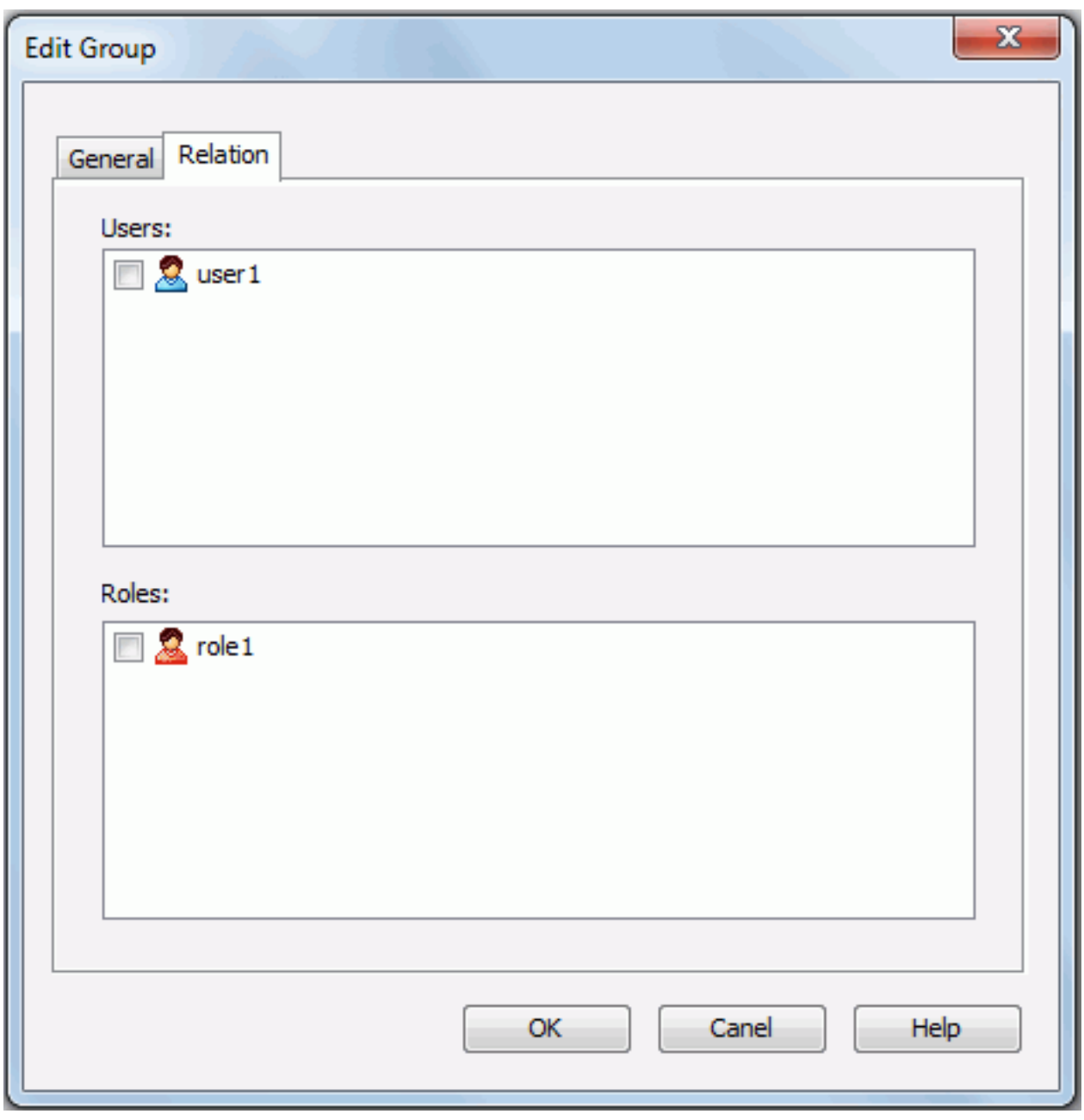
Displays the parent groups of the current group. It is read only.

Description

Specifies the description of the group.

Relation tab

Specifies the relationship of the group and the existing users and roles.



Users

All available users will be listed here. If a user is checked, the user will be added into the group.

Roles

All available roles will be listed here. If a role is checked, the group will be added into the role.

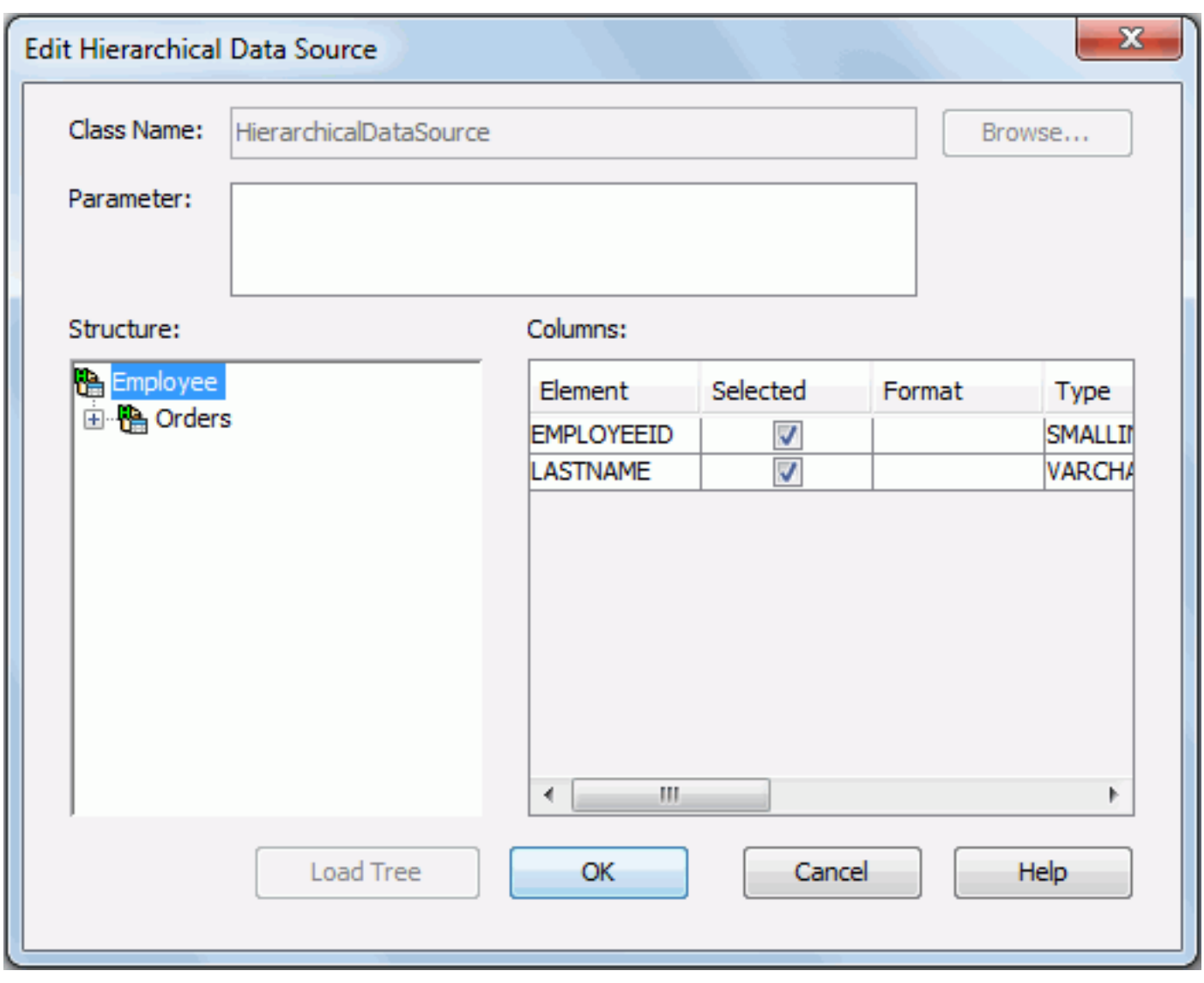
Related topics:

- Setting up a connection-scope security policy
- Configuring member-level security

Edit Hierarchical Data Source dialog

This dialog appears when you right-click a hierarchical data source and select Edit Hierarchical from the shortcut menu in the Catalog Browser. It helps you to edit an HDS in a catalog.

When the HDS you are to edit is a general HDS, this dialog displays



Class Name

Specifies the name of the entrance class that implements the HDS APIs. The class should be in the class path in the system environment.

Parameter

Specifies the parameter string that is required by the class, if any. The parameter string must match the format defined in the HDS class.

Structure

Shows the structure of the hierarchical data source.

Columns

Specifies the column settings.

- **Element**
  Displays the column name in the selected node of table.
- **Selected**
  Specifies whether to import the column into the catalog.
- **Format**
  Specifies the format in which the column values are to be displayed in the report result. Varies for different types of data, and can be manually defined.
- **Type**
  Specifies the data type of the column.
- **Precision**
  Specifies the precision of the column's values. The larger the precision is, the more memory it might take, however, the more accurate values you will get.
- **Scale**
  Specifies the number of digits to the right of the decimal point for the column's values.
- **Nullable**
  Specifies whether the column is nullable. You have three options to choose: Nullable, No Nulls and Nullable Unknown.
- **Currency**
  Specifies whether to use the Currency type for this column or not
- **Array**
  Specifies whether to use the Array type for this column or not.

Load Tree

Parses and loads data tree from the external data source file. The structure is shown in the Structure box after you click this button.

OK

Applies all changes here and close the dialog.

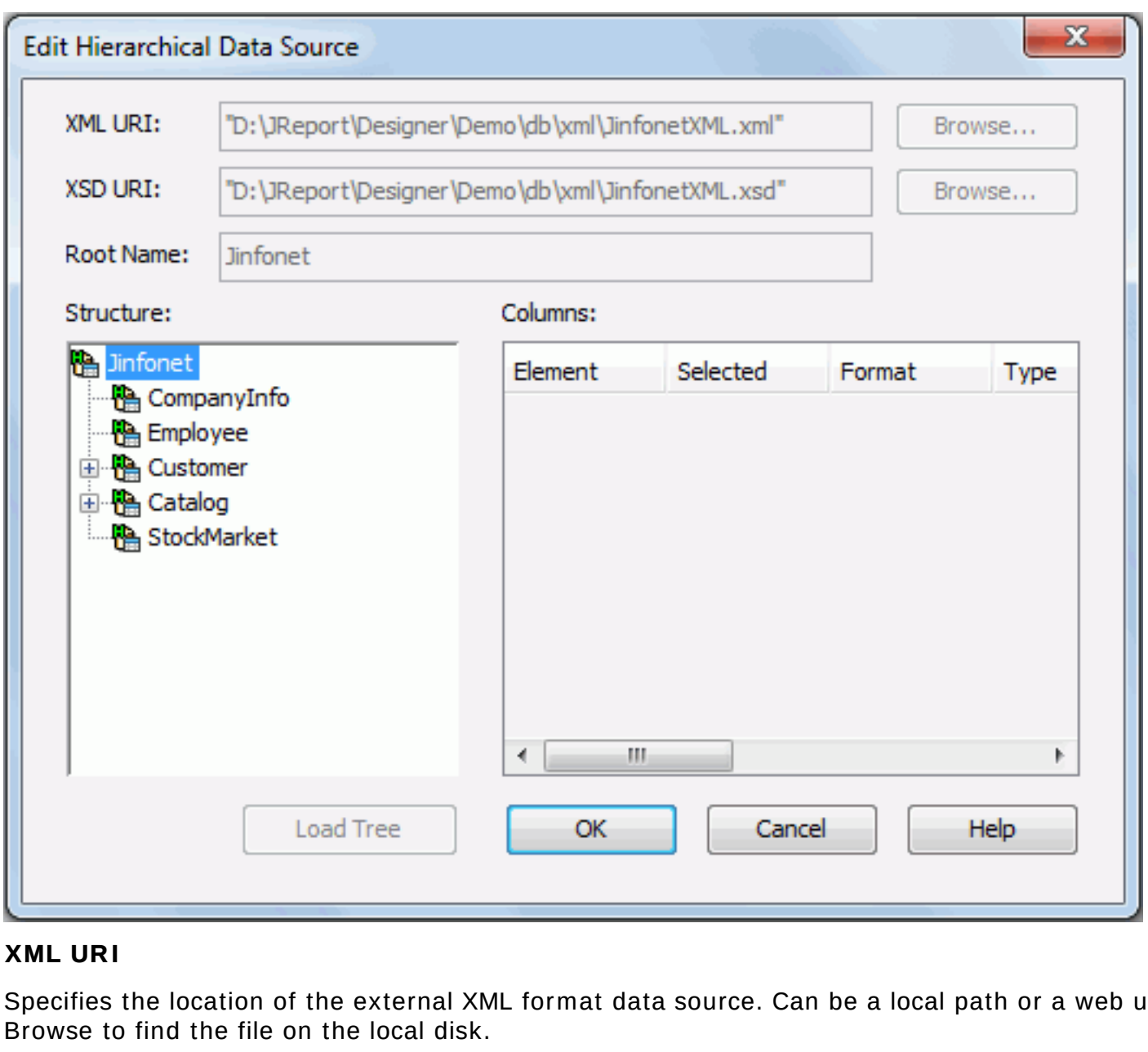
Cancel

Does not retain changes and closes the dialog.

Help

Displays the help document about this feature.

When the HDS you are to edit is an XML HDS, this dialog displays



XML URI

Specifies the location of the external XML format data source. Can be a local path or a web url. Click Browse to find the file on the local disk.

XSD URI

Specifies the location of the schema file for the XML format data source. Can be a local path or a web url. Click Browse to find the file on the local disk.

Root Name

Indicates the HDS root, starting from which JReport Designer imports the data.

Structure

Shows the structure of the hierarchical data source.

Columns

Specifies the column settings.

- **Element**
  Displays the column name in the selected node of table.
- **Selected**
  Specifies whether to import the column into the catalog.
- **Format**
  Specifies the format in which the column values are to be displayed in the report result. Varies for different types of data, and can be manually defined.
- **Type**
  Specifies the data type of the column.
- **Precision**
  Specifies the precision of the column's values. The larger the precision is, the more memory it might take, however, the more accurate values you will get.
- **Scale**
  Specifies the number of digits to the right of the decimal point for the column's values.
- **Nullable**
  Specifies whether the column is nullable. You have three options to choose: Nullable, No Nulls and Nullable Unknown.
- **Currency**
  Specifies whether to use the Currency type for this column or not
- **Array**
  Specifies whether to use the Array type for this column or not.

Load Tree

Parses and loads data tree from the external data source file. The structure is shown in the Structure box after you click this button.

OK

Applies all changes here and closes the dialog.

Cancel

Does not retain changes and closes the dialog.

Help

Displays the help document about this feature.

Related topics:

- Adding an HDS to a catalog

Edit Link dialog for page report

The dialog appears when you select an object in a page report, which is defined with a link, right-click it and select Edit Link from the shortcut menu. It helps you to edit the link on the object.

Conditional Link

Specifies whether it is a conditional link. If checked, the object can be linked to different targets based on different conditions.

- Adds a new condition in the Edit Conditions dialog.
- Edits the selected condition.
- Removes the selected condition.
- Moves the selected condition one step up.
- Moves the selected condition one step down.
- **Others**
  If checked, you can define to which target the object will be linked when none of the conditions you have specified is met.

Link Type

Specifies the type of the link target for the object if it is not a conditional link or for the object under the selected condition if it is a conditional link. It can be one of the following:

- Report
- Master/Detail Report
- URL
- E-mail
- Content

More

Displays more link options. Available only for the Report, Master/Detail Report and Content link types.

OK

Accepts the link editing result and closes the dialog.

Cancel

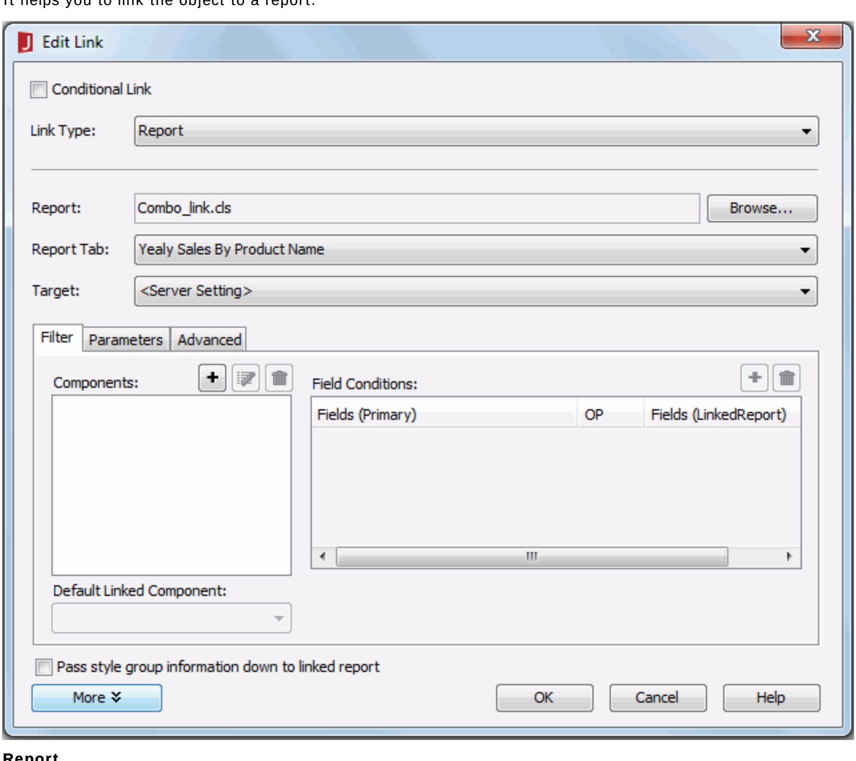
Cancels the link editing process and exits the dialog.

Help

Displays the help document about this feature.

Report

It helps you to link the object to a report.



Report

Specifies the report in which the linked report tab is. Click the Browse button to modify the desired report in the Select a Report dialog.

Report Tab

Specifies the linked report tab.

Target

Specifies the window or frame in which to load the linked report. Applied only when the link is triggered in Page Report Studio.

- **<Server Setting>**
  Loads the linked report according to setting of the Pop Up New Window for Links option in the Profile > Configure Profile > Page Report Studio > Properties > Default tab on JReport Server.
- **Same Frame**
  Loads the linked report into the same frame as the primary report.
- **Whole Window**
  Loads the linked report into the full browser window.
- **New Window**
  Loads the linked report into a new window. This window is not named.
- **Parent Frame**
  Loads the linked report into the parent frame of the frame in which the primary report is.
- **Other Frame**
  Loads the linked report into some other specified frame. If the frame name does not exist, the linked report will be loaded into a new window.

Filter tab

Specifies the filter conditions based on which the link relationship between the primary report and the linked report is set up.

- **Components**
  Specifies the components in the linked report that will be interlinked with the primary report.
  - Adds a component in the linked report to be interlinked with the primary report.
  - Edits the component in the linked report to be interlinked with the primary report.
  - Removes the selected component from the Component in Report Tab box.
- **Default Linked Component**
  Specifies the default linked component in the linked report.
- **Field Conditions**
  Lists the filter conditions between the primary report and the linked report.
  - **Fields(Primary)**
    Lists the selected fields of the primary report.
  - **OP**
    Specifies the operator to compose the condition. It can be "=", "<>", "<", ">", "<=", ">=" or "IN".
  - **Fields(LinkedReport)**
    Lists the DBFields in the datasets of the linked report which are of the same data type as the selected fields in the primary report.
  - Adds a new condition line.
  - Removes the selected condition.

Parameters tab

This tab is available when the linked report uses parameters. It helps you to assign values to the parameters of the linked report automatically.

- **Parameters**
  Lists all parameters contained in the datasets of the linked report.
- **Value**
  Lists all the DBFields, formulas, summaries and parameters in the datasets of the primary report which are of the same data type as the parameters of the linked report. Select the field the value of which you want to assign to parameter of the linked report from the drop-down list.

Advanced tab

Specifies to pass the values of the filter objects such as filter controls and sliders in the primary report to those in the linked report.

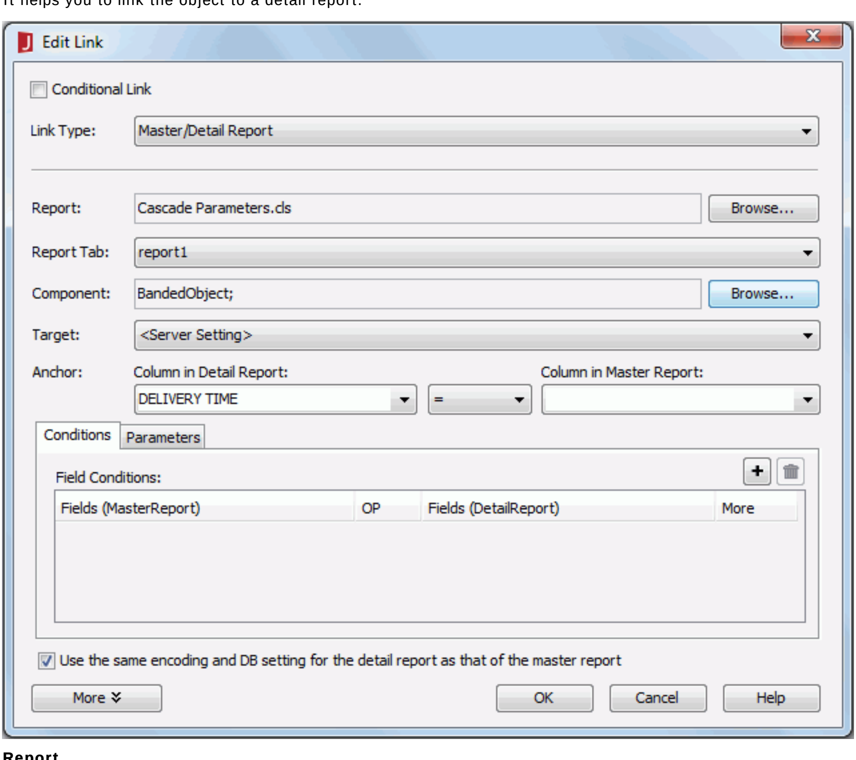
- **Primary Report Property/Object**
  Displays the filter objects in the primary report the values of which will be passed to those in the linked report. Click … to select a filter object in the primary report in the Select Value dialog.
- **Linked Report Property/Object**
  Displays the filter objects in the linked report which will adopt the values passed from filter objects in the primary report. Click … to select a filter object in the linked report in the Select Value dialog. You need to make sure that values of the filter object in the primary report can be applied to the filter object in the linked report in the same line.
- Adds a new line to pass values between the filter objects in the primary report and the linked report.
- Removes the selected line.

Pass style group information down to linked report

Specifies whether or not to apply the style group of the primary report to the linked report.

Master/Detail Report

It helps you to link the object to a detail report.



Report

Specifies the report in which the detail report tab is. Click the Browse button to modify the desired report in the Select a Report dialog.

Report Tab

Specifies the target detail report tab.

Component

Specifies the component in the detail report tab that will be interlinked with the master report. Click … to add the components. You can add more than one component as long as the components are based on the same dataset.

Target

Specifies the window or frame where to load the detail report.

- **<Server Setting>**
  Loads the detail report according to setting of the Pop up New Window for Links option in the Profile > Configure Profile > Page Report Studio > Properties > Default tab on JReport Server.
- **Same Frame**
  Loads the detail report into the same frame as the master report.
- **Whole Window**
  Loads the detail report into the full browser window.
- **New Window**
  Loads the detail report into a new window, which is not named. If this is selected, the detail report will be opened independently when you click the trigger in Page Report Studio, that is to say the join and filter conditions you set between the master report and detail report will not be applied.
- **Parent Frame**
  Loads the detail report into the parent frame of the frame in which the master report is.
- **Other Frame**
  Loads the detail report into some other specified frame. If the frame name does not exist, the detail report will be loaded into a new window.

Anchor

Specifies the relationships between the master report and the detail report.

- **Column in Detail**
  Specifies the field in the detail report which will be used to link the detail report with the master.
- **OP**
  Specifies the operator. It can be "=", "<>", "<", ">", "<="or ">=".
- **Column in Master**
  Specifies the field in the master report.

Conditions tab

Specifies the filter conditions between the master report and the detail report.

- **Field Conditions**
  Lists all the filter conditions.
  - **Fields(MasterReport)**
    Lists the selected fields of the master report.
  - **OP**
    Specifies the operator to compose the condition. It can be "=", "<>", "<", ">", "<="or ">=".
  - **Fields(DetailReport)**
    Lists the DBFields in the datasets of the detail report which are of the same data type as the selected fields in the master report.
  - **More**
    Specifies relations between two conditions. Can be AND or OR.
  - Adds a new condition line.
  - Removes the selected condition.

Parameters tab

This tab is only available when there is at least one parameter applied in the filters of the queries that the detail report uses. It helps you to assign values to the parameters of the detail report automatically.

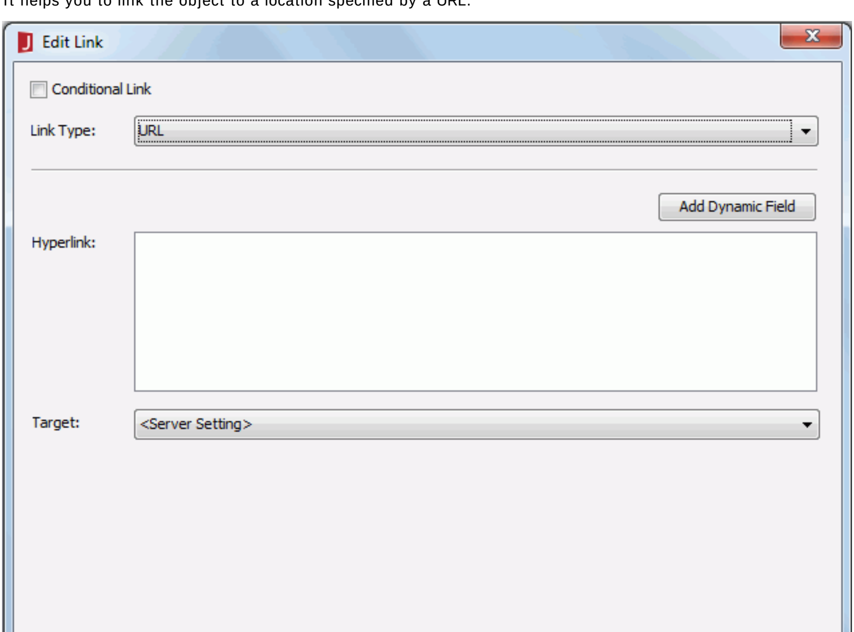
- **Parameters (Detail Report)**
  Lists all parameters contained in the datasets of the detail report.
- **Column (Master Report)**
  Lists all the DBFields, formulas and summaries in the datasets of the master report which are of the same data type as the parameters of the detail report. Select the field the value of which you want to assign to parameter of the detail report from the drop-down list.

Use the same encoding and DB setting for the detail report as that of the master report

Specifies whether the detail report uses the same encoding and DB settings as per the master report.

URL

It helps you to link the object to a location specified by a URL.



Hyperlink

Specifies the URL of the location that the object will be linked to.

- **Add Dynamic Field**
  Opens the Select Field dialog to insert a field into the URL.

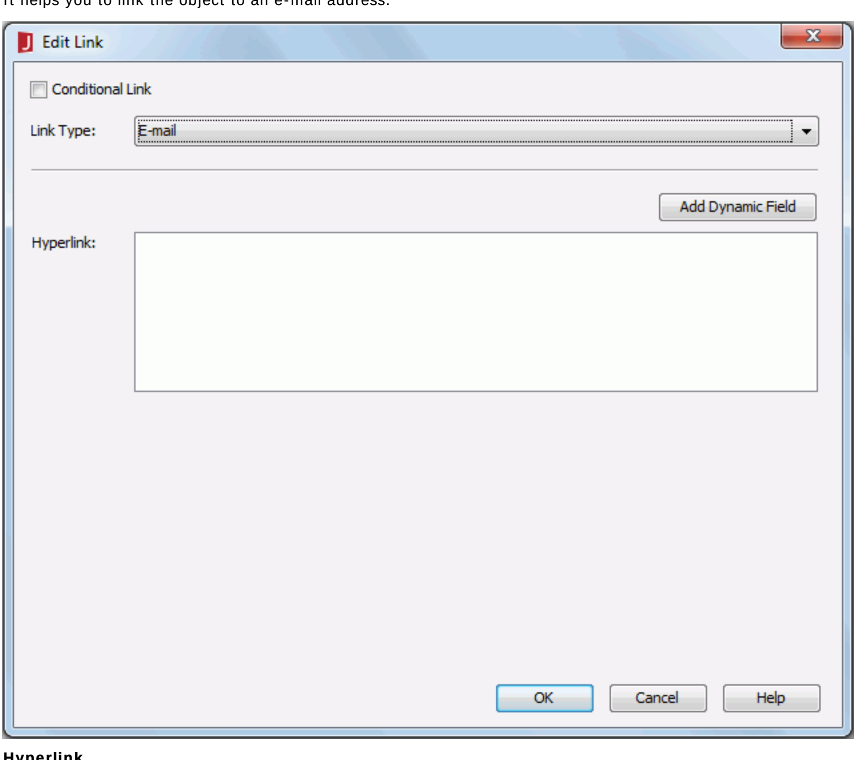
Target

Specifies the window or frame in which to load the location specified by the URL. Applied only when the link is triggered in Page Report Studio.

- **<Server Setting>**
  Loads the linked file according to setting of the Pop Up New Window for Links option in the Profile > Configure Profile > Page Report Studio > Properties > Default tab on JReport Server.
- **Same Frame**
  Loads the linked file into the same frame as the link.
- **Whole Window**
  Loads the linked file into the full browser window.
- **New Window**
  Loads the linked file into a new window. This window is not named.
- **Parent Frame**
  Loads the linked file into the parent frame of the frame that contains the link.
- **Other Frame**
  Loads the linked file into some other specified frame. If the frame name does not exist, the linked file will be loaded into a new window.

E-mail

It helps you to link the object to an e-mail address.



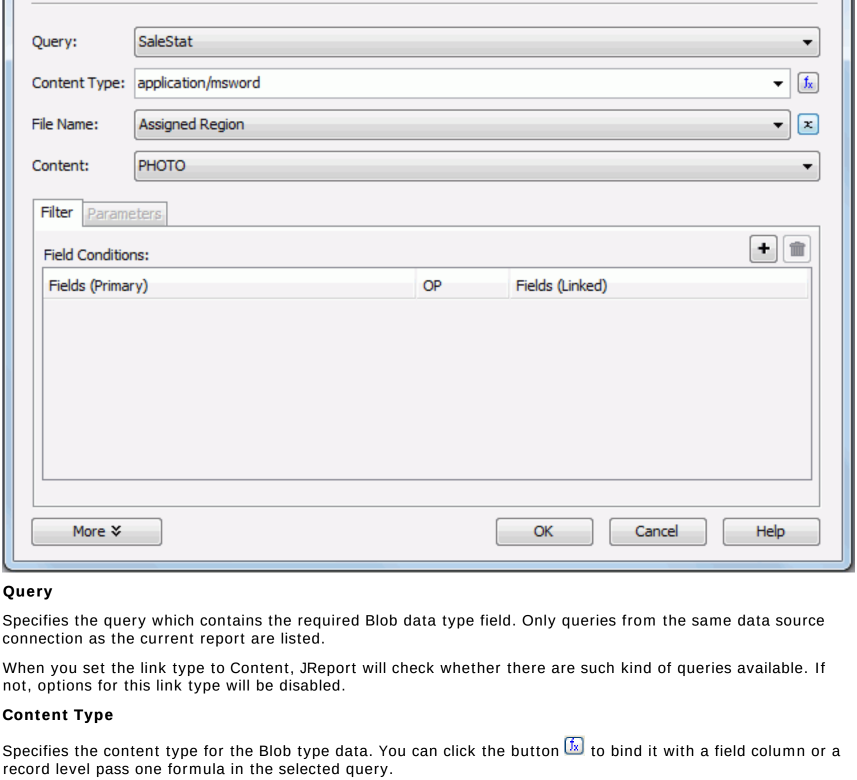
Hyperlink

Specifies the e-mail address that the object will be linked to. Click here for details about how to compose the e-mail address.

- **Add Dynamic Field**
  Opens the Select Field dialog to insert a field into the e-mail address.

Content

It helps you to link the object to a Blob data type field. The Blob data type field will be bound to the object and displayed as a hyperlink. You can download the Blob content by clicking this hyperlink. In JReport, Blob could be image, Binary, Blob, Clob, LongBlob, LongClob, Varbinary, Longvarbinary, and so on.



Query

Specifies the query which contains the required Blob data type field. Only queries from the same data source connection as the current report are listed.

When you set the link type to Content, JReport will check whether there are such kind of queries available. If not, options for this link type will be disabled.

Content Type

Specifies the content type for the Blob type data. You can click the button fx to bind it with a field column or a record level pass one formula in the selected query.

File Name

Specifies the file name for the linked Blob type data. You can click the button fx to bind it with a field column or a record level pass one formula in the selected query.

Content

Specifies the Blob data type field in the selected query.

Filter tab

Specifies the link conditions between the query used by the current report and the query that contains the linked Blob content.

- **Field Conditions**
  Lists the filter conditions between the two queries.
  - **Fields(Primary)**
    Lists the selected fields of the query used by the current report.
  - **OP**
    Specifies the operator to compose the condition. It can be "=", "<>", "<", ">", "<=", ">=" or "IN".
  - **Fields(Linked)**
    Lists the DBFields in the datasets of the query that contains the linked Blob content, which are of the same data type as the selected fields in the query used by the current report.
  - Adds a new condition line.
  - Removes the selected condition.

Parameters tab

This tab is available when the query that contains the linked Blob content uses parameters. It helps you to assign values to the parameters automatically.

- **Parameters**
  Lists all the parameters the query that contains the linked Blob content uses.
- **Value**
  Lists all the DBFields, formulas, summaries and parameters in the query used by the current report, which are of the same data type as the parameters the query that contains the linked Blob content uses. Select the field the value of which you want to assign to a parameter from the drop-down list.

Related topic:

- Binding a link to a label

Edit Link dialog for web report/library component

The dialog appears when you select an object in a web report or library component, which is defined with a link, right-click it and select Edit Link from the shortcut menu. It helps you to edit the link on the object.

Conditional Link

Specifies whether it is a conditional link. If checked, the object can be linked to different targets based on different conditions.

- Adds a new condition in the Edit Conditions dialog.
- Edits the selected condition.
- Removes the selected condition.
- Moves the selected condition one step up.
- Moves the selected condition one step down.
- **Others**
  If checked, you can define to which target the object will be linked when none of the conditions you have specified is met.

Link Type

Specifies the type of the link target for the object if it is not a conditional link or for the object under the selected condition if it is a conditional link. It can be one of the following:

- Report
- URL
- E-mail
- Content

More

Displays more link options. Available only for the Report, Master/Detail Report and Content link types.

OK

Accepts the link editing result and closes the dialog.

Cancel

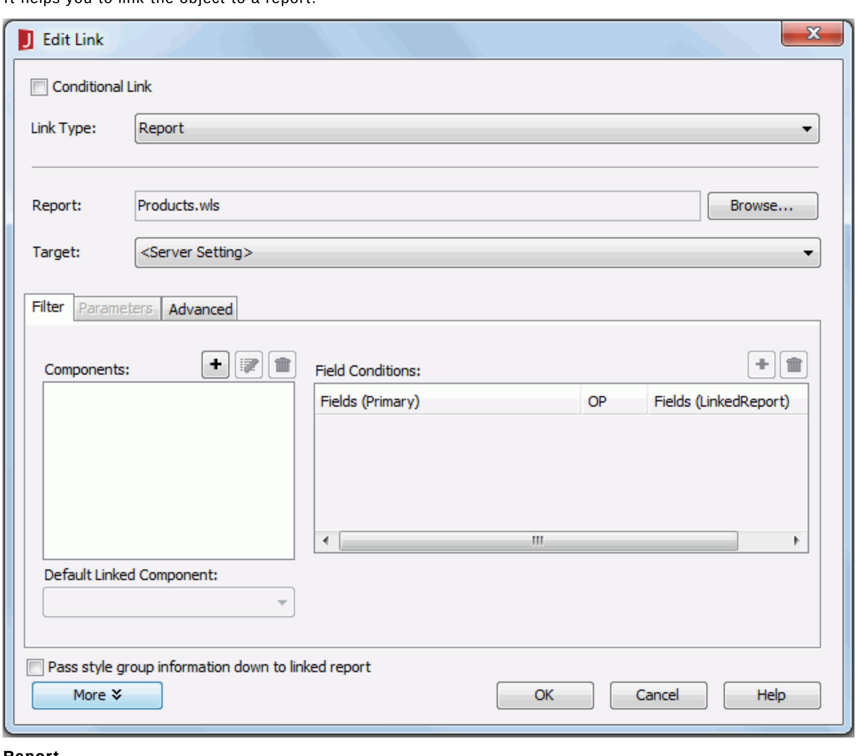
Cancels the link editing process and exits the dialog.

Help

Displays the help document about this feature.

Report

It helps you to link the object to a report.



Report

Specifies the target report. Click the Browse button to modify the desired report in the Select a Report dialog.

Target

Specifies the window or frame in which to load the linked report. Applied only when the link is triggered in Web Report Studio; when triggered in JDashboard, except for Same Frame, all the other targets are treated as New Window.

- **<Server Setting>**
  Loads the linked report according to setting of the Target of Detail Table Report and Links option in the Profile > Configure Profile > Page Report Studio > Properties > Default tab on JReport Server.
- **Same Frame**
  Loads the linked report into the same frame as the primary report.
- **Whole Window**
  Loads the linked report into the full browser window.
- **New Window**
  Loads the linked report into a new window. This window is not named.
- **Parent Frame**
  Loads the linked report into the parent frame of the frame in which the primary report is.
- **Other Frame**
  Loads the linked report into some other specified frame. If the frame name does not exist, the linked report will be loaded into a new window.

Filter tab

Specifies the filter conditions based on which the link relationship between the primary report and the linked report is set up.

- **Components**
  Specifies the components in the linked report that will be interlinked with the primary report.
  - Adds a component in the linked report to be interlinked with the primary report.
  - Edits the component in the linked report to be interlinked with the primary report.
  - Removes the selected component from the Component in Report Tab box.
- **Default Linked Component**
  Specifies the default linked component in the linked report.
- **Field Conditions**
  Lists the filter conditions between the primary report and the linked report.
  - **Fields(Primary)**
    Lists the selected fields of the primary report.
  - **OP**
    Specifies the operator to compose the condition. It can be "=", "<>", "<", ">", "<=", ">=" or "IN".
  - **Fields(LinkedReport)**
    Lists the DBFields in the datasets of the linked report which are of the same data type as the selected fields in the primary report.
  - Adds a new condition line.
  - Removes the selected condition.

Parameters tab

This tab is available when the linked report uses parameters. It helps you to assign values to the parameters of the linked report automatically.

- **Parameters**
  Lists all parameters contained in the datasets of the linked report.
- **Value**
  Lists all the DBFields, formulas, summaries and parameters in the datasets of the primary report which are of the same data type as the parameters of the linked report. Select the field the value of which you want to assign to parameter of the linked report from the drop-down list.

Advanced tab

Specifies to pass the values of the filter objects such as filter controls and sliders in the primary report to those in the linked report.

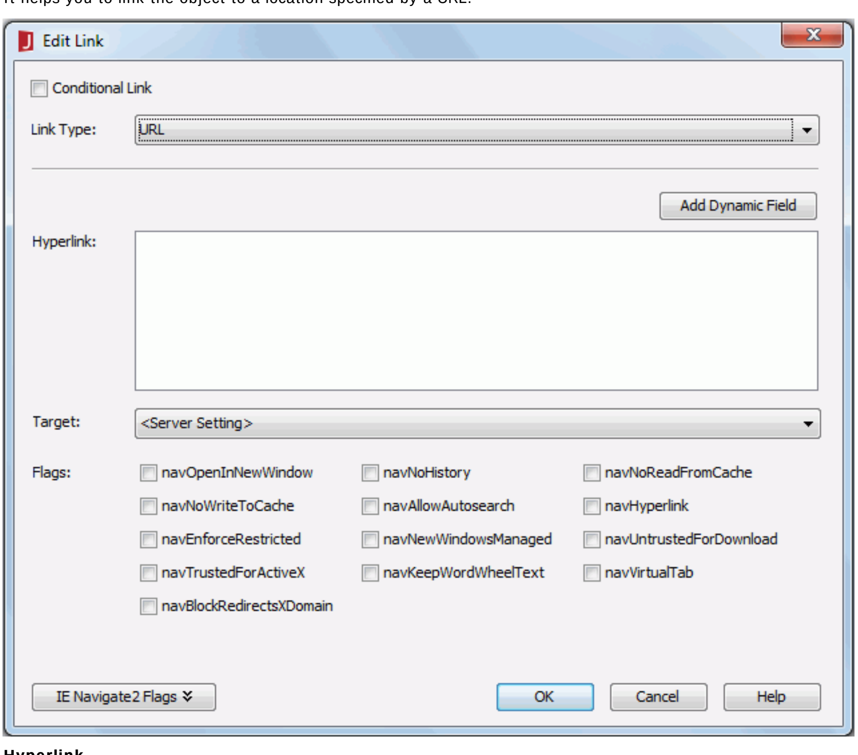
- **Primary Report Property/Object**
  Displays the filter objects in the primary report the values of which will be passed to those in the linked report. Click to select a filter object in the primary report in the Select Value dialog.
- **Linked Report Property/Object**
  Displays the filter objects in the linked report which will adopt the values passed from filter objects in the primary report. Click to select a filter object in the linked report in the Select Value dialog. You need to make sure that values of the filter object in the primary report can be applied to the filter object in the linked report in the same line.
- Adds a new line to pass values between the filter objects in the primary report and the linked report.
- Removes the selected line.

Pass style group information down to linked report

Specifies whether or not to apply the style group of the primary report to the linked report.

URL

It helps you to link the object to a location specified by a URL.



Hyperlink

Specifies the URL of the location that the object will be linked to.

- **Add Dynamic Field**
  Opens the Select Field dialog to insert a field into the URL.

Target

Specifies the window or frame in which to load the location specified by the URL. Applied only when the link is triggered in Web Report Studio; when triggered in JDashboard, except for Same Frame, all the other targets are treated as New Window.

- **<Server Setting>**
  Loads the linked file according to setting of the Target of Detail Table Report and Links option in the Profile > Configure Profile > Page Report Studio > Properties > Default tab on JReport Server.
- **Same Frame**
  Loads the linked file into the same frame as the link.
- **Whole Window**
  Loads the linked file into the full browser window.
- **New Window**
  Loads the linked file into a new window. This window is not named.
- **Parent Frame**
  Loads the linked file into the parent frame of the frame that contains the link.
- **Other Frame**
  Loads the linked file into some other specified frame. If the frame name does not exist, the linked file will be loaded into a new window.

IE Navigate2 Flags

Specifies to use the IE Navigate2 method and its flags to control the behavior of IE when loading the linked file with Internet Explorer. Available only when the link is triggered in a table in Web Report Studio.

Flags

Lists all the flags of the IE Navigate2 method.

To make sure the navigate2 flags work well, uncheck the option **Enable Protected Mode** in Internet Options dialog > Security tab of the Internet Explorer browser, then click the **Custom level** button, in the Security Settings dialog, enable all ActiveX relative options under the ActiveX controls and plug-ins.

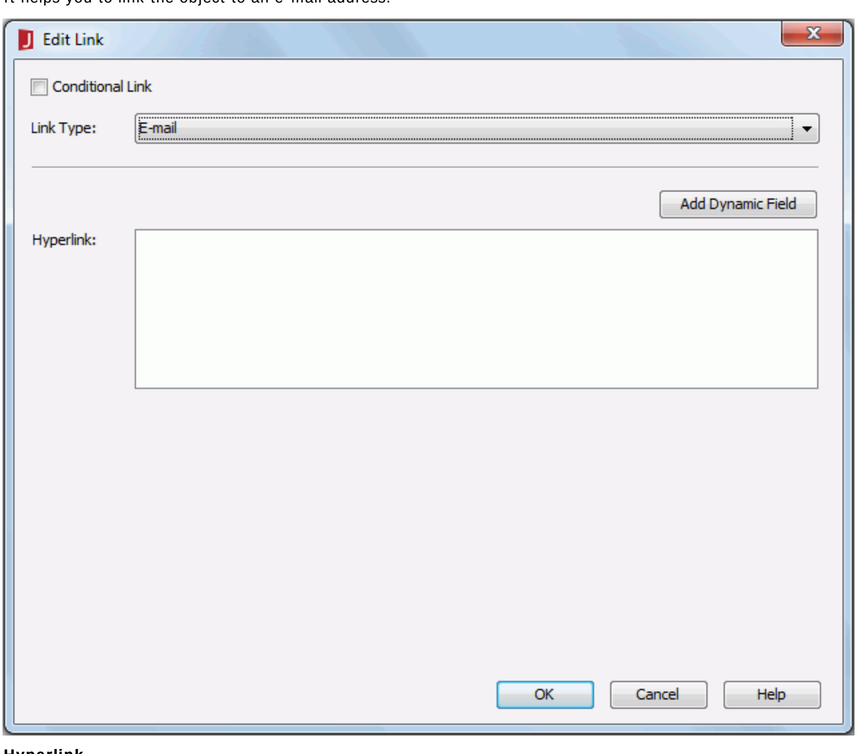
You can see that multiple flags can be selected at a time. Some of them may conflict with each other. The conflict relationship between the flags strictly follows the Microsoft MSDN specification. JReport loyally delivers the user state to the Microsoft interface and does not present or check the conflict.

When you create a new link to a URL, with none of the Navigate2 flags being checked, the linked file will be opened in the specified window if a window name is provided via the "Other Frame or Window" setting in the Target field. If the window name is not given, the linked file will be opened in a new window.

- **navOpenInNewWindow**
  Opens the resource or file in a new window.
- **navNoHistory**
  Does not add the resource or file to the history list. The new page replaces the current page in the list.
- **navNoReadFromCache**
  Not implemented.
- **navNoWriteToCache**
  Not implemented.
- **navAllowAutosearch**
  If the navigation fails, the auto search functionality attempts to navigate common root domains (.com, .edu, and so on). If this also fails, the URL is passed to a search engine.
- **navHyperlink**
  Internet Explorer 6 for Windows XP SP2 and later. If the navigation fails when a hyperlink is being followed, this constant specifies that the resource should then be bound to the moniker using the BINDF_HYPERLINK flag.
- **navEnforceRestricted**
  Internet Explorer 6 for Windows XP SP2 and later. Forces the URL into the restricted zone.
- **navNewWindowsManaged**
  Internet Explorer 6 for Windows XP SP2 and later. Uses the default Popup Manager to block pop-up windows.
- **navUntrustedForDownload**
  Internet Explorer 6 for Windows XP SP2 and later. Blocks files that normally trigger a file download dialog box.
- **navTrustedForActiveX**
  Internet Explorer 6 for Windows XP SP2 and later. Prompts for the installation of ActiveX controls.
- **navKeepWordWheelText**
  For Internet Explorer 7. Maintains state for dynamic navigation based on the filter string entered in the search band text box (wordwheel). Restores the wordwheel text when the navigation completes.
- **navVirtualTab**
  For Internet Explorer 8. Opens the resource as a replacement for the current or target tab. The existing tab is closed while the new tab takes its place in the tab bar and replaces it in the tab group, if any. Browser history is copied forward to the new tab. On Windows Vista, this flag is implied if the navigation would cross integrity levels and navOpenInNewTab, navOpenInBackgroundTab, or navOpenInNewWindow is not specified.
- **navBlockRedirectsXDomain**
  For Internet Explorer 8. Blocks cross-domain redirect requests. The navigation triggers the DWebBrowserEvents2::RedirectXDomainBlocked event if blocked.

E-mail

It helps you to link the object to an e-mail address.



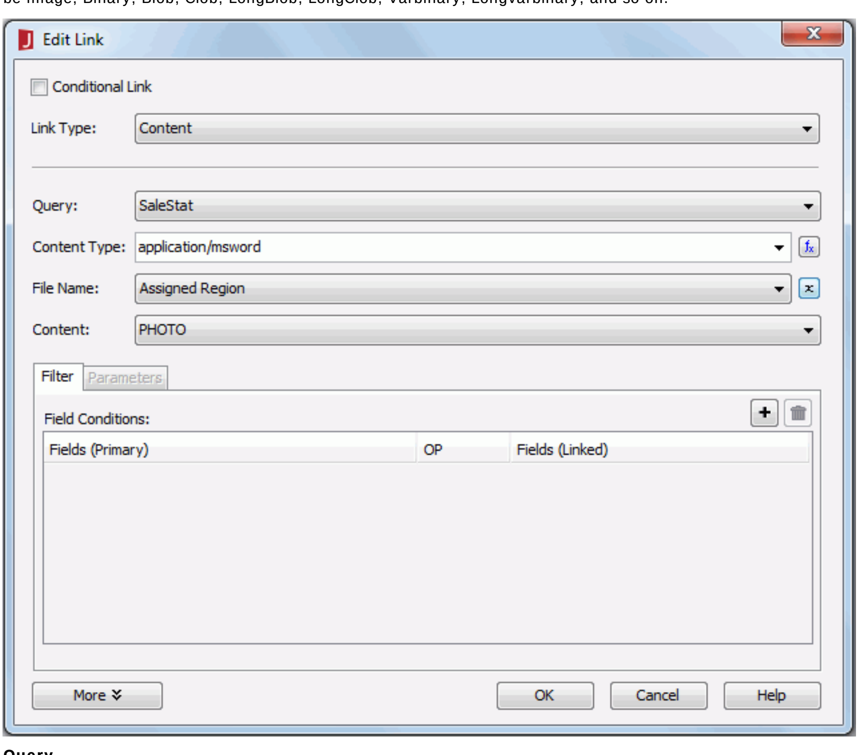
Hyperlink

Specifies the e-mail address that the object will be linked to. Click here for details about how to compose the e-mail address.

- **Add Dynamic Field**
  Opens the Select Field dialog to insert a field into the e-mail address.

Content

It helps you to link the object to a Blob data type field. The Blob data type field will be bound to the object and displayed as a hyperlink. You can download the Blob content by clicking this hyperlink. In JReport, Blob could be image, Binary, Blob, Clob, LongBlob, LongClob, Varbinary, Longvarbinary, and so on.



Query

Specifies the business view which contains the required Blob data type field. Only query based business views from the same data source connection as the current report are listed.

When you set the link type to Content, JReport will check whether there are such kind of business views available. If not, options for this link type will be disabled.

Content Type

Specifies the content type for the Blob type data. You can click the button to bind it with a string type business view element or a record level pass one formula in the selected business view, or with a dynamic formula in the current report.

File Name

Specifies the file name for the linked Blob type data. You can click the button to bind it with a string type business view element or a record level pass one formula in the selected business view.

Content

Specifies the Blob data type field in the selected business view.

Filter tab

Specifies the link conditions between the business view used by the current report and the business view that contains the linked Blob content.

- **Field Conditions**
  Lists the filter conditions between the two business views.
  - **Fields(Primary)**
    Lists the selected fields of the business view used by the current report.
  - **OP**
    Specifies the operator to compose the condition. It can be "=", "<>", "<", ">", "<=", ">=" or "IN".
  - **Fields(Linked)**
    Lists the DBFields in the datasets of the business view that contains the linked Blob content, which are of the same data type as the selected fields in the business view used by the current report.
  - Adds a new condition line.
  - Removes the selected condition.

Parameters tab

This tab is available when the business view that contains the linked Blob content uses parameters. It helps you to assign values to the parameters automatically.

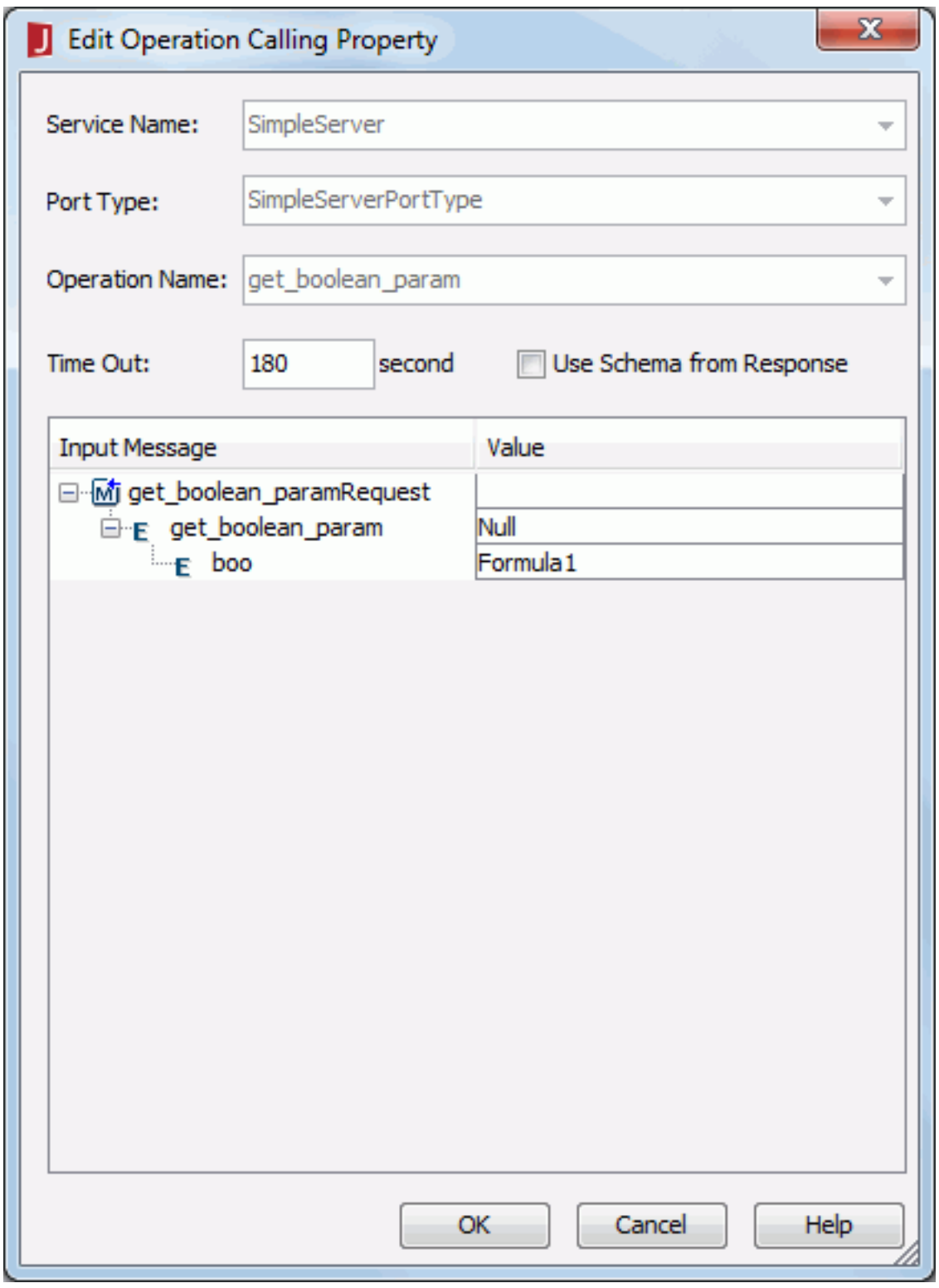
- **Parameters**
  Lists all the parameters the business view that contains the linked Blob content uses.
- **Value**
  Lists all the DBFields, formulas, summaries and parameters in the business view used by the current report, which are of the same data type as the parameters the business view that contains the linked Blob content uses. Select the field the value of which you want to assign to a parameter from the drop-down list.

Related topic:

- Binding a link to a label

Edit Operating Calling Property dialog

The dialog appears when you right-click an operation node under the Tables node in a web service connection and select Edit from the shortcut menu in the Catalog Browser. It helps you to edit the information of the specified operation.



Service Name

Displays the service name you have selected to create the table.

Port Type

Displays the port type for the selected service. Available only when the web service is defined by WSDL 1.1.

Operation Name

Displays the selected operation name.

Time Out

Specifies how long to wait for the operation to complete.

Use Schema from Response

This option is available only when the web service is defined by WSDL 1.1.

When the XML schema in the output message does not match the specific XML instance in the SOAP responded from web service, JReport will not be able to read data properly from that XML instance. In this condition, you can use this option to control whether to ignore the XML schema in the output message and parse the XML schema directly from the specific XML instance in the SOAP for the web service.

Input Message

Lists the input message information for the selected operation.

Value

Specifies the value for the input message. You can either input the value, or define a constant level formula or parameter to be the value of the input message.

OK

Accepts the editing operation result and closes the dialog.

Cancel

Cancels the editing operation process and exists the dialog.

Help

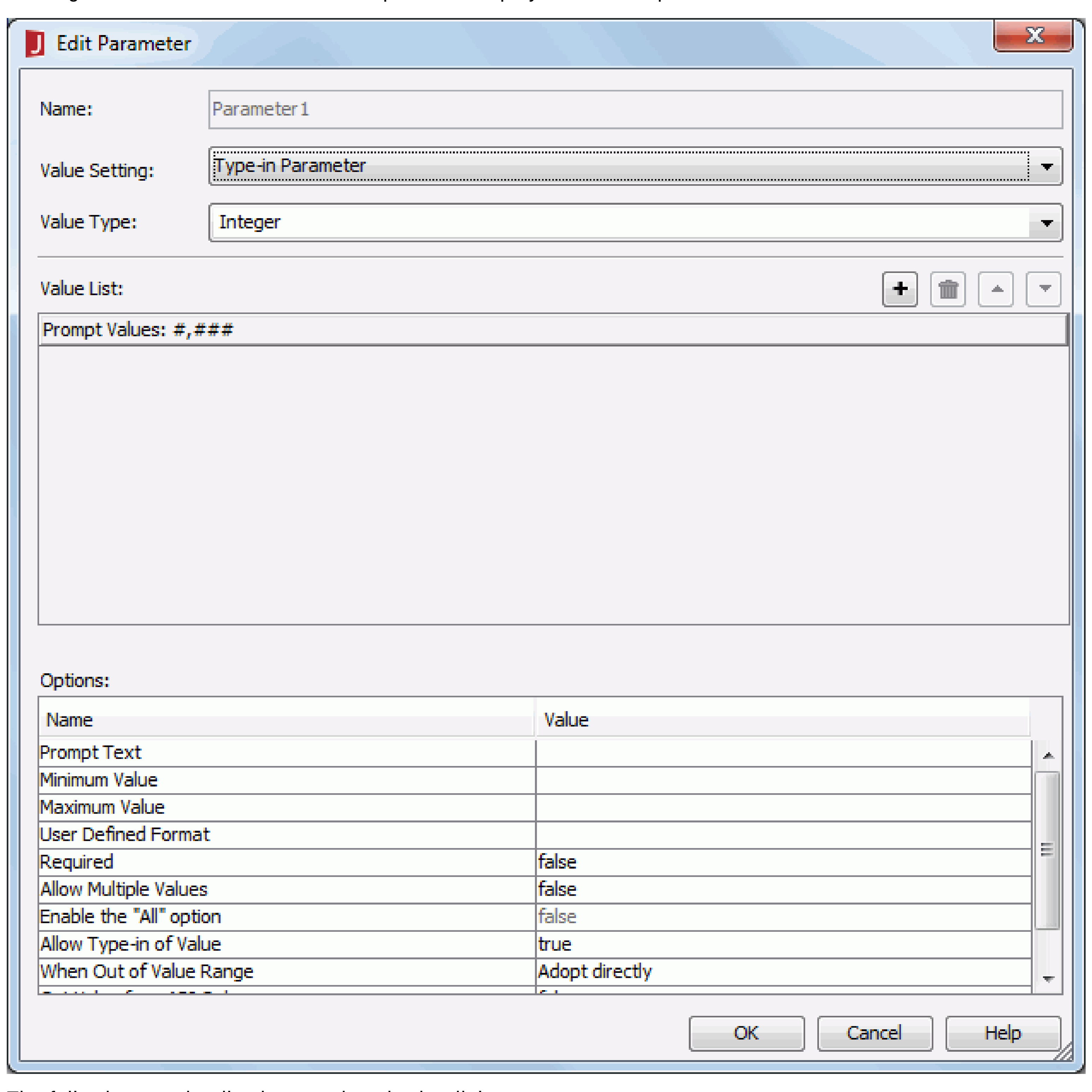
Displays the help document about this feature.

Related topics:

- Editing the operation used to create tables

Edit Parameter dialog

This dialog appears when you right-click a parameter and select Edit Parameter from the shortcut menu in the Catalog Browser or the Resource View panel. It helps you to edit a parameter.



The following are details about options in the dialog:

Name

Displays the name of the parameter.

Value Setting

Specifies the parameter type.

- **Type-in Parameter**
  If selected, you can predefine parameter values by typing manually.
- **Bind with Single Column**
  If selected, the parameter will be bound with a DBField. The values of the DBField will be retrieved as the parameter values at runtime. Not available for parameters in XML connection.
- **Bind with Cascading Columns**
  If selected, you can create a group of cascading parameters so as to achieve the function of filtering parameters with parameters in a simple way. Not available when the parameter is in a data source that contains XML connections.

Value Type

Specifies the data type of the parameter.

Values

The section after Value Type varies with the type you select from the Value Setting drop-down list.

- **For Type-in Parameter**
  - **Value List**
    Lists the predefined parameter values.
  - **Prompt Values** [format hint]
    Predefines the parameter values. The [format hint] shows what the predefined values should look like.

    There can be more than one prompt value. All the prompt values must be of the same type as specified in the Value Type field.
    - Adds a new prompt value to the list. Double-click in the value line to edit the value. If the parameter is of Date, DateTime, or Time type, you can also click the calendar button to set a date and time value in the Calendar dialog.
    - Removes the selected prompt value.
    - Adjusts the display sequence of selected value by moving it up a step.
    - Adjusts the display sequence of selected value by moving it down a step.
- **For Bind with Single Column**
  JReport provides such a mechanism that links each field value of Display Column with the exact value of Bind Column. When you select the field value from Display Column, the field value of Bind Column is actually sent to the query and filters the query result. It helps you to bind the parameter to a DBField (column) in order to provide a list of DBField values for users to select at runtime. The selected value of the bound DBField will be used as the parameter value. Additionally, instead of displaying the values of the bound one, you can also choose to display values of other DBField, which probably makes more sense to the report end users. For example, it might be confusing if you provide a list of customer ID numbers for users to select at runtime, since the ID numbers could probably mean nothing to them. In cases like this, it is wise for you to display values of other fields, which could probably make more sense. For this case, you might prefer to display customer names instead of ID numbers. When the report end users select a customer name from the list, its ID number is passed to the query as the parameter value so that the search criteria can be fulfilled. So, before the report with a parameter bound to a DBField is processed, a query statement will be executed first to retrieve the values of the bound field and display field.
  - **Source**
    Specifies the source type from which to retrieve the columns which can be used to bind with the parameter. The available source types are: Tables and Views (including synonyms), Stored Procedures, Imported SQLs, and User Defined. However, if the catalog data source in which the parameter is contains only a MongoDB connection, only Tables and User Defined are available in the Source drop-down list.
  - **Bind Column**
    Specifies the DBField to be bound with the parameter.
  - **Display Column**
    When viewing a report which contains the parameter, the field values related to the column name you selected in the drop-down list of Display Column will be displayed in the Enter Parameter Values dialog at runtime.
- **For Bind with Cascading Columns**
  - **Data Source**
    Specifies the data source from which to retrieve the columns which can be used to bind with the parameters. The available data sources are tables, views, synonyms, stored procedures, imported SQLs, and user defined data sources in the catalog data source in which the parameter is. However, if the catalog data source contains only a MongoDB connection, only tables and user defined data sources are available.
  - **Value List**
    Defines a group of cascading parameters.
    - Adds a new parameter.
    - Removes the selected parameter.
    - Adjusts the parameter level by moving it up a step. The higher the position is, the higher the level is. The lower level field values are controlled by the higher level field values.
    - Adjusts the parameter level by moving it down a step.
    - **Bind Column**
      Specifies the DBField to be bound with the parameter.
    - **Display Column**
      When viewing a report which contains the parameter, the field values related to the column name you selected in the drop-down list of Display Column will be displayed in the Enter Parameter Values dialog at runtime.
    - **Parameter**
      Specifies whether to set a group of Bind Column and Display Column as a parameter which will become a member of the cascading parameter group. All parameters in the cascading group will be created as independent parameters in the catalog after you finish the dialog.

      The parameter name is provided by JReport automatically, and the format is *[The name specified as parameter name]-[Bind Column name]*.

Options

Specifies options for the parameter. The options vary with the type you select from the Value Setting drop-down list.

- **Prompt Text**
  Specifies the text for prompting users to provide the value of the parameter.

  Tips:
  - To help users avoid typing a value out of range, you can add the value range to the end of the prompt text. For example, "Type a customer ID (1~100):".
  - To help users avoid typing a value in a wrong pattern, you can add the value pattern to the end of the prompt text. For example, "Type an order date (MMM-dd-yy):".
- **Minimum Value/Length**
  Specifies the minimum value allowed for the parameter. If the parameter value type is String, this option is defined as Minimum Length, which means that you can set the minimum allowed length of the string.
- **Maximum Value/Length**
  Specifies the maximum value allowed for the parameter. If the parameter value type is String, this option is defined as Maximum Length, which means that you can set the maximum allowed length of the string.
- **String Encoding**
  Specifies the encoding for the parameter. Available only when the parameter's data type is defined as String.
- **User Defined Format**
  For certain types of parameters, the value format supplied by JReport may not satisfy your requirement. In this case, you can define your own preferred format. When you have defined the value format, all the values you supply at runtime should be based on it. Otherwise there will be an error message. For details, see User defined format for parameters.

  Notes:
  - The letters that stand for the user defined format should follow the JDK standard. In addition, the user defined format only limits the format of the input value, which does not mean that the display value will be in the same format. To change the display format, you will need to specify the Format property of this parameter in the Report Inspector.
  - The user defined format will show different appearance under different locale, and you need input values according to the displayed format. For example, under the Spanish or French locale, ####.## will be displayed as #0,## and you should follow #0,## to input values.
- **Required**
  If set to true, the parameter will be treated as a required parameter. A required parameter must be fulfilled, which means that you must supply a value in the Enter Parameter Values dialog before the report can be run. If set to false, the parameter will be treated as an optional parameter. An optional parameter can be left blank, which means you can either type a value, or leave it as it is.

  In the Enter Parameters Value dialog, the prompt text of a required parameter will be marked with a * sign.

  Note: If the parameter is optional and you do not expect it to take effect in your report, in the Enter Parameter Values dialog, clear the content in the value field. Otherwise, any value left in the field will function in reports.
- **Hide Parameter When Single Value Returned**
  Specifies to hide the parameter at runtime when only a single value is returned for the parameter. The returned single value will be used as default value of the parameter.
- **Allow Multiple Values**
  Enables multiple parameter values to be selected. If it is set to true, when running a report with the parameter, the Enter Values dialog is provided for you to specify multiple values.

  Note: Multi-value parameters can only be supported in dataset filters, query filters, imported SQL files, and subreport/link report conditions.
- **Enable the "All" Option**
  Displays the All checkbox in the Enter Values dialog which means that the parameter value result is all the values that match the parameter data type. Available when Allow Multiple Values is set to true.
- **Treat Blank as Null**
  Available only when the parameter's data type is defined as String. If it is true, the parameter value will be null; if false, the default parameter value is "".
- **Use Current Date-Time**
  If set to true, the current date time in your system will be used as value of the parameter. Available only when the parameter's data type is defined as Date, Time or DateTime.
- **Use Current Date-Time When Blank**
  If set to true, when the value of the parameter is blank, the current date time in your system will be used as value of the parameter. Available only when the parameter's data type is defined as Date, Time or DateTime.

  If the type of the parameter is set to Date, Time or DateTime, you can choose whether to get parameter value dynamically from the system's date time by enabling one of the above options - Use Current or Use Current When Blank. However, if you want to use this feature, you need to make sure your system's DateTime format is set to yyyy-MM-dd H:mm:ss as it is the only format JReport supports; otherwise, you need to make use of the User Defined Parameter Format feature. Also, if one of the two options is enabled, you need to provide the parameter's prompt value according to the data type of your parameter.
  - When the data type of the parameter is Date, you can specify the value according to the format yyyy-MM-dd.
    - yyyy-01-01: Year will be dynamically changed.
    - 2007-MM-01: Month will be dynamically changed.
    - 2007-01-dd: Date will be dynamically changed.
    - yyyy-MM-01: Year and month will be dynamically changed.
    - yyyy-01-dd: Year and date will be dynamically changed.
    - yyyy-MM-dd: Year, month, and date will be dynamically changed.
    - 2007-MM-dd: Month and date will be dynamically changed.
  - When the data type of the parameter is Time, you can specify the value according to the format H:mm:ss.
    - H:00:00: Hour will be dynamically changed.
    - 12:mm:00: Minute will be dynamically changed.
    - 12:00:ss: Second will be dynamically changed.
    - H:mm:00: Hour and minute will be dynamically changed.
    - 12:mm:ss: Minute and second will be dynamically changed.
    - H:00:ss: Hour and second will be dynamically changed.
    - H:mm:ss: Hour, minute and second will be dynamically changed.
  - When the data type of the parameter is DateTime, specify the value in a combination of the above.
- **Record Level Security**
  Specifies the RLS policy defined on the bound column in the data source to the parameter. For details, refer to Applying RLS to a parameter.
- **Sort**
  Specifies how to sort values of cascading parameters.
  - **Ascend**
    Sorts parameter values in an ascending order.
  - **Descend**
    Sorts parameter values in a descending order.
  - **No Sort**
    The parameter values are displayed in the original order as in database.
  - **Sort By**
    Sorts parameter values by sorting other fields in the same data source as the cascading group. When Sort By is selected, the Sort By dialog will be displayed for specifying fields and their sorting orders.
- **Distinct**
  If set to true, when viewing a report that contains the parameter, in the Enter Parameter Values dialog, identical values will appear only once and no duplicate values will be listed.
- **Import SQL**
  Displays the SQL statement for retrieving values of the bound column and display column. You can edit the statement if required. To do this, click in the value cell, then click and edit the statement in the Import SQL dialog.

  Notes:
  - If you have made changes to the SQL statement, the later changes for Bind Column and Display Column will not take effect unless you remove the SQL lines or bind the parameter all over again. To remove the SQL lines, click in the value cell and clear the content in the Import SQL dialog.
  - The special field User Name can also be used in the SQL statement as *@username*.
- **Allow Type-in of Value**
  Specifies whether to enable typing values besides selecting values when setting parameter values. The property is enabled for Type-in Parameter by default.
- **When Out of Value Range**
  Specifies how JReport will respond if the typed value is out of the predefined value list. The property is enabled only when the option Allow Type-in of Value is set to true.

  There are three solutions as follows:
  - **Adopt directly**
    The typed value will be used directly.
  - **Use default**
    The default value defined when creating the parameter will be used.
  - **Warn**
    Displays a warning message telling that "Value xxx is out of the available value list".

    Note: For a multi-value parameter, if one of the input values is not in the value list, the whole input result will be treated as out of range.
- **Get Value from API Only**
  Specifies whether the parameter values can be input via GUI or not.

  If true, the parameter values should only be obtained from API and there will be no parameter dialogs available on any JReport Designer or Server GUI for inputting values. Users will have to give the values of the parameter by API such as URLs, coding, and sessions, otherwise the default value will be used.

  If false, the parameter can be obtained from both GUI and API.

  If a member of a cascading parameter group is set to get values from API only, it means that all the higher level members shall get values from API only. This also applies to parameters that some other parameters depend on. If in JDashboard several parameters share values, when any of them is set to get values from API only, they will share the values from API.
- **On Parameter Value Change**
  Specifies to define an action that will be called when the parameter value changes. Choose a formula from the drop-down list to develop an action, which will be called before the other actions defined in the report or library component. It can work as a parameter value validation rule.

OK

Applies all changes and closes the dialog.

Cancel

Does not retain any changes and closes the dialog.

Help

Displays the help document about this feature.

Related topics:

- Creating a parameter

Edit Role dialog

The dialog appears when you select a role and then click in one of the following dialogs or tabs:

- Security dialog
- Edit Business View Security dialog
- Category Property dialog > Security tab
- Add View Element dialog > Security tab
- Edit View Element dialog > Security tab

It helps you to edit the role, and consists of the following tabs:

- General tab
- Relation tab

OK

Accepts all changes and closes the dialog.

Cancel

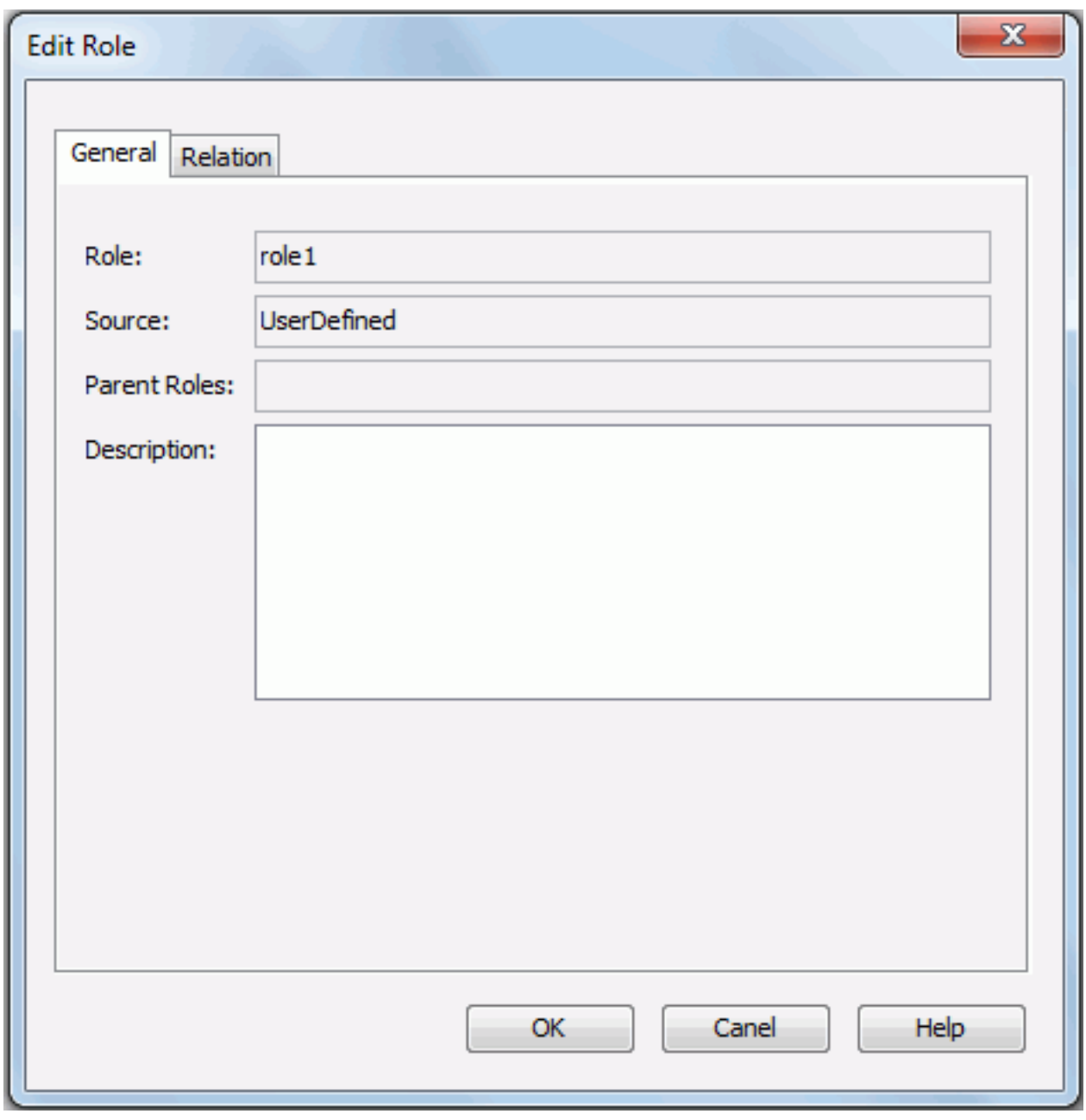
Does not retain any changes and closes the dialog.

Help

Displays the help document about this feature.

General tab

Specifies the role information.



Role

Specifies the name of the role.

Source

Displays the source of the role. It is read only.

Parent Roles

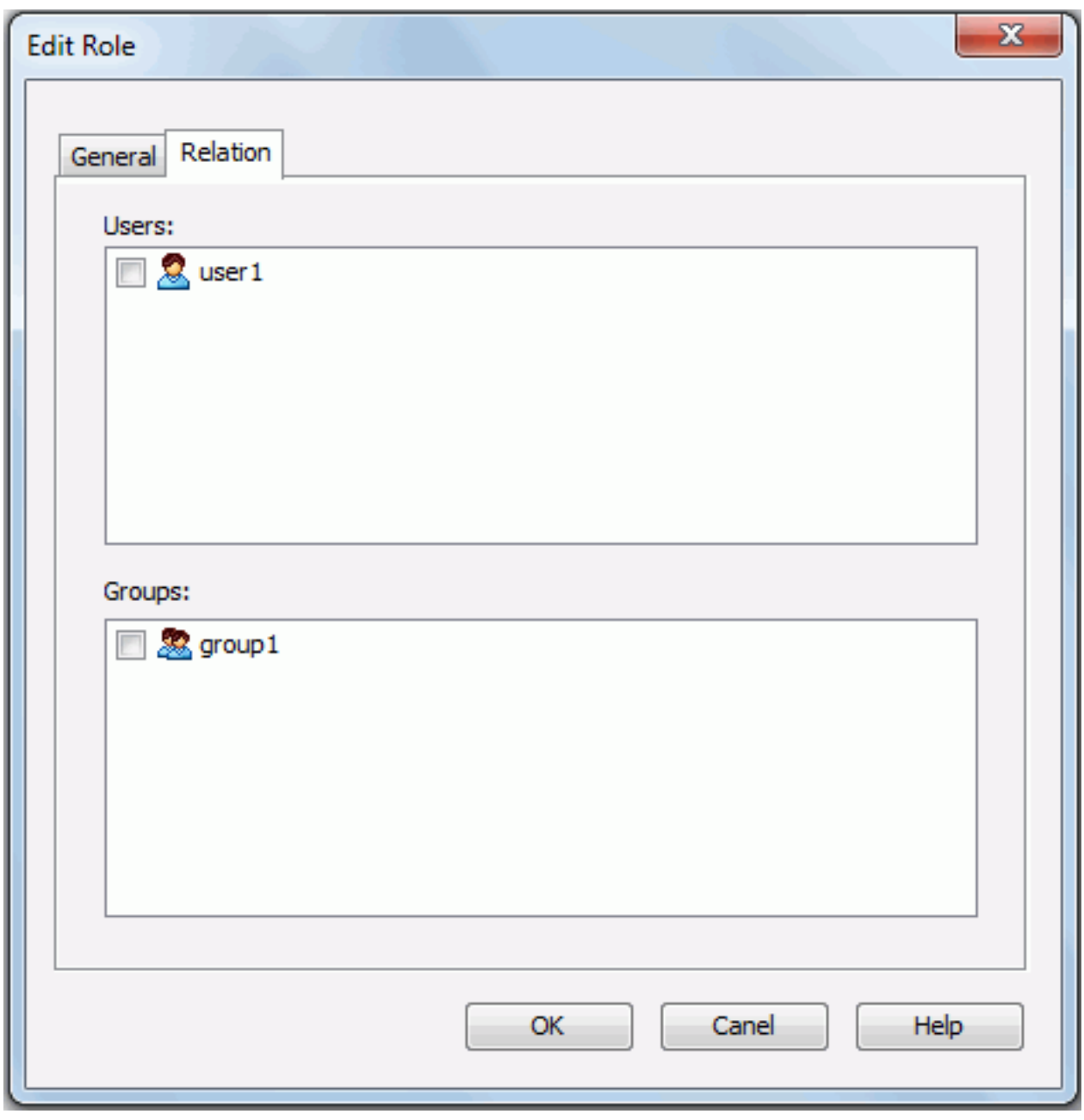
Displays the parent roles of the current role. It is read only.

Description

Specifies the description of the role.

Relation tab

Specifies the relationship of the role and the existing users and groups.



Users

All available users will be listed here. If a user is checked, the user will be added into the role.

Groups

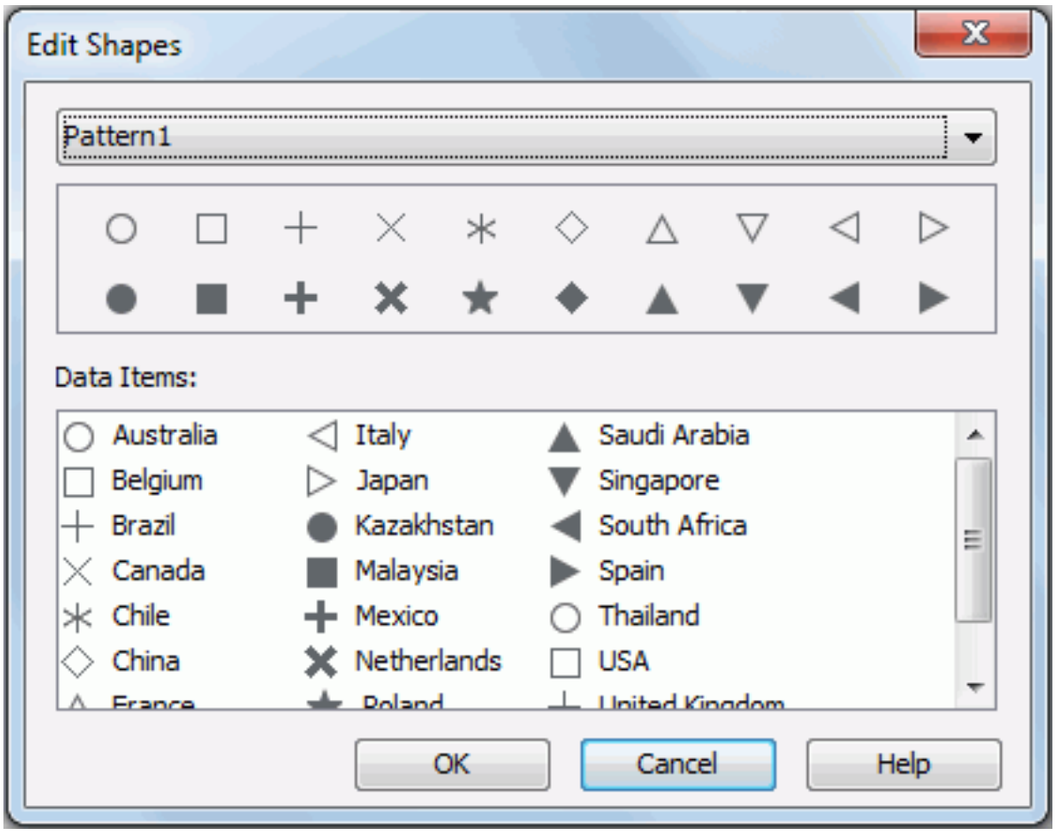
All available groups will be listed here. If a group is checked, the group will be added into the role.

Related topics:

- Setting up a connection-scope security policy
- Configuring member-level security

Edit Shapes dialog

The dialog appears when you click the button next to the Shape By text box when a field has been added to the box in the Display screen of the map wizard. It helps you to specify the shapes of the markers in a geographic map.



The following are details about options in the dialog:

Shape pattern drop-down list

Specifies the shape pattern to be used.

Preview box

Displays a preview of the shapes in the selected pattern.

Data Items

Lists the field values with the shapes in the selected pattern. You can also change the shape for each value by selecting it and then clicking a shape in the preview box.

If an aggregation field has been added to control the shapes, the Data Items box will be blank. You cannot customize the shapes and the default shapes in the selected pattern will be applied at runtime.

OK

Applies the specified shapes to the markers and closes the dialog.

Cancel

Cancels the operation and closes the dialog.

Help

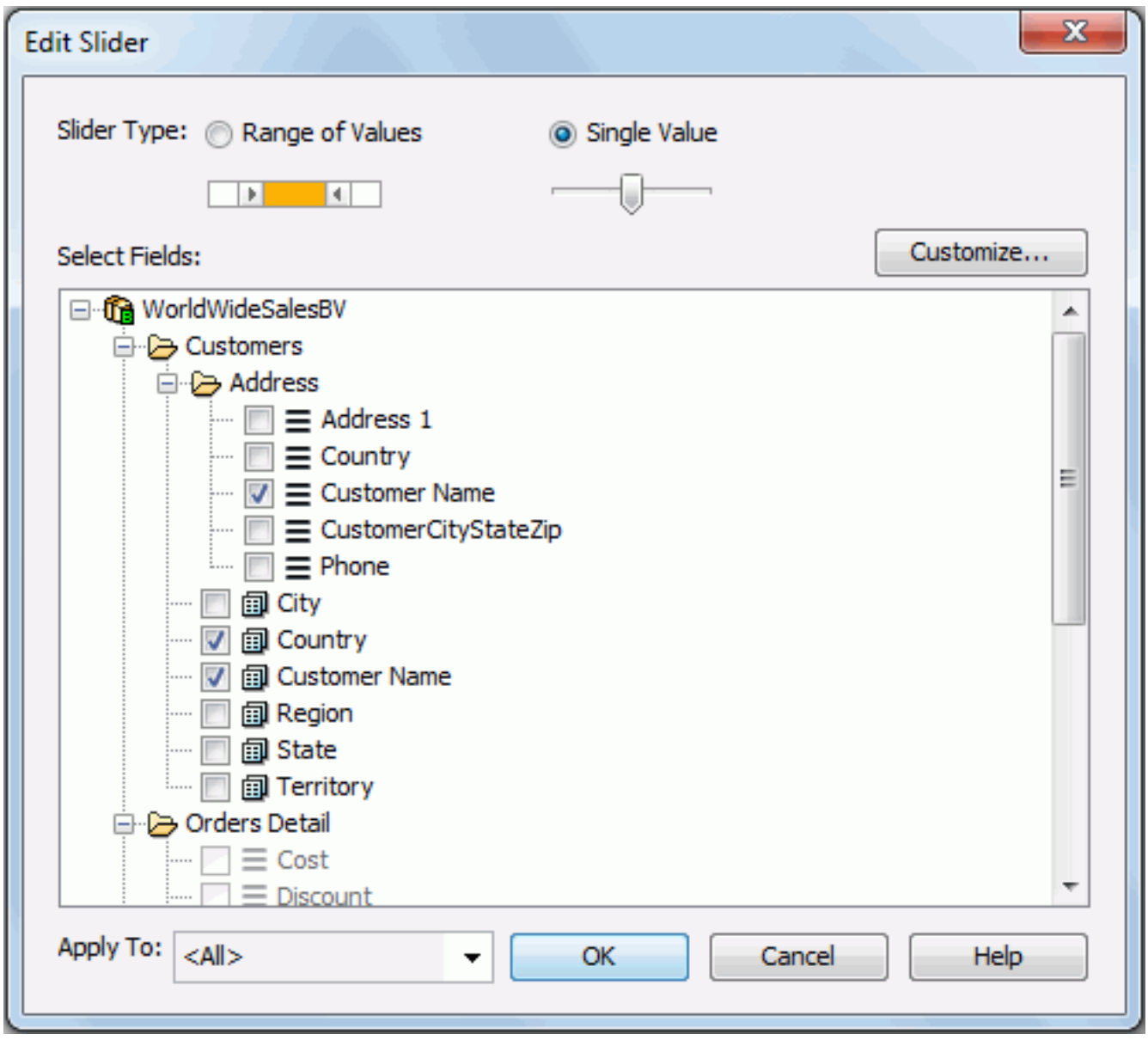
Displays the help document about this feature.

Related topics:

- Inserting a geographic map to a web report

Edit Slider dialog

The dialog appears when you right-click a slider and select Edit Slider from the shortcut menu. It helps you to edit the slider.



The following are details about options in the dialog:

Slider Type

Specifies the type of the slider.

- **Range of Values**
  If selected, you can select a range of values on the slider at runtime.
- **Single Value**
  If selected, you can select only one value on the slider at runtime, not available for numeric data type.

Select Fields

Specifies the fields to bind to the slider. All the selected fields should be of the same data type.

The common usage is to select one field, and then based on the field to filter the data of the components created from the same data source as the field.

To filter components using different data sources, choose a common field all the data sources contain and select the field in all the data sources.

Customize

Opens the Customize Value dialog to customize the values in the selected fields you want to use on the slider.

Apply To

Specifies the target data components you want to apply the slider to.

OK

Applies the changes to the slider and closes the dialog.

Cancel

Cancels the operation and closes the dialog.

Help

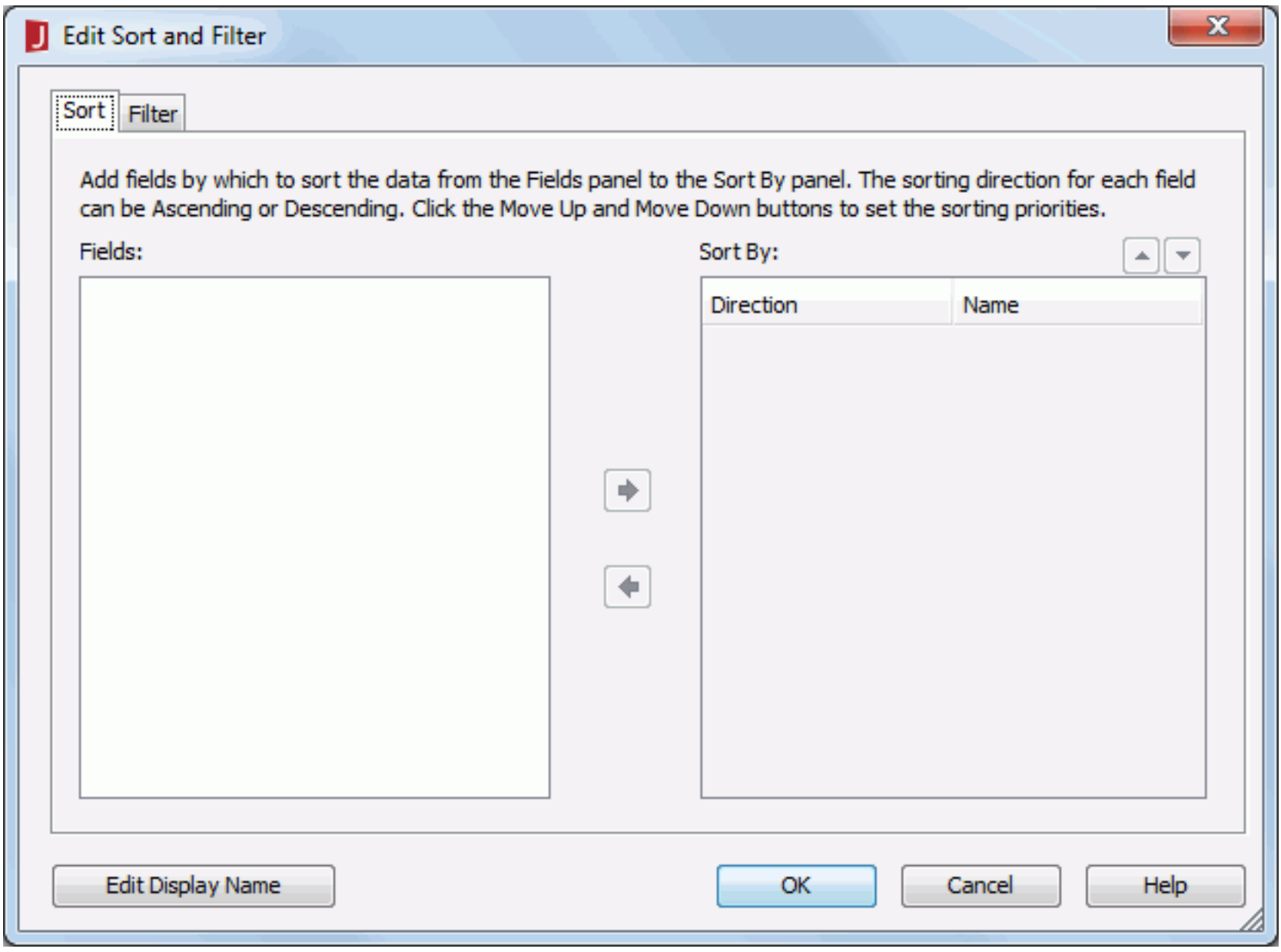
Displays the help document about this feature.

Related topics:

- Using slider or filter control to filter data

Edit Sort and Filter dialog

This dialog appears when you select a data component, which refer to banded object, table, chart and crosstab, in a page report, and click Report > Edit Sort and Filter on the menu bar. It helps you to predefine the sort and filter conditions for the data component.



The following are details about options in the dialog:

Sort tab

Defines the sorting conditions.

- **Fields**
  Lists all fields that are available to be sorted on.
-
  Adds the selected fields in the Fields box to the Sort By box.
-
  Removes the selected fields from the Sort By box.
- **Sort By**
  Lists the fields that are selected. Report data will be sorted according to the sort by fields you selected.
  - **Direction**
    Specifies the sort order of the specified field.
  - **Name**
    Displays the field name.
-
  Moves the selected field up one step. The order of the sort by fields determines the order in which the report data will be sorted.
-
  Moves the selected field down one step.

Filter tab

Defines the filtering conditions. For details about options in the tab, see Edit Filter dialog.

Edit Display Name

Opens the Edit Display Name dialog to add or remove display name fields.

OK

Accepts the changes you've made and closes this dialog.

Cancel

Does not retain any changes and closes the dialog.

Help

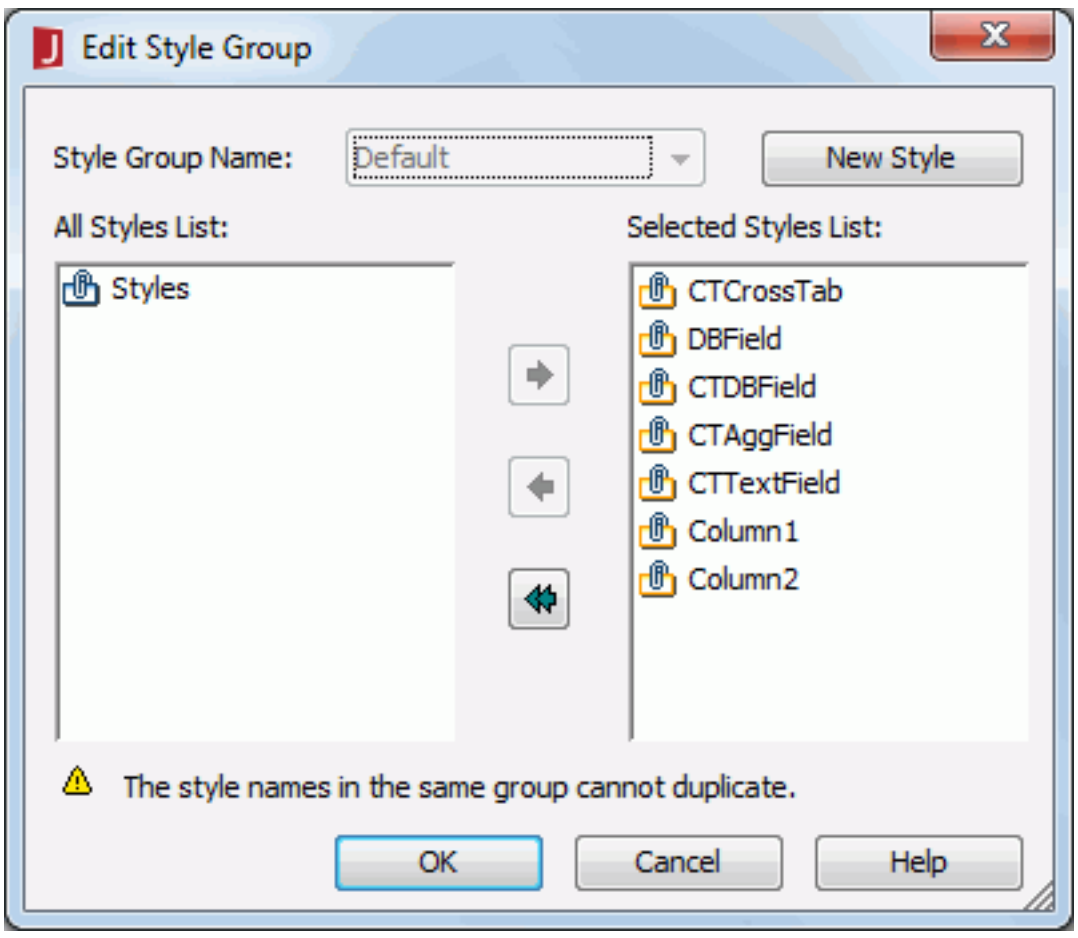
Displays the help document about this feature.

Related topics:

- Predefining sort and filter conditions

Edit Style Group dialog

This dialog appears when you right-click a style group and select Edit Group from the shortcut menu in the Styles tab of the Catalog Browser. It helps you to edit a style group.



The following are details about options in the dialog:

Style Group Name

Shows the name of the style group.

New Style

Opens the New Style dialog to create a new style.

- **Style Name**
  Specifies the name of the new style.
- **Style Type**
  Specifies the type of the new style.
- **Description**
  Specifies the description of the new style.
- **OK**
  Creates a new style in this style group.
- **Cancel**
  Cancels creating a new style.

All Styles List

Lists all the styles that are not selected to apply to this group.

Selected Styles List

Lists all the selected styles that are selected to apply to this group.

Adds the selected style into the group.

Removes the selected style from the group.

Removes all the styles from the group.

OK

Applies the changes and closes the dialog.

Cancel

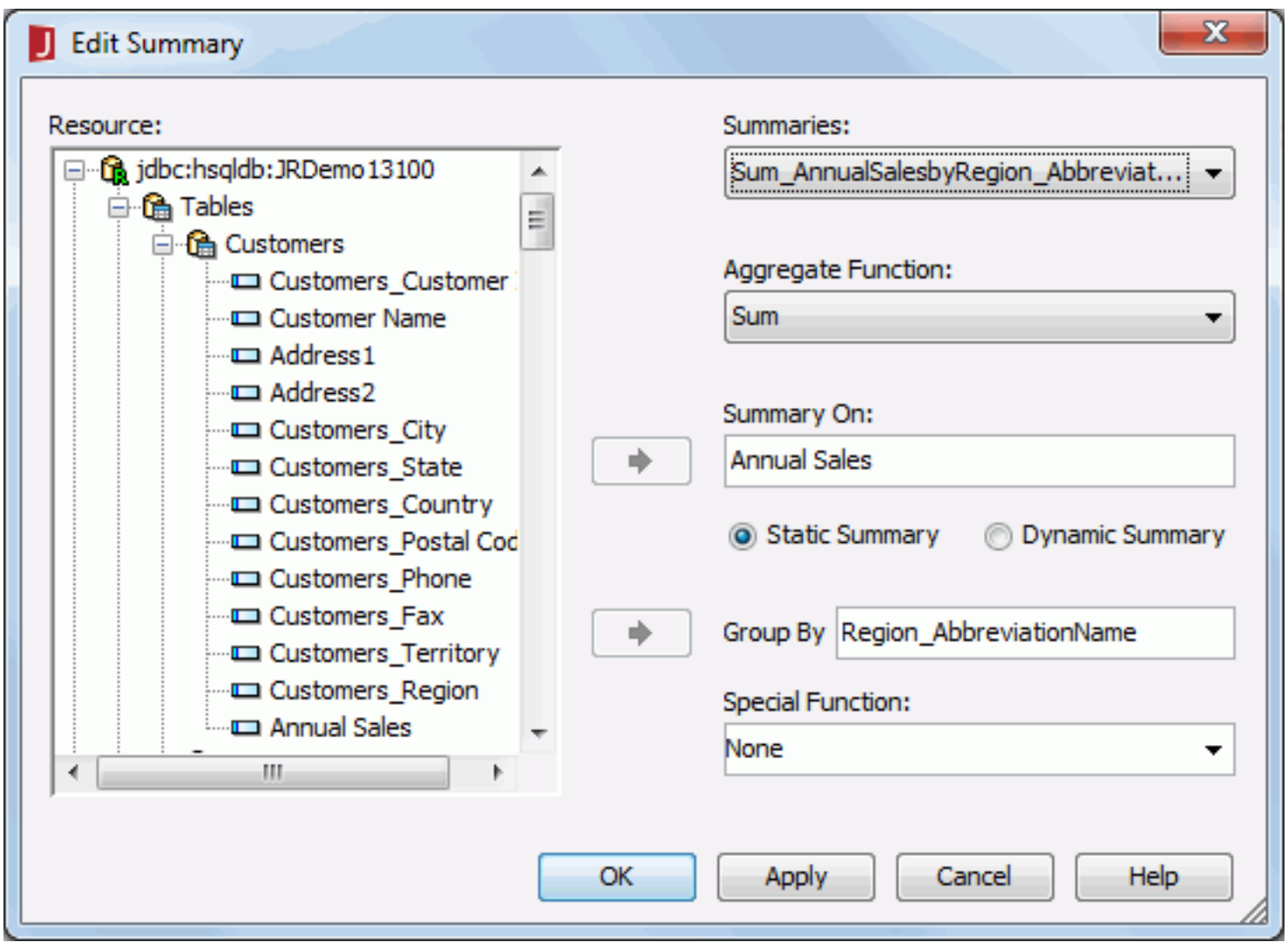
Closes the dialog, leaving any changes unsaved.

Related topics:

- Modifying an XSD style

Edit Summary dialog

This dialog appears when you right-click a summary and select Edit Summary from the shortcut menu in the Catalog Browser or the Resource View panel. It helps you to edit a summary.



The following are details about options in the dialog:

Resource

Lists the resources that can be used to make a summary.

Summaries

Displays the name of the summary that is to be edited. You can also choose to create a new summary or edit another one by selecting <Create...> or the name of the summary from the drop-down list.

Aggregate Function

Lists the functions which are available to the selected field. For details about each function, see Math functions.

Summary On

Specifies the field on which the summary is to compute.

To specify the sum on field, highlight it in the Resource box and then click next to Summary On to add it to the field box.

Static Summary

If selected, the summary will take effect on a specific group that you define in the Group By field and it can only be inserted into the group.

- **Group By**
  Specifies the field used to group the data.

  To specify the group by field, highlight it in the Resource box and then click next to Group By to add it to the field box.

Dynamic Summary

If selected, the summary will take effect on the group according to the number, which is specified in the Group By field and it can be inserted into any banded panels of a component.

- **Group By**
  Specifies on which group the summary will take effect.

  To specify the group level, first select Up or Down from the drop-down list, then in the combo box, select a value or input an integer, which should be between 0 to 127, to specify the group on which the summary will take effect. For example, if you select Up or Down and input 0 in the combo box and insert this summary into any group, the summary will take effect on this group; if you select Up and input 1 and insert this summary to a group, the summary will take effect on the group higher than the group where the summary is inserted, and if you specify Down and 1, it will take effect on the lower group; and the rest may be deduced by analogy.

  Notes:

  - If the dynamic summary is to be used in a chart, when defining the summary, you must choose Down from the Group By drop-down list; if it is to be inserted to other data containers, the summary must be defined on Up level.
  - When you insert a dynamic summary into a report and there is no corresponding group level to match settings of the dynamic summary, an error message will be generated.
  - If you want to insert a summary with a special function into a table or banded object, there must be a group with the same special function in the table or banded object.

Special Function

If the group by field is of Numeric/String/Date/Time type, you can select a special function for the field in the Special Function drop-down list to further specify to which level the data will be grouped by. If Customize is selected, the Customized Function dialog will be displayed, in which you can set the function by your own.

OK

Accepts all changes and closes the dialog.

Apply

Applies all changes to the existing summary or adds the new summary to the catalog.

Cancel

Does not retain any changes and closes this dialog.

Help

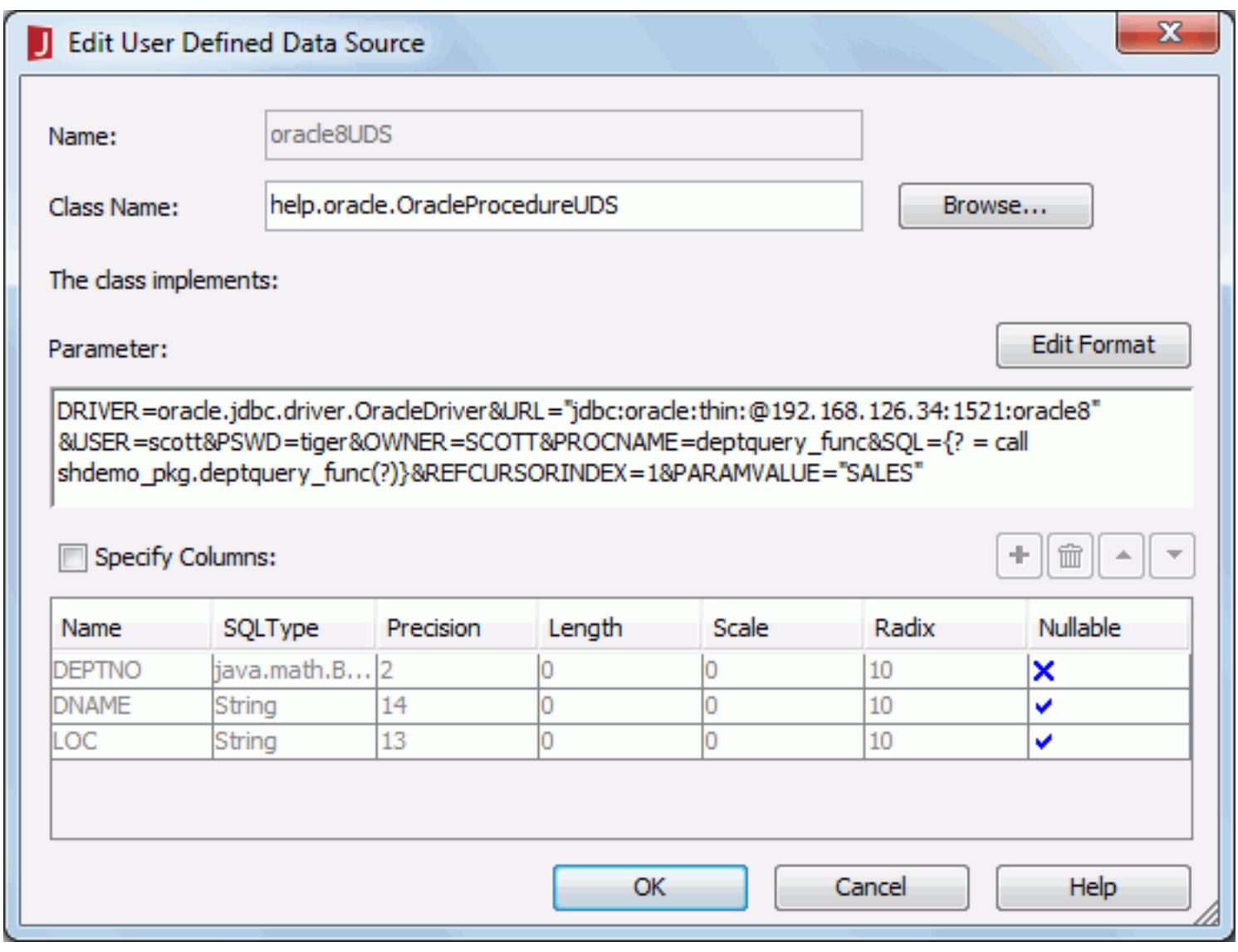
Displays the help document about this feature.

Related topics:

- Creating a summary

Edit User Defined Data Source dialog

This dialog appears when you right-click a user data source and select Edit User Defined Data Source from the shortcut menu in the Catalog Browser. It helps you to edit a UDS that you have added to a catalog.



The following are details about options in the dialog:

Name

Specifies the name of the UDS. The name cannot be the same as any of the existing procedures, SQL files, HDSs, business views or queries. It also cannot be empty.

Class Name

Specifies the full name (including package name) of the class represented by the UDS. The class should be appended to the class path in the system environment.

The class implements

After filling in the Class Name field, the class name of the interface that the class implements will be displayed here.

Parameter

Specifies the parameter string for the UDS. The parameter string must match the format defined in the UDS class. You can use parameters predefined in a JReport catalog in the parameter string, and the format used in the string is "@" + parameter name. For example, if the parameter string of a UDS is SQL=select * from employee, and you want to use the parameter sql in a catalog to replace the part after "=" in the string, then the parameter string of the UDS will be SQL=@sql.

Edit Format

Opens the Edit Parameter Format dialog to edit the parameter format:

- **Name**
  Displays names of parameters used in the user data source.
- **Format**
  Specifies formats for parameters as required.
- **OK**
  Accepts changes and closes the dialog.
- **Cancel**
  Does not retain changes and closes the dialog.

Specify Columns

Specifies the column definitions. If you don't specify the column definitions, JReport will retrieve them from result set.

- **Name**
  Specifies the name of a column. The name of the column should have the same validation with a common table column. The default names for column definitions are ‘column1', ‘column2', and so on.
- **SQLType**
  Specifies the data type of the column. The following types are supported:
  - **String**
    The corresponding JDBC type is VARCHAR or LONGVARCHAR.
  - **java.math.BigDecimal**
    The corresponding JDBC type is NUMERIC.
  - **boolean**
    The corresponding JDBC type is BIT.
  - **byte**
    The corresponding JDBC type is TINYINT.
  - **short**
    The corresponding JDBC type is SMALLINT.
  - **int**
    The corresponding JDBC type is INTEGER.
  - **long**
    The corresponding JDBC type is BIGINT.
  - **float**
    The corresponding JDBC type is REAL.
  - **double**
    The corresponding JDBC type is DOUBLE.
  - **java.io.InputStream**
    The corresponding JDBC type is VARBINARY or LONGVARBINARY.
  - **java.sql.Date**
    The corresponding JDBC type is DATE.
  - **java.sql.Time**
    The corresponding JDBC type is TIME.
  - **java.sql.Timestamp**
    The corresponding JDBC type is TIMESTAMP.
- **Precision, Length, Scale, Radix**
  The default value for each SQL type will be displayed. Click the cell to modify the value if necessary.
- **Nullable**
  Specifies whether the value of the column can be null. X stands for NoNulls, •stands for Nullable and ? stands for Nullable Unknown.

Adds a default row in the column definitions. If there is no row selected, an empty row will be added to the last line.

Removes the selected row. .

Moves the selected column one step up.

Moves the selected column one step down.

OK

Adds the UDS into a JReport catalog.

Cancel

Does not retain any changes and closes this dialog.

Help

Displays the help document about this feature.

Related topics:

- Adding a UDS to a catalog

Edit User dialog

The dialog appears when you select a user and then click in one of the following dialogs or tabs:

- Security dialog
- Edit Business View Security dialog
- Category Property dialog > Security tab
- Add View Element dialog > Security tab
- Edit View Element dialog > Security tab

It helps you to edit the user, and consists of the following tabs:

- General tab
- Relation tab

OK

Accepts all changes and closes the dialog.

Cancel

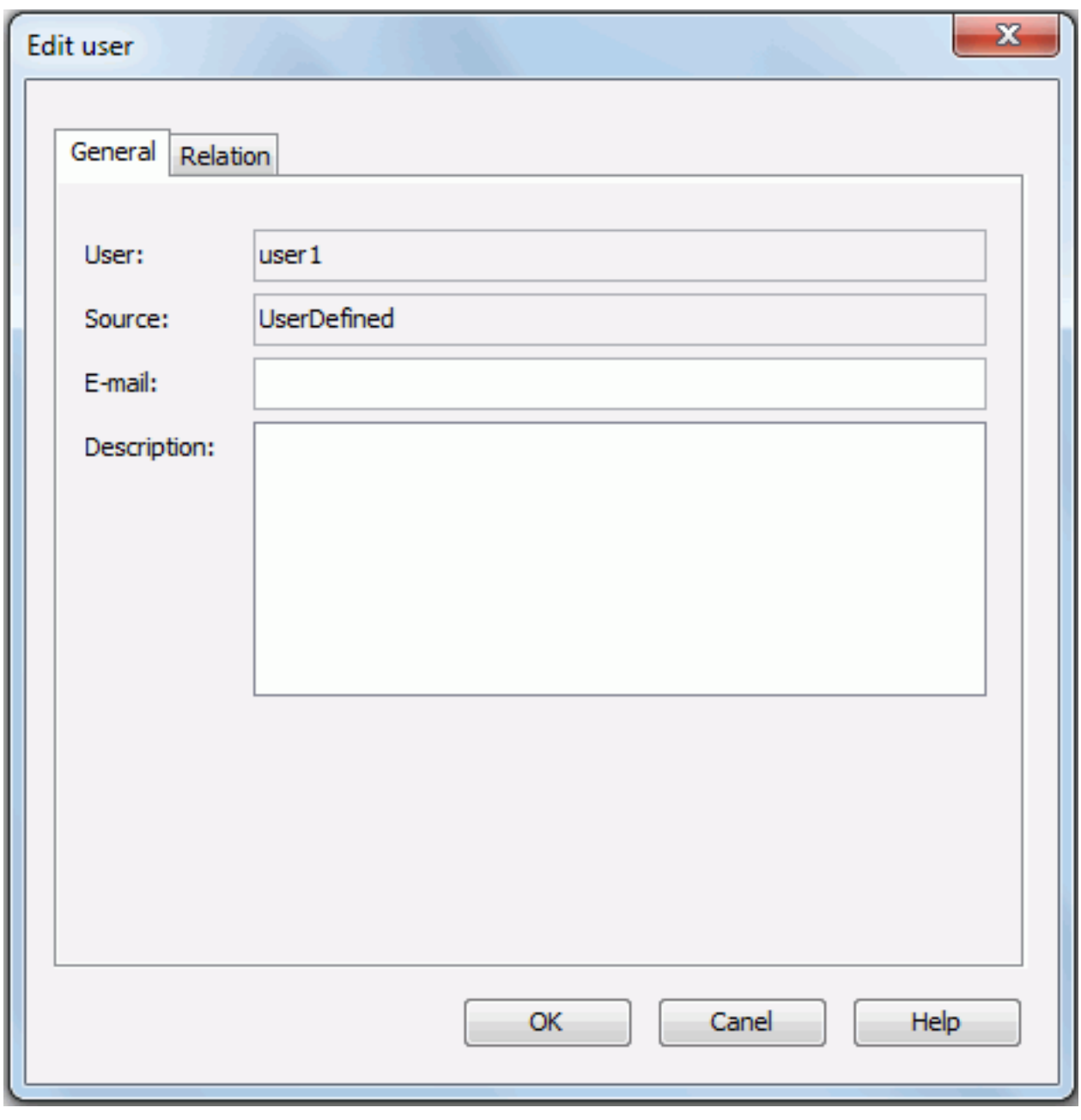
Does not retain any changes and closes the dialog.

Help

Displays the help document about this feature.

General tab

Specifies the user information.



User

Specifies the name of the user.

Source

Displays the source of the user. It is read only.

E-mail

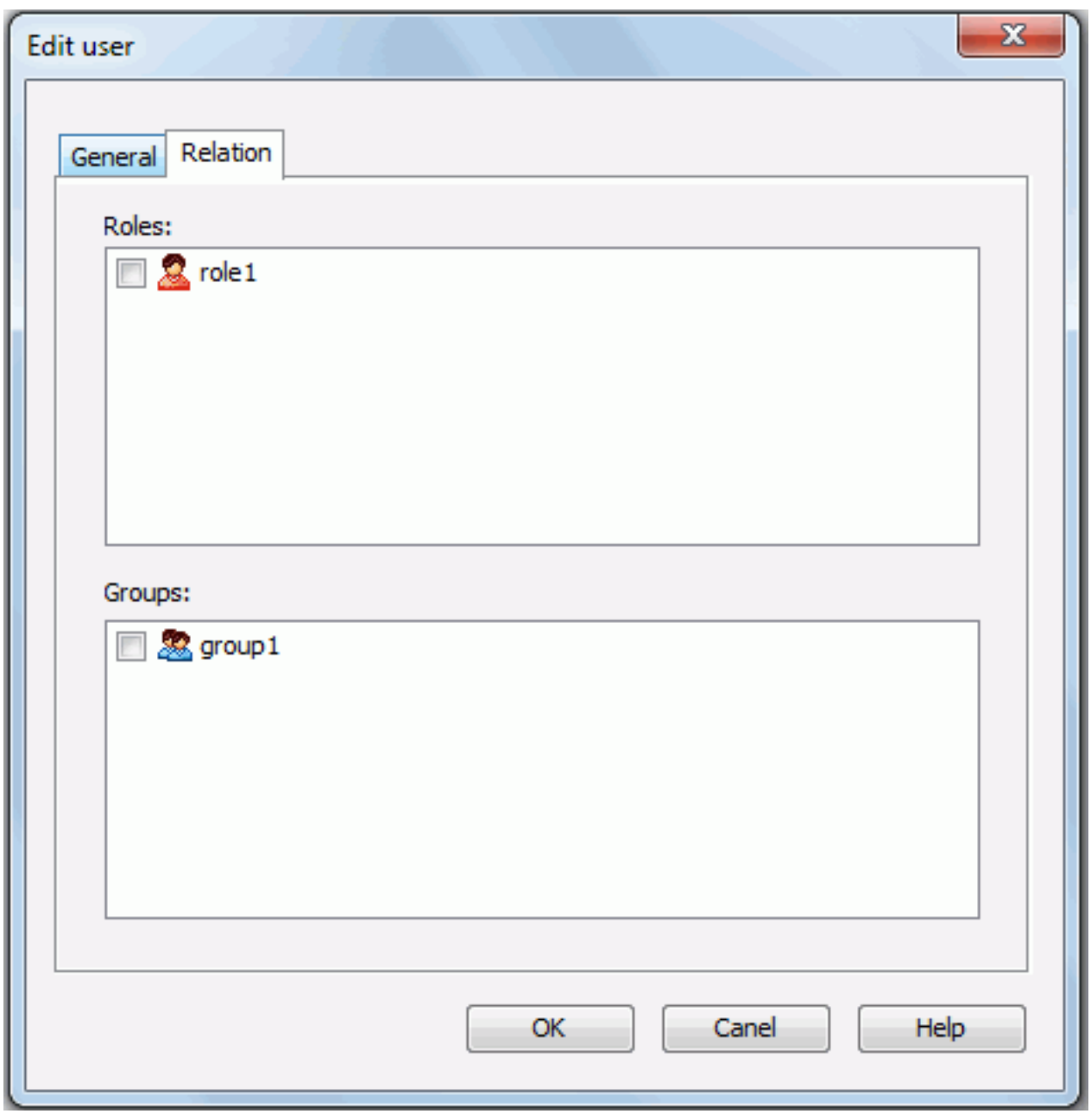
Specifies the e-mail address of the user.

Description

Specifies the description of the user.

Relation tab

Specifies the relationship of the user and the existing roles and groups.



Roles

All available roles will be listed here. If a role is checked, the user will belong to the role.

Groups

All available groups will be listed here. If a group is checked, the user will belong to the group.

Related topics:

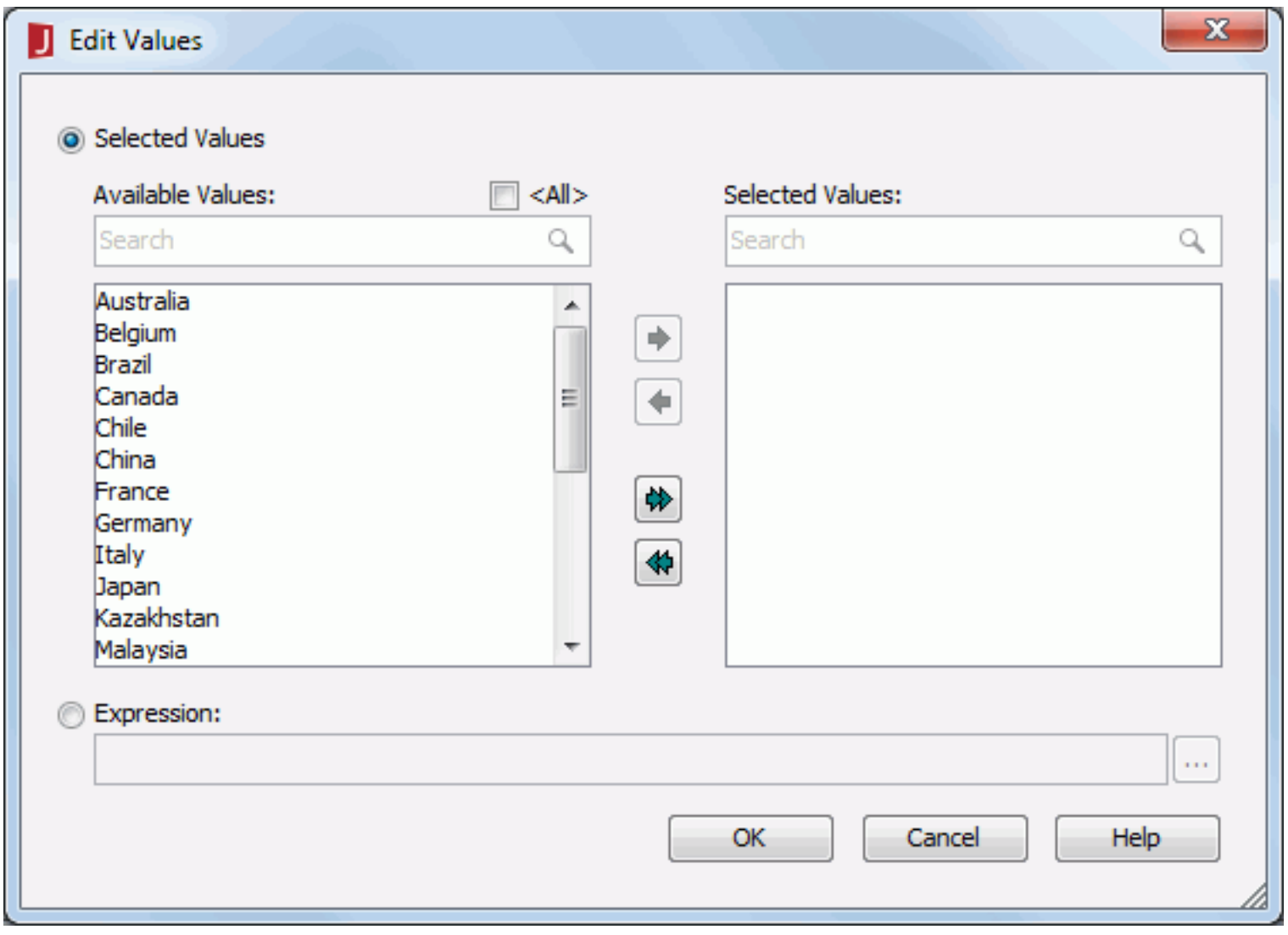
- Setting up a connection-scope security policy
- Configuring member-level security

Edit Values dialog

The dialog appears when you select a user/role/group then select a group field in the Resources panel and then click in the Member Level Security box, in one of the following dialogs or tabs:

- Edit Business View Security dialog
- Category Property dialog > Security tab
- Add View Element dialog > Security tab
- Edit View Element dialog > Security tab

It allows you to edit the field values for the allowed set or the denied set.



There are two ways of specifying the field values: select values from the available ones or compose an expression to retrieve the values. Only one can be used.

- **Selected Values**
  In this way, you can select values from the available ones.
  - **Available Values**
    Lists all the available values of the field.
  - **<All>**
    If checked, it means all possible values, not restricted to all the current displayed values. If unchecked, you can select the specific values.
  - **Quick search toolbar**
    Type in text in the toolbar to search for the values in the list below. The values containing the matched text will be shown.
  - Adds the selected value from the Available Values box to the Selected Values box.
  - Removes the selected value from the Selected Values box.
  - Adds all the values from the Available Values box to the Selected Values box.
  - Removes all the values from the Selected Values box.
  - **Selected Values**
    Lists the values that are selected.
- **Expression**
  In this way, you can compose an expression to retrieve the values.
  - Opens the Edit Conditions dialog for composing an expression to retrieve the values.

OK

Accepts the changes and closes the dialog.

Cancel

Cancels the changes and closes the dialog.

Help

Displays the help document about this feature.

Related topics:

- Configuring member-level security

Edit View Element dialog

This dialog appears when you right-click any view element of a business view and select Edit from the shortcut menu in the Catalog Browser. It helps you to edit the specified view element, and consists of the following tabs:

- General tab
- Security tab

OK

Applies all changes and closes the dialog.

Cancel

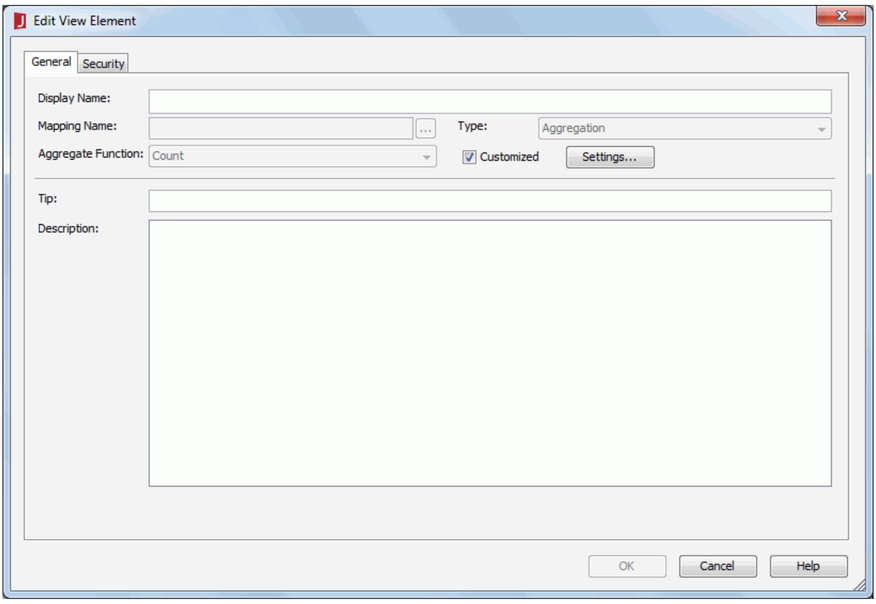
Does not retain changes and closes the dialog.

Help

Displays the help document about this feature.

General tab

Sets properties for the view element.



Display Name

Specifies the display name of the view element which is intuitive to the end user.

Mapping Name

Specifies the mapping name of the field to which the view element is mapped. Click ... to select the field.

Type

Specifies the type of the view element.

- **Group**
  Group objects are the view elements that will become the basis for analyzing data in a report. They characteristically return text data or dates. For example they may return items like country, state and year to use for grouping data in reports.
- **Aggregation**
  Aggregation objects are numeric view elements that are calculated at the time the report runs. The values they return are dependent on the group objects which they are used with. For example you may want a sales total by year.
- **Detail**
  Detail objects provide additional information. They can be bound to group, aggregation, or category elements. Detail objects are not used for analysis but are useful in tables to show detail information such as addresses and phone numbers.

Aggregate Function

Available only when the view element is an aggregation. It specifies the function for the aggregation object. For details about each function, see Math functions.

Customized

Available only when the type of the view element is defined as Aggregation. If checked, you can create a custom aggregation by writing a formula.

- **Settings**
  Opens the Formula Editor dialog to compose a formula using resources in the current business view, which can return an aggregation.

Tip

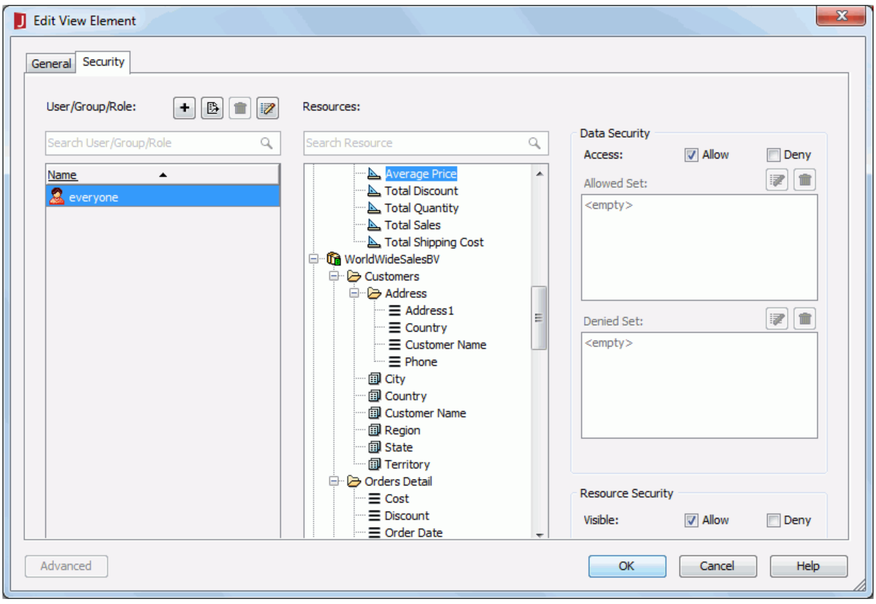
Specifies the tooltip of the view element which will be shown when you hover the mouse pointer over the element in the Resources panel of Web Report Studio or the Resource View panel of Page Report Studio.

Description

Specifies the description of the view element.

Security tab

Specifies user accessibility to members of the group objects in all the business views in current data source.



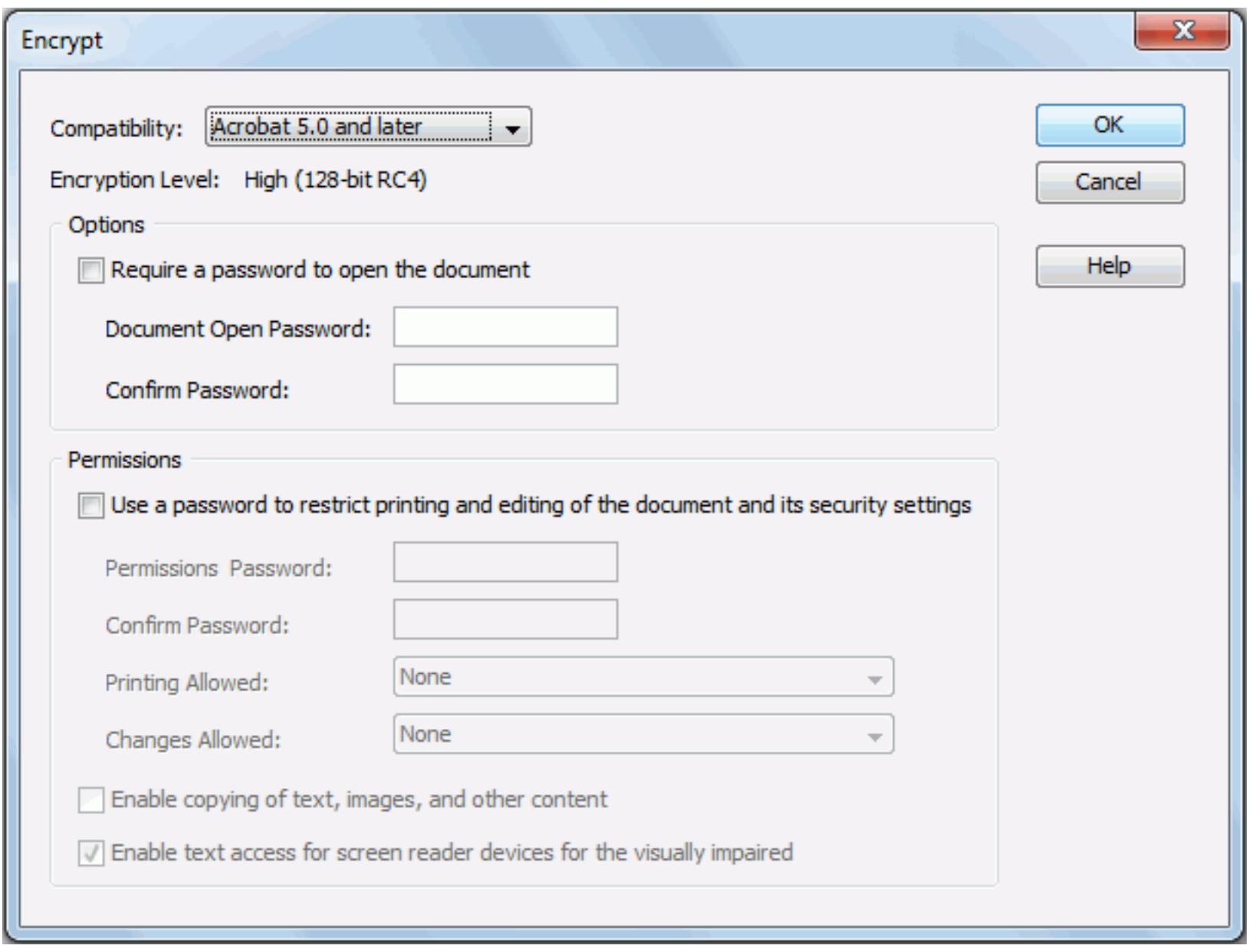
For details about options in the tab, refer to Edit Business View Security dialog.

Related topics:

- Business view elements
- Adding/Editing elements

Encrypt dialog

This dialog appears when you click the Setting button after checking the Encrypt checkbox in the Export to PDF dialog. It helps you to configure the encryption settings for the PDF file.



The following are details about options in the dialog:

Compatibility

Specifies the encryption type to encrypt the PDF document. The option Acrobat 3.0 and later uses a low encryption level (40-bit RC4), while the other option Acrobat 5.0 and later uses a high encryption level (128-bit RC4).

Encryption level

Shows the level of the encryption compatibility that you specified in the Compatibility drop-down list.

Options

Specifies options for the encryption you defined for the PDF file.

- **Require a password to open the document**
  Specifies whether the document is password protected. If checked, set the password here.
  - **Document Open Password**
    Specifies the password to prevent users from opening the document without authorization.
  - **Confirm Password**
    Confirms the password you specified in the Document Open Password text box.

Permissions

Sets permissions for the encryption that you specified for the PDF file.

- **Use a password to restrict printing and editing of the document and its security settings**
  Specifies whether to use a permission password to restrict users from printing and editing.
  - **Permission Password**
    Specifies the password to restrict users from printing and editing.

    Note: The password here cannot be the same as the one used for opening the document.
- **Confirm Password**
  Confirms the password you have specified in the Permission Password text box.
- **Printing Allowed**
  Specifies the printing quality for the PDF document.
  - **None**
    Not allowed prevents users from printing the document.
  - **Low Resolution**
    Available only when a high encryption level - Acrobat 5.0 is selected. If checked, users can print the document at no higher than 150-dpi resolution. Printing may be slower because each page is printed as a bitmapped image.
  - **High Resolution**
    Lets users print at any resolution, directing high-quality vector output to PostScript and other printers that support advanced high-quality printing features.
- **Changes Allowed**
  Specifies what editing actions are allowed in the PDF document.
  - **None**
    Prevents the user from making any changes to the document, including filling in signature and form fields.
  - **Inserting, deleting, and rotating pages**
    Available only when a high encryption level - Acrobat 5.0 is selected. If checked, users can insert, delete, and rotate pages, as well as create bookmarks and thumbnail pages.
  - **Filling in form fields and signing**
    Lets users fill in forms and add digital signatures. This option doesn't allow users to add comments or create form fields.
  - **Commenting, filling in form fields and signing**
    Lets users fill in forms and add digital signatures and comments.
  - **Any except extracting pages**
    Lets users change the document using any method listed in the Changes Allowed menu, except removing pages.
- **Enable copying of text, images and other content**
  Allows users to select and copy the contents of the PDF document.
- **Enable text access for screen reader devices for the visually impaired**
  Available only when the Compatibility option is set to Acrobat 5.0 or later Acrobat version. Checking it will enable the use of screen readers for visually impaired users.

OK

Applies all changes and closes the dialog.

Cancel

Cancels the changes and exits the dialog.

Help

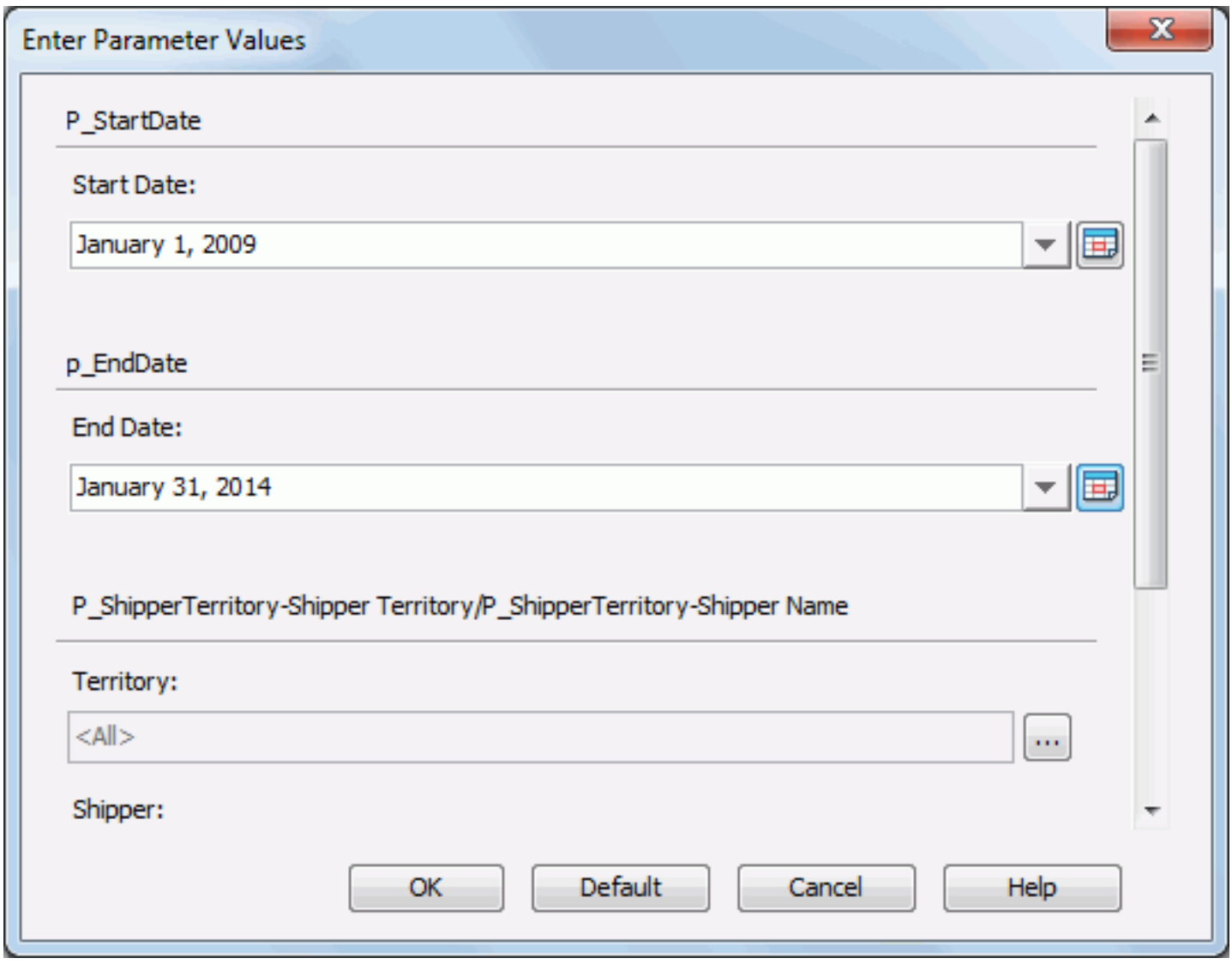
Displays the help document about this feature.

Related topics:

- Exporting report results - To PDF

Enter Parameter Values dialog

This dialog appears when you preview, print or export a report with one or more parameters. In this dialog, all the parameter names used in the report and prompting information are listed.



The way to specify a parameter value varies with the type and properties of the parameter. Here are several ways you can use to specify parameter values:

- In the parameter value combo box, select the required one from the drop-down list or input the value manually.
- Select or unselect the checkbox to specify a Yes/No value.
- Click the button ... to specify multiple values in the Enter Values dialog.
- Click the calendar button to specify a date and time value using either calendar or expression in the Calendar dialog. If you use an expression to specify the value, when you hover the mouse pointer over this value, a tip will appear showing its expression. Click on the value, the expression will be displayed in the value text box and you can edit the expression in the text box directly if you want. After editing, when you click elsewhere outside of the value text box:
  - If the edited expression is correct, a new value calculated by the expression will be displayed in the parameter value text box.
  - If the edited expression is wrong, no value will be created and the expression itself will be displayed and highlighted in red in the value text box. Correct the expression, or click the calendar button to specify a value.

OK

Applies your input and runs the report.

Default

Accepts the default value and runs the report.

Cancel

Cancels your operation and exits running the report.

Help

Displays the help document about this feature.

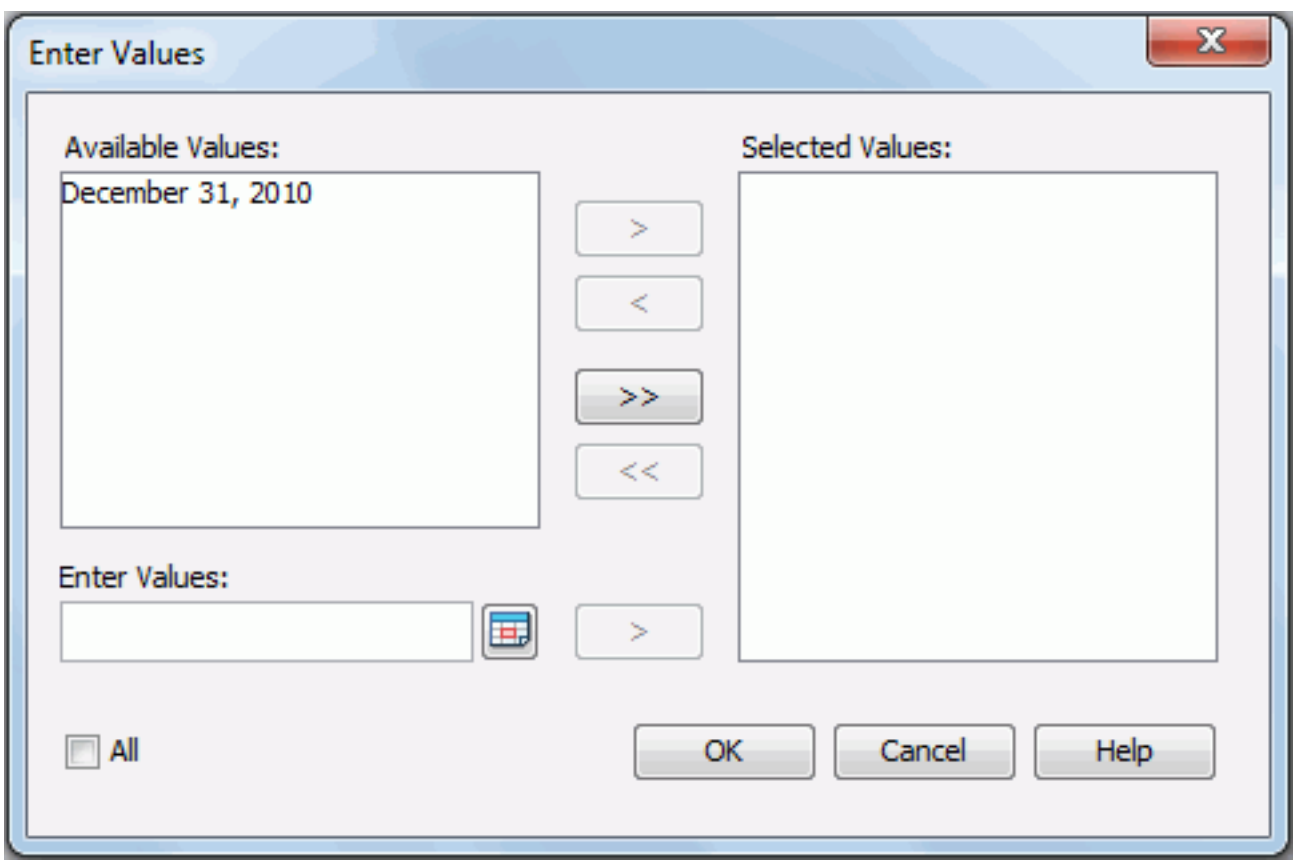
Note: You are recommended not to use blank as the thousands separator in Number-typed parameter values under French locale, otherwise your input will not be correctly recognized because of a JVM bug. For details, see http://bugs.sun.com/view_bug.do;jsessionid=c8cdaf911b20ffffffffd9fc6340b30d670?bug_id=4510618.

Related topics:

- Using built-in functions to set date and time parameter values

Enter Values dialog

The dialog appears when you click the button ... in the Enter Parameter Values dialog (the button is available when you set the parameter's Allow Multiple Values to true while creating it). The dialog helps you to specify multiple values for the parameter.



Available Values

Lists all predefined parameter values for selection. When the parameter is bound with a column, but the Display Column is different from the Bind Column, values of the Display Column are listed here.

Adds the selected values from the Available Values box to the Selected Values box.

Removes the selected values from the Selected Values box.

Adds all the selected values from the Available Values box to the Selected Values box.

Removes all the selected values from the Selected Values box.

Selected Values

Lists the values selected. The selected values are case sensitive.

Search

This option is available when you set the parameter's Allow Type-in of Value to false while creating it. It is used to search for values among the available values. After you type text in the Search text box, the matched text in the available values will be highlighted.

Enter Values

This option is available when you set the parameter's Allow Type-in of Value to true while creating it.

Enter a value manually in the text box and then click the button next to add the value to the Selected Values box. When the parameter is bound with a column, but the display column is different from the bind column, make sure the value you enter is that of the bind column.

Opens the Calendar dialog to specify a date and time value. Available only for Date, DateTime, or Time parameter which allows for type-in values and is not bound with any column.

All

If selected, it means that the parameter value result is all the values in the DBMS. This option is available when you set the parameter's Enable the "All" Option to true while creating it. This is translated in SQL to remove the parameter which may select more values than listed in the available values list.

For example, when the parameter query is:

```
SELECT CUSTOMERS.CUSTOMERID,CUSTOMERS.CUSTOMERNAME FROM CUSTOMERS where CUSTOMERS.
CUSTOMERID>0 and CUSTOMERS.CUSTOMERID<4
```

When you click >> to add 1, 2 and 3 as the parameter values, which are all the available values, the SQL is:

```
(CUSTOMERS.CUSTOMERID IN ( 1,2,3))
```

Run the same report again and this time check the All checkbox in the Enter Values dialog, the query is then:

```
( 1 = 1)
```

In this case, you will get more customers even though available values are only 1 •3.

Note: When a multi-value parameter is inserted as a field into a report and All is selected as the value, the field will show the string "All". If the parameter allows type-in values, "All" means all possible values of the parameter data type; if the parameter does not, "All" represents all the values of the parameter that come from the DBMS.

OK

Closes the dialog and applies the selected values to the parameter.

Cancel

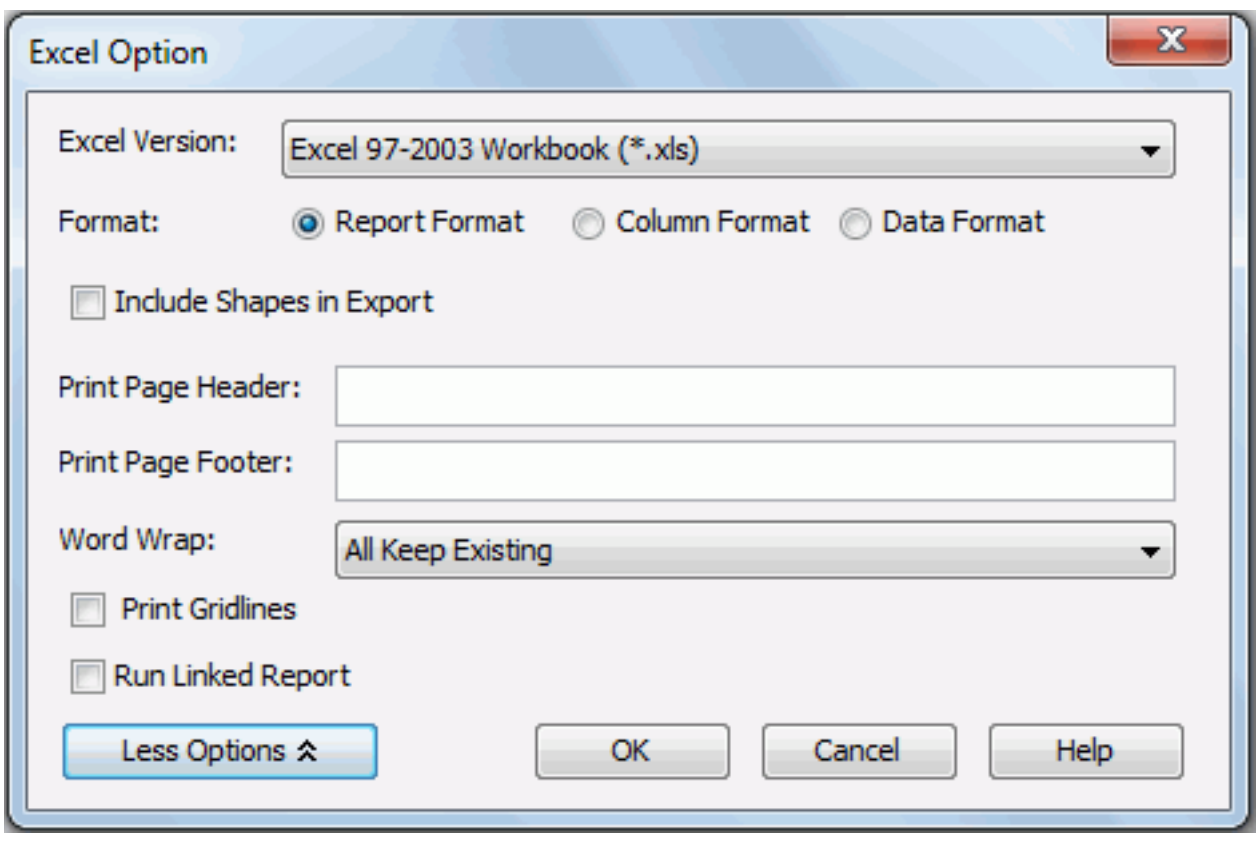
Does not retain any changes and closes the dialog.

Help

Displays the help document about this feature.

Excel Option dialog

The dialog appears when you set the Default Format for Viewing Report property of a report tab or a web report to Excel and then click ... in the value cell in the Report Inspector. It helps you to specify the parameters for running the report in Excel format.



The following are details about options in the dialog:

Excel Version

Specifies the Excel version to be used.

- **Excel 97-2003 Workbook (*.xls)**
  Exports the file as .xls format.
- **Excel Workbook (*.xlsx)**
  Exports the file as .xlsx format.

Format

Specifies the format of the Excel result.

- **Report Format**
  If checked, JReport will attempt to make the formatting of the report in Excel match the format as designed in the template. Usually this format is recommended if you just want to view the report in Excel.
- **Column Format**
  If checked, you can set the Columned property value for the report in the Report Inspector to decide the calculation method used for all components' row/column values in the Excel result.
- **Data Format**
  If checked, only the report data will be displayed without format in the Excel result. Available only for the Excel version of Excel 97-2003 Workbook (*.xls).

More/Less Options

Click to show/hide the additional settings for running the report to Excel. When Data Format is selected, only Word Wrap and Run Linked Report are available here.

- **Include Shapes in Export**
  Specifies whether to include the drawing objects in the Excel result, such as line, oval, and box.
- **Print Page Header**
  Specifies the page header text for the printed file.
- **Print Page Footer**
  Specifies the page footer text for the printed file.
- **Word Wrap**
  Specifies the word-wrap settings.
  - **All Keep Existing**
    Keeps all settings of each component's Word Wrap property as specified in the report.
  - **All Disabled**
    Disables the Word Wrap property for all components. That is, the Word Wrap property is made false for all components.
  - **All Enabled**
    Enables the Word Wrap property for all components. That is, the Word Wrap property will be made true for all components.
- **Print Gridlines**
  Specifies whether to include gridlines when printing the Excel result.
- **Run Linked Report**
  If the report is linked with other reports, you can check whether or not to generate the linked reports in the Excel result. If you are only interested in the primary report, leave this option unchecked. Generating linked reports at the same time, especially when the linked reports contain a large amount of data, will cause performance issue.

OK

Applies the changes and closes the dialog.

Cancel

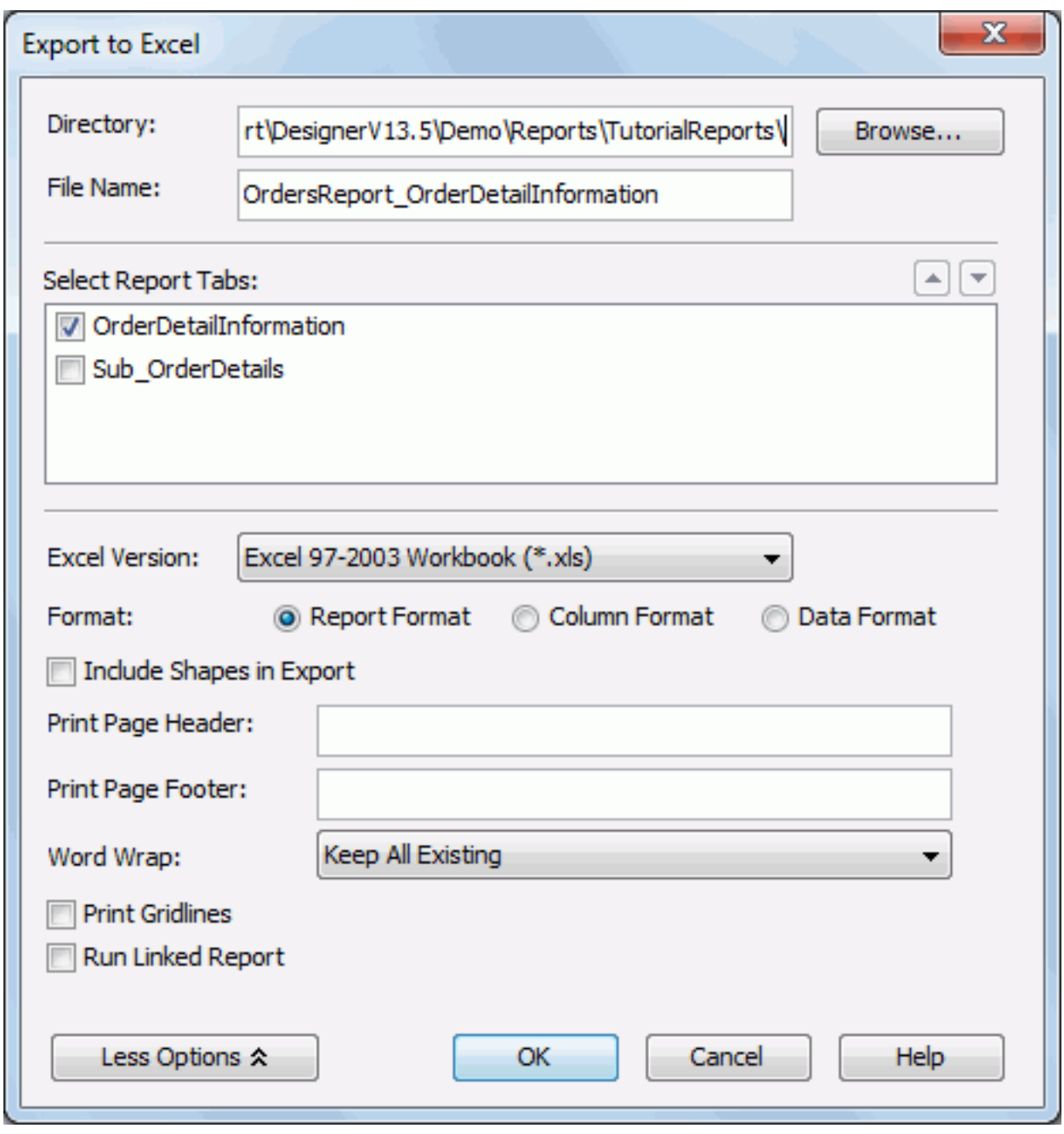
Cancels the changes and exits the dialog.

Help

Displays the help document about this feature.

Export to Excel dialog

This dialog appears when you click File > Export To > Excel on the menu bar. It helps you to export a report to Microsoft Excel format.



The following are details about options in the dialog:

Directory

Specifies the destination directory where the exported Excel file will be placed.

File Name

Specifies the name of the exported Excel file. If you do not type the name, JReport will use the report name as the Excel file name by default.

Select Report Tabs

When exporting a page report, you can specify the report tabs in the page report you want to export to the Excel file. The selected report tabs will be exported into multiple sheets of the Excel file in the list order. If the report has only one report tab, it is selected by default.

Moves the specified report tab one step up.

Moves the specified report tab one step down.

Excel Version

Specifies the Excel version to be used.

- **Excel 97-2003 Workbook (*.xls)**
  Exports the file as .xls format.
- **Excel Workbook (*.xlsx)**
  Exports the file as .xlsx format.

Format

Specifies the format of the exported Excel file.

- **Report Format**
  If checked, JReport will attempt to make the formatting of the report in Excel match the format as designed in the template. Usually this format is recommended if you just want to view the report in Excel.
- **Column Format**
  If checked, you can set the Columned property value for the report in the Report Inspector to decide the calculation method used for all components' row/column values in the report when exporting.
- **Data Format**
  If checked, only the report data will be exported without format. Available only for Excel 97-2003 Workbook (*.xls).

More / Less Options

Click to show/hide the additional settings for exporting the report to Excel. When Data Format is selected, only Word Wrap and Run Linked Report are available here.

- **Include Shapes in Export**
  Specifies whether to include the drawing objects in the exported Excel file, such as line, oval, and box.
- **Print Page Header**
  Specifies the page header text for the printed file.
- **Print Page Footer**
  Specifies the page footer text for the printed file.
- **Word Wrap**

  Specifies the word-wrap settings.
  - **All Keep Existing**
    Keeps all settings of each component's Word Wrap property as specified in the report.
  - **All Disabled**
    Disables the Word Wrap property for all components. That is, the Word Wrap property is made false for all components.
  - **All Enabled**
    Enables the Word Wrap property for all components. That is, the Word Wrap property will be made true for all components.
- **Print Gridlines**
  Specifies whether to include gridlines when printing the exported Excel file.
- **Run Linked Report**
  If the report is linked with other reports, you can check whether or not to generate the linked reports in the exported Excel file. If you are only interested in the primary report, leave this option unchecked. Generating linked reports at the same time, especially when the linked reports contain a large amount of data, will cause performance issue.

OK

Applies all changes and closes the dialog.

Cancel

Cancels the changes and exits the dialog.

Help

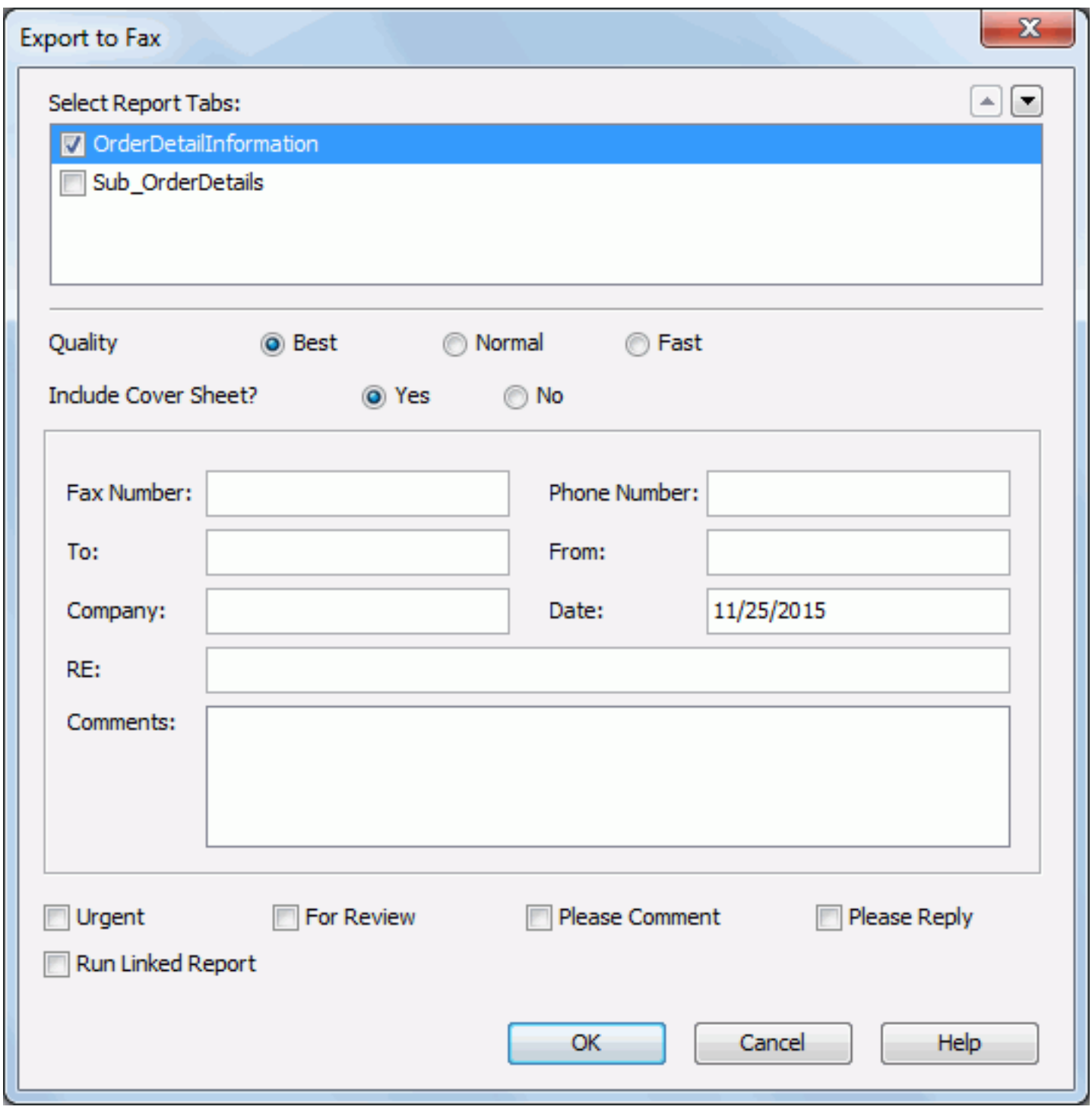
Displays the help document about this feature.

Related topics:

- Exporting report results - To Excel

Export to Fax dialog

This dialog appears when you click File > Export To > Fax on the menu bar. It helps you to export a report to fax.



The following are details about options in the dialog:

Select Report Tabs

When exporting a page report, you can specify the report tabs in the page report you want to export. The selected report tabs will be exported in the list order. If the report has only one report tab, it is selected by default.

Moves the specified report tab one step up.

Moves the specified report tab one step down.

Quality

Specifies the quality of the fax. It can be one of the following: Best, Normal, or Fast.

Include Cover Sheet

Specifies whether to send a cover sheet with the fax.

Fax Number

Specifies the fax number of the recipient.

Phone Number

Specifies the phone number of the sender.

To

Specifies information of the recipient.

From

Specifies information of the sender.

Company

Specifies information of the company.

Date

Specifies the day on which the fax is sent.

RE

Specifies the subject of the fax.

Comments

Specifies some comments to the fax.

Urgent

Specifies whether or not the fax is urgent.

For Review

Specifies whether or not only review is required for the fax.

Please Comment

Specifies whether or not comments are required for the fax.

Please Reply

Specifies whether or not a reply is required for the fax.

Run Linked Report

If the report is linked with other reports, you can check whether or not to generate the linked reports for the fax. If you are only interested in the primary report, leave this option unchecked. Generating linked reports at the same time, especially when the linked reports contain a large amount of data, will cause performance issue.

OK

Applies all changes and closes the dialog.

Cancel

Cancels the changes and exits the dialog.

Help

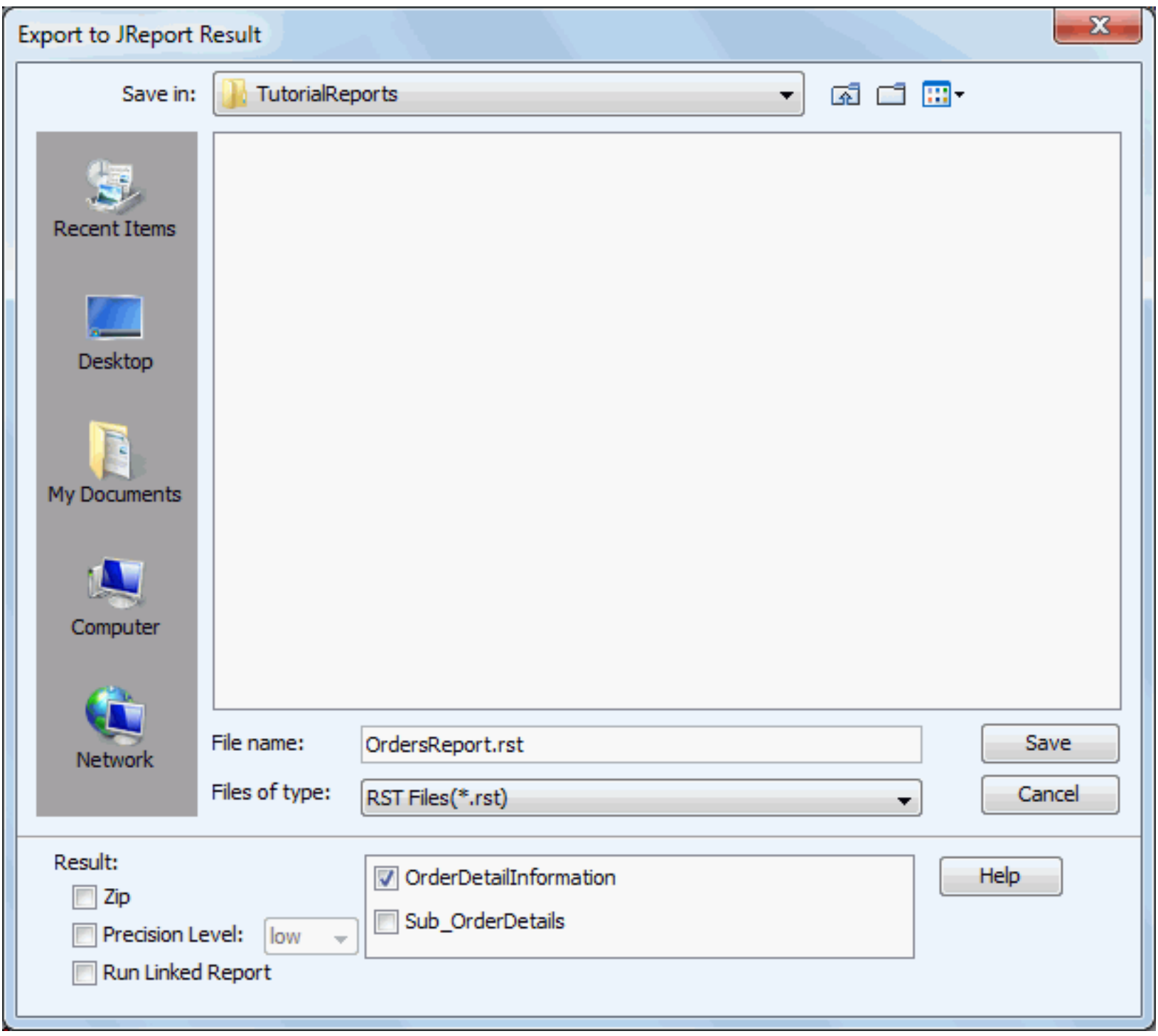
Displays the help document about this feature.

Related topics:

- Exporting report results - To fax

Export to JReport Result dialog

This dialog appears when you click File > Export To > JReport Result on the menu bar. It helps you to export a report to JReport Result format.



The following are details about options in the dialog:

Save in

Specifies the destination directory where you want to save the report result.

File Name

Specifies the name of the result file.

Files of type

Specifies the type of the result file.

Result

Lists all the report tabs in this report. You can check the checkboxes to export the corresponding report tabs.

Zip

Specifies whether or not to export the report to a zip file.

Precision Level

Specifies the precision level with which the selected report tabs will be exported.

Run Linked Report

If the report is linked with other reports, you can check whether or not to generate the linked reports in the exported JReport result file. If you are only interested in the primary report, leave this option unchecked. Generating linked reports at the same time, especially when the linked reports contain a large amount of data, will cause performance issue.

Save

Saves the report as a JReport Result format to the specific directory.

Cancel

Does not retain the changes and closes the dialog.

Help

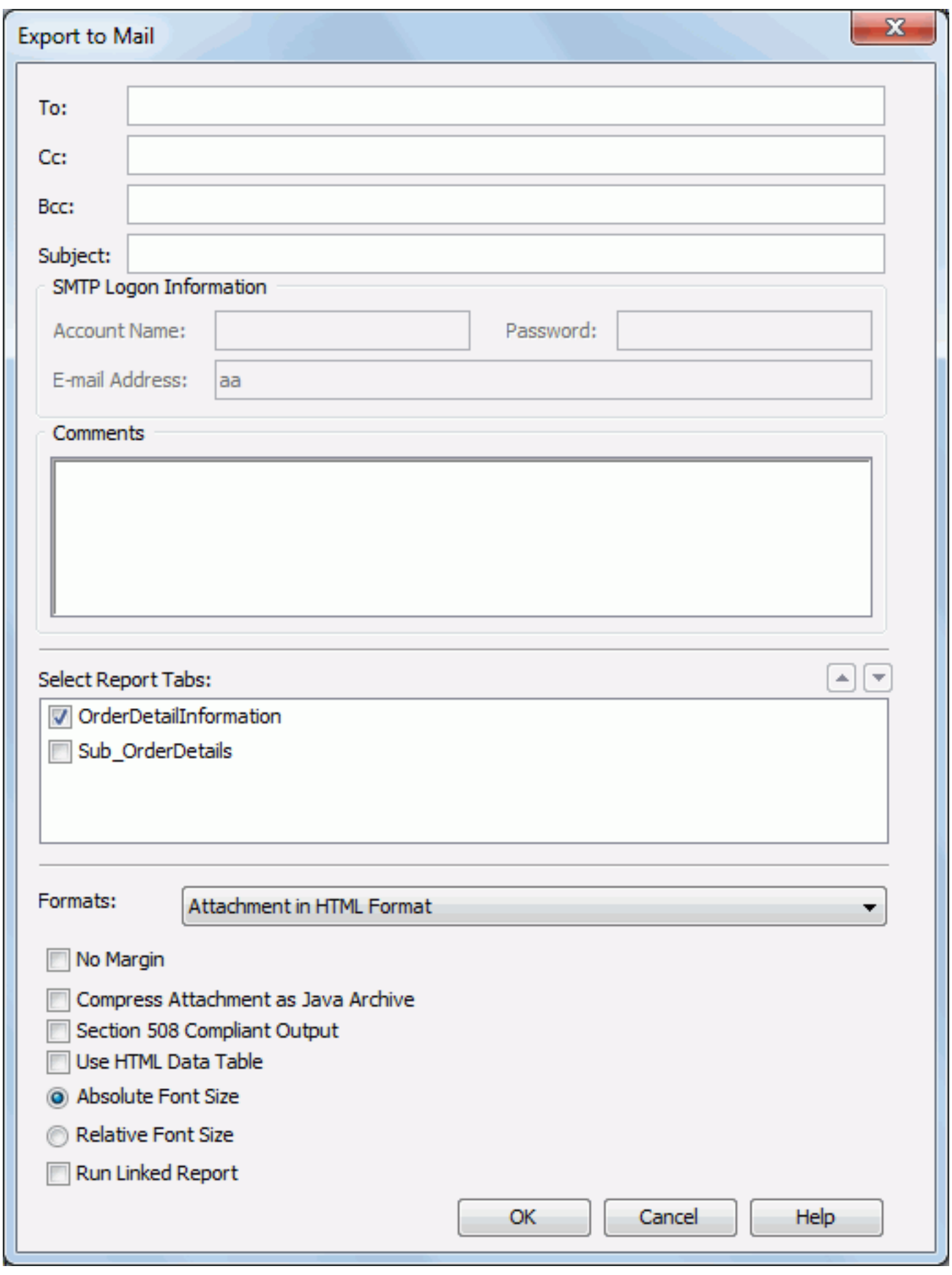
Displays the help document about this feature.

Related topics:

- Exporting report results - To JReport result

Export to Mail dialog

This dialog appears when you click File > Export To > Mail on the menu bar. It helps you to send the current report as an attachment in a specified format to someone else via e-mail.



The following are the details about the options in this dialog:

To

Specifies the e-mail address of the recipients to whom you want to send the report.

Cc

Specifies the e-mail address of the recipients to whom you want to send a copy of the report.

Bcc

Specifies the e-mail address of the recipients to whom you want to send a blind carbon copy of the report.

Subject

Specifies the subject of the e-mail that is to be sent.

SMTP Logon Information

The following options are available only when you have checked Server requires authentication in the Options dialog (**File** > **Options** > **Export to** > **E-mail** > **Server requires authentication**).

- **Account Name**
  Specifies the account name in the SMTP server.
- **Password**
  Specifies the password for logging on to the SMTP server.
- **E-mail Address**
  Specifies the e-mail address in the SMTP server.

  Note: Some SMTP may not need the mail address as one part of authentication information).

Comments

Specifies some comments about the report. The contents in this text box will be shown in the text part of the mail.

Select Report Tabs

When exporting a page report, you can specify the report tabs in the page report you want to send via e-mail. The selected report tabs will be sent in the list order. If the report has only one report tab, it is selected by default.

Moves the specified report tab one step up.

Moves the specified report tab one step down.

Formats

Specifies in which mail format the report will be sent.

- **E-mail Result in HTML E-mail Format**
  Sends the report result via e-mail to the specified address in HTML format. The report result will be shown in HTML format in the mail body. If this format is selected, the comments that you input will be overwritten by the report. For the parameter settings, refer to Export to HTML dialog.
- **E-mail Result in Plain Text E-mail Format**
  Sends the report result via e-mail to the specified address in plain text format. The report result will be shown in plain text format in the mail body with no other information such as color, images and so on. For the parameter settings, refer to Export to Text dialog.
- **Attachment in JReport Result Format**
  Sends the report result via e-mail to the specified address with a JReport result file as attachment. You can specify whether to compress the attachment as Java Archive. For the other parameter settings, refer to Export to JReport Result dialog.
- **Attachment in HTML Format**
  Sends the report result via e-mail to the specified address with an HTML file as attachment. You can specify whether to compress the attachment as Java Archive. For the other parameter settings, refer to Export to HTML dialog.
- **Attachment in PDF Format**
  Sends the report result via e-mail to the specified address with a PDF file as attachment. You can specify whether to compress the mail attachment as Java Archive. For the other parameter settings, refer to Export to PDF dialog.
- **Attachment in Excel Format**
  Sends the report result via e-mail to the specified address with an Excel file as attachment. You can specify whether to compress the attachment as Java Archive. For the other parameter settings, refer to Export to Excel dialog.
- **Attachment in Text Format**
  Sends the report result via e-mail to the specified address with a Text file as attachment. You can specify whether to compress the attachment as Java Archive. For the other parameter settings, refer to Export to Text dialog.
- **Attachment in RTF Format**
  Sends the report result via e-mail to the specified address with a RTF file as attachment. You can specify whether to compress the attachment as Java Archive. For the other parameter settings, refer to Export to RTF dialog.
- **Attachment in XML Format**
  Sends the report result via e-mail to the specified address with an XML file as attachment. You can specify whether to compress the attachment as Java Archive. For the other parameter settings, refer to Export to XML dialog.
- **Attachment in PostScript Format**
  Sends the report result via e-mail to the specified address with a PostScript file as attachment. You can specify whether to compress the attachment as Java Archive. For the other parameter settings, refer to Export to PostScript dialog.

OK

Applies all changes, and sends the report to the specified E-mail address.

Cancel

Does not retain any changes and closes the dialog.

Help

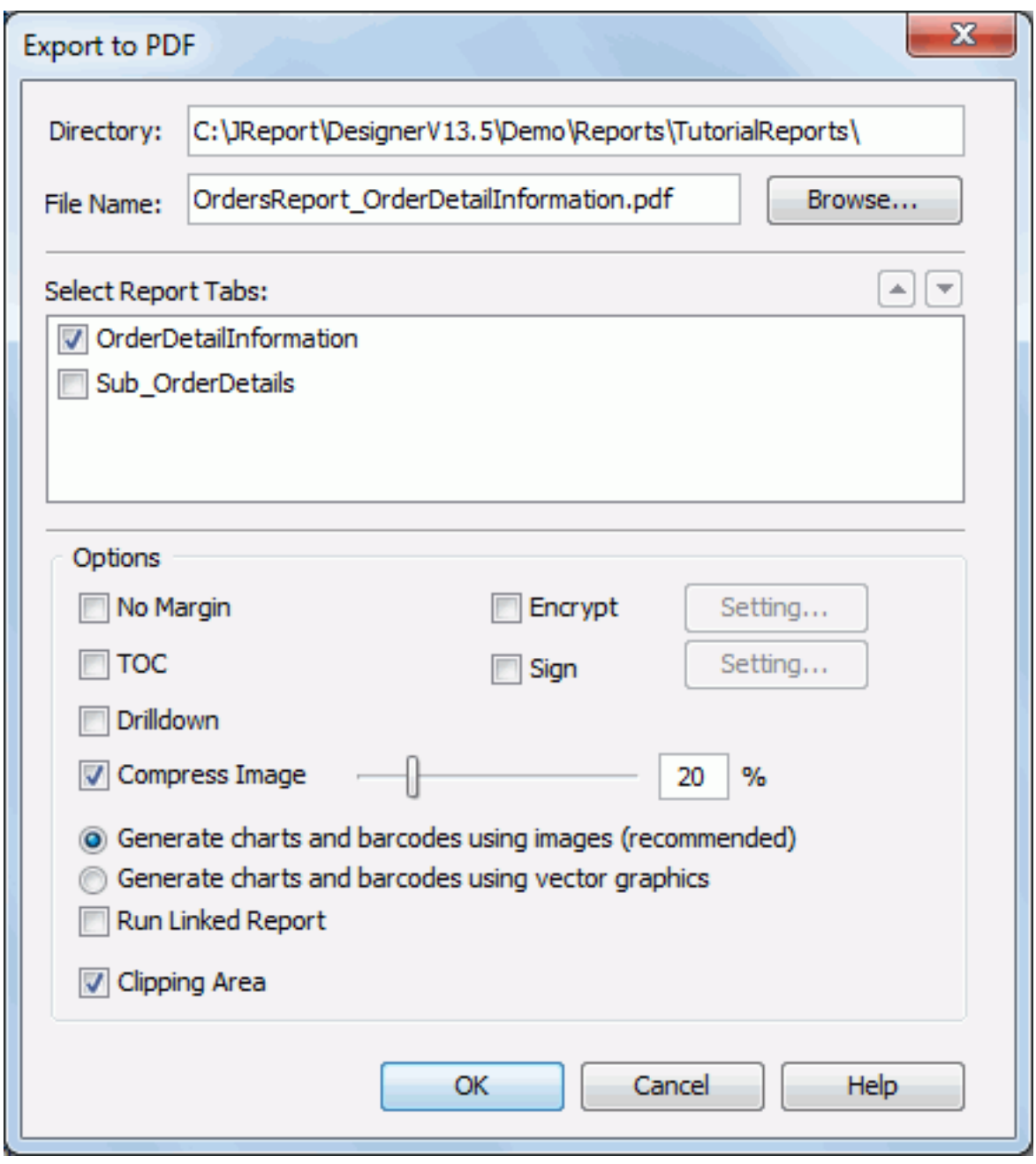
Displays the help document about this feature.

Related topics:

- Exporting report results - To mail

Export to PDF dialog

This dialog appears when you click File > Export To > PDF on the menu bar. It helps you to export a report to PDF format.



The following are details about options in the dialog:

Directory

Specifies the destination directory where the exported PDF file will be placed.

File Name

Specifies the name of the exported PDF file. If you do not type a name, JReport will use the report name as the PDF file name.

Select Report Tabs

When exporting a page report, you can specify the report tabs in the page report you want to export to the PDF file. The selected report tabs will be exported in the list order. If the report has only one report tab, it is selected by default.

Moves the specified report tab one step up.

Moves the specified report tab one step down.

No Margin

Specifies whether to remove the margins originally set when the report was designed.

TOC

Specifies whether to include a Table of Contents (TOC) in the exported PDF file.

Drilldown

Specifies whether to generate the report result to a PDF file with the Drilldown feature enabled. The Drilldown feature enables you to inspect certain items for further detailed data.

Encrypt

Specifies whether to encrypt the PDF file. Click the Setting button to configure the encrypt settings in the Encrypt dialog.

Sign

Specifies whether to add the digital sign to the PDF file. Click the Setting button to configure the sign settings in the Sign Digital ID dialog.

Compress Image

If checked, the size of the images or pictures in the report will be compressed to the specified proportion of its original size.

Generate charts and barcodes using images (recommended)

If selected, charts, UDOs and barcodes in the report will be exported to common pictures, which will look dimmed and the file size may be relatively large. But, the transparency property is supported.

Generate charts and barcodes using vector graphics

If selected, charts, UDOs and barcodes in the report will be exported to vector graphics, which will not be anamorphic when it is zoomed out or zoomed in. But, the transparency property is not supported.

Run Linked Report

If the report is linked with other reports, you can check whether or not to generate the linked reports in the exported PDF file. If you are only interested in the primary report, leave this option unchecked. Generating linked reports at the same time, especially when the linked reports contain a large amount of data, will cause performance issue.

Clipping Area

Specifies whether to keep the clipping area of the report in the exported PDF file.

OK

Applies all changes and closes the dialog.

Cancel

Cancels the changes and exits the dialog.

Help

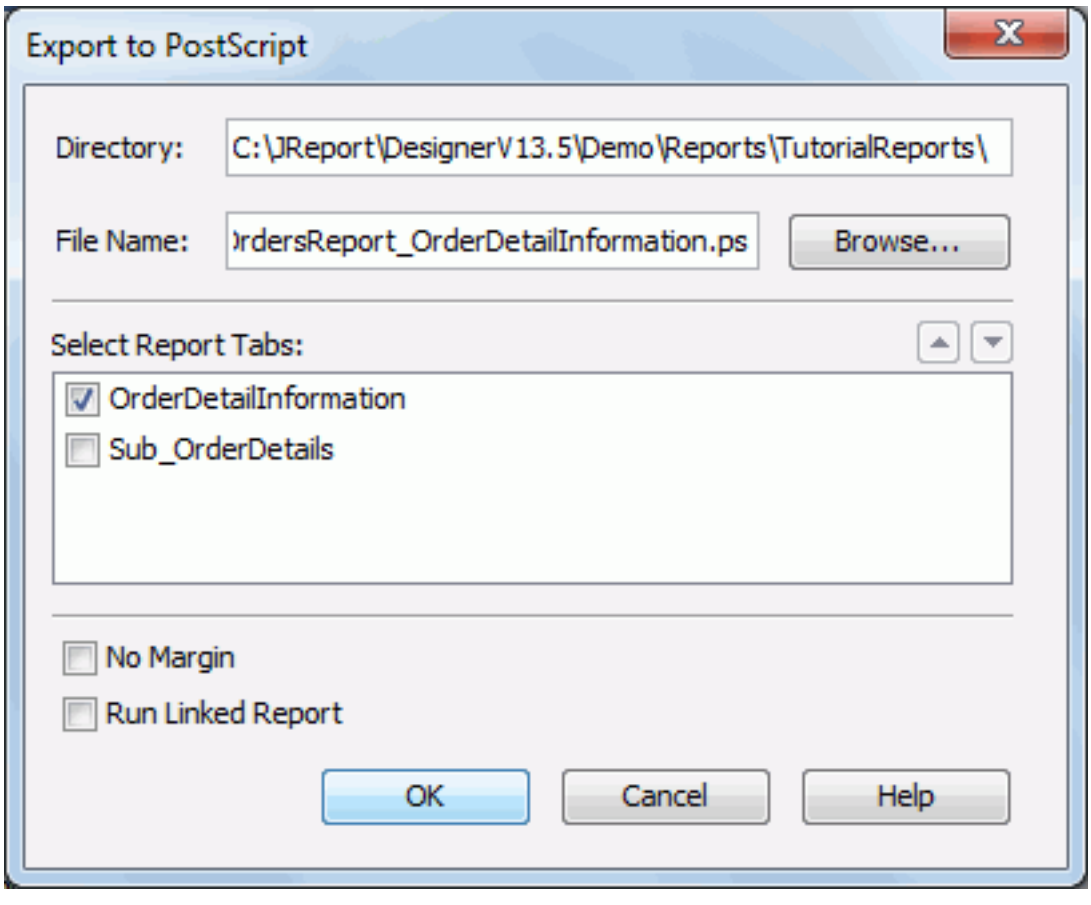
Displays the help document about this feature.

Related topics:

- Exporting report results - To PDF

Export to PostScript dialog

This dialog appears when you click File > Export To > PostScript on the menu bar. It helps you to export a report to PostScript format.



The following are details about options in the dialog:

Directory

Specifies the destination directory where you want to place the exported PostScript file.

File Name

Specifies the name of the exported PostScript file. If you do not type the name, JReport will use the report name PostScript file name as default.

Select Report Tabs

When exporting a page report, you can specify the report tabs in the page report you want to export to the PostScript file. The selected report tabs will be exported in the list order. If the report has only one report tab, it is selected by default.

Moves the specified report tab one step up.

Moves the specified report tab one step down.

No Margin

Specifies whether to remove the margins originally set when the report was designed.

Run Linked Report

If the report is linked with other reports, you can check whether or not to generate the linked reports in the exported PostScript file. If you are only interested in the primary report, leave this option unchecked. Generating linked reports at the same time, especially when the linked reports contain a large amount of data, will cause performance issue.

OK

Applies the changes and closes the dialog.

Cancel

Cancels the changes and exits the dialog.

Help

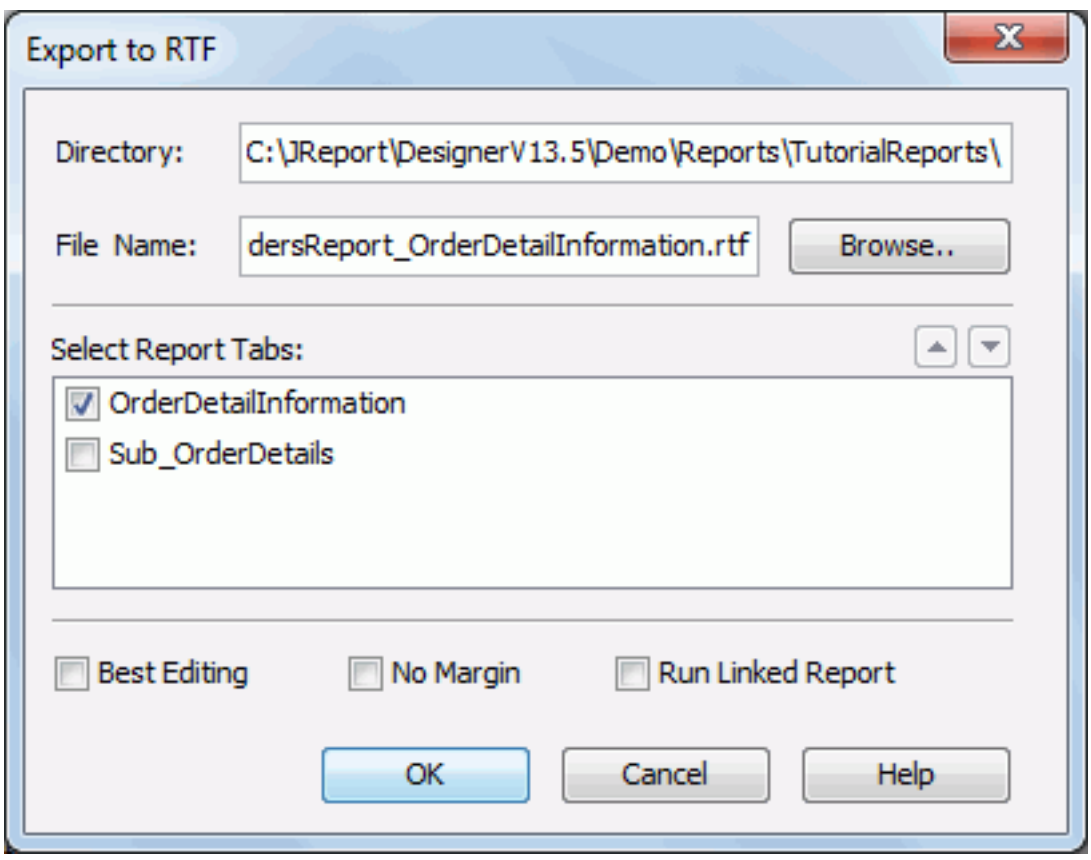
Displays the help document about this feature.

Related topics:

- Exporting report results - To PostScript

Export to RTF dialog

This dialog appears when you click File > Export To > RTF on the menu bar. It helps you to export a report to RTF format.



The following are details about options in the dialog:

Directory

Specifies the destination directory where the exported RTF file will be placed.

File Name

Specifies the name of the exported RTF file. If you do not type the name, JReport will use the report name as the default RTF file name.

Select Report Tabs

When exporting a page report, you can specify the report tabs in the page report you want to export to the RTF file. The selected report tabs will be exported in the list order. If the report has only one report tab, it is selected by default.

Moves the specified report tab one step up.

Moves the specified report tab one step down.

Best Editing

Specifies whether to apply flow layout when exporting the report to RTF.

No Margin

Specifies whether to remove the margins originally set when the report is designed.

Run Linked Report

If the report is linked with other reports, you can check whether or not to generate the linked reports in the exported RTF file. If you are only interested in the primary report, leave this option unchecked. Generating linked reports at the same time, especially when the linked reports contain a large amount of data, will cause performance issue.

OK

Applies the changes and closes the dialog.

Cancel

Cancels the changes and exits the dialog.

Help

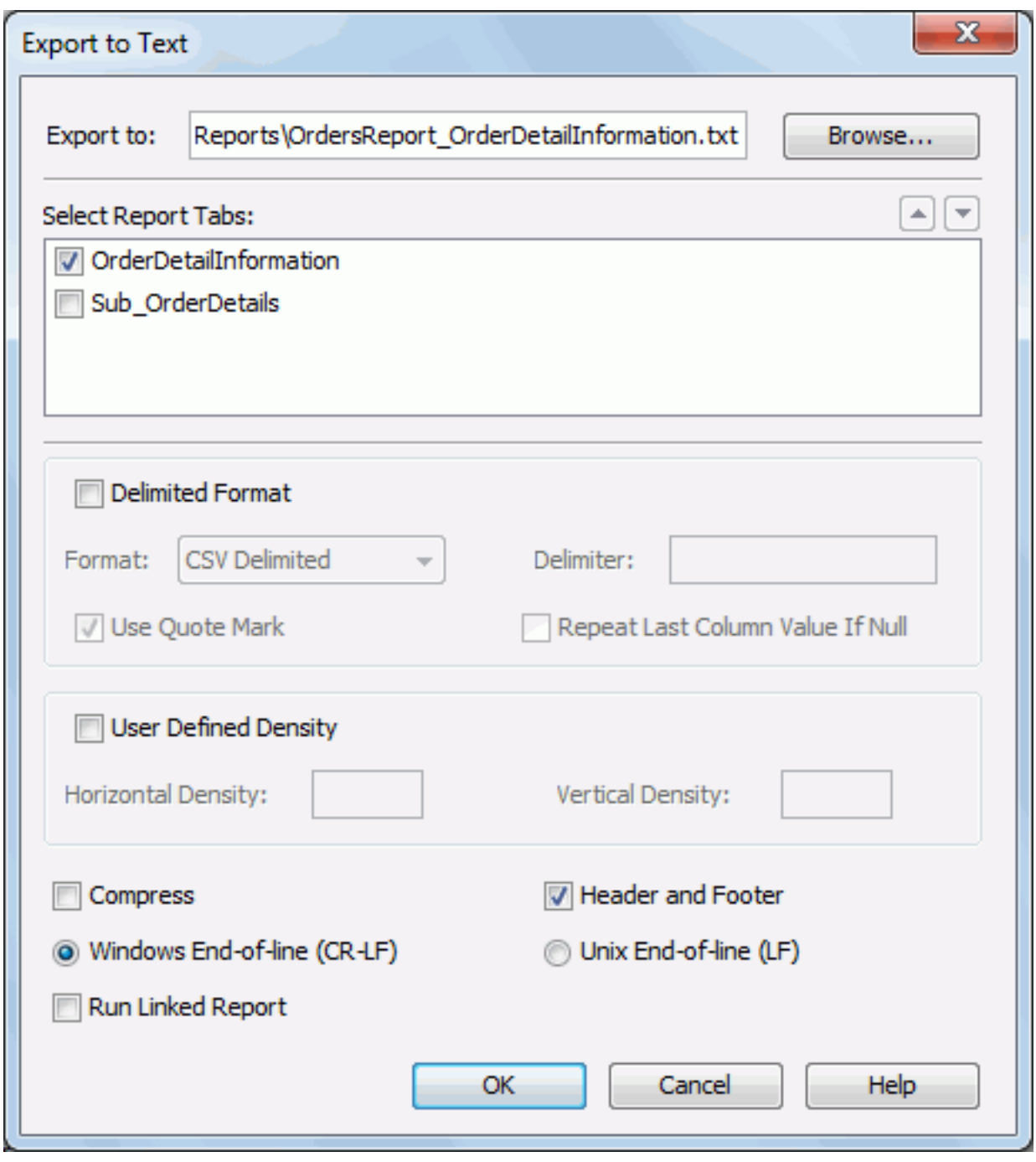
Displays the help document about this feature.

Related topics:

- Exporting report results - To RTF

Export to Text dialog

This dialog appears when you click File > Export To > Text on the menu bar. It helps you to export a report to a Text file.



The following are details about options in the dialog:

Export to

Specifies the file name and destination directory where you want to store the exported Text file.

Select Report Tabs

When exporting a page report, you can specify the report tabs in the page report you want to export to the Text file. The selected report tabs will be exported in the list order. If the report has only one report tab, it is selected by default.

Moves the specified report tab one step up.

Moves the specified report tab one step down.

Delimited Format

Specifies whether to use the delimited format to export the report result.

- **Format**
  Specifies the delimiter to separate fields.
  - **CSV Delimited**
    Fields in the exported Text file will be separated by a comma.
  - **Tab Delimited**
    Fields in the exported Text file will be separated by a tab.
  - **Customize Delimited**
    Fields in the exported Text file will be separated by a user defined delimiter. You can type your own delimiter in the Delimiter box (note that the delimiter should be only one character).
- **Use Quote mark**
  Specifies whether the fields in the exported Text file will be marked with quotation marks.
- **Repeat Last Column Value If Null**
  If checked, when a cell in the exported CSV Text has no value, value of the previous cell in the same column will be used.

User Defined Density

Specifies whether to use user defined density to export the report result.

- **Horizontal Density**
  Specifies the value for each unit of the horizontal density between columns. The resulting density is a direct ratio of the value you specify. That is, the greater the value, the smaller the width between columns.
- **Vertical Density**
  Specifies the value for each unit of the vertical density between columns. The resulting density is a direct ratio of the value you specify. That is, the greater the value, the smaller the height between columns.

Notes:

- By exporting using user defined densities, if the densities are not set appropriately, the fields in the report may overlap each other, so you are not recommended to use this way to export the report result to Text.
- When setting the value of Horizontal/Vertical Density, you need to pay attention to the following:
  - The value of Horizontal/Vertical Density must be greater than the character's width/height of the smallest field in the report (smallest field is the field with the smallest font size), otherwise, the value you set will not be applied.
  - If the value of Vertical Density is greater than 0 and the value of Horizontal Density is less than 0, the value that you specify for the Vertical Density will be applied and the value of Horizontal Density will be specified by JReport.
  - If the value of Vertical Density is less than 0 and the value of Horizontal Density is greater than 0, the value that you specify for the Horizontal Density will be applied and the value of Vertical Density will be specified by JReport.
  - If the values of Vertical Density and Horizontal Density are both greater than 0 and the value of Horizontal Density is less than 11, the specified value of the two densities will be applied. Otherwise, they will be specified by JReport.
  - If the values of Vertical Density and Horizontal Density are both less than 1, these densities will be specified by JReport.

Compress

Specifies whether to generate the report result to Text format in a compressed size, that is to say, there will be no clearance between the columns.

Header and Footer

If checked, the exported file will contain all headers and footers in the report, including Report Header/Footer, Page Header/Footer and Group Header/Footer. Otherwise, the exported file will only contain data in Detail panel.

Windows End-of-line (CR-LF)

Specifies whether to use the CR-LF in Windows convention as the end-of-line character.

Unix End-of-line (LF)

Specifies whether to use the LF in Unix convention as the end-of-line character.

Run Linked Report

If the report is linked with other reports, you can check whether or not to generate the linked reports in the exported Text file. If you are only interested in the primary report, leave this option unchecked. Generating linked reports at the same time, especially when the linked reports contain a large amount of data, will cause performance issue.

OK

Applies all changes and exits the dialog.

Cancel

Cancels the changes and exits the dialog.

Help

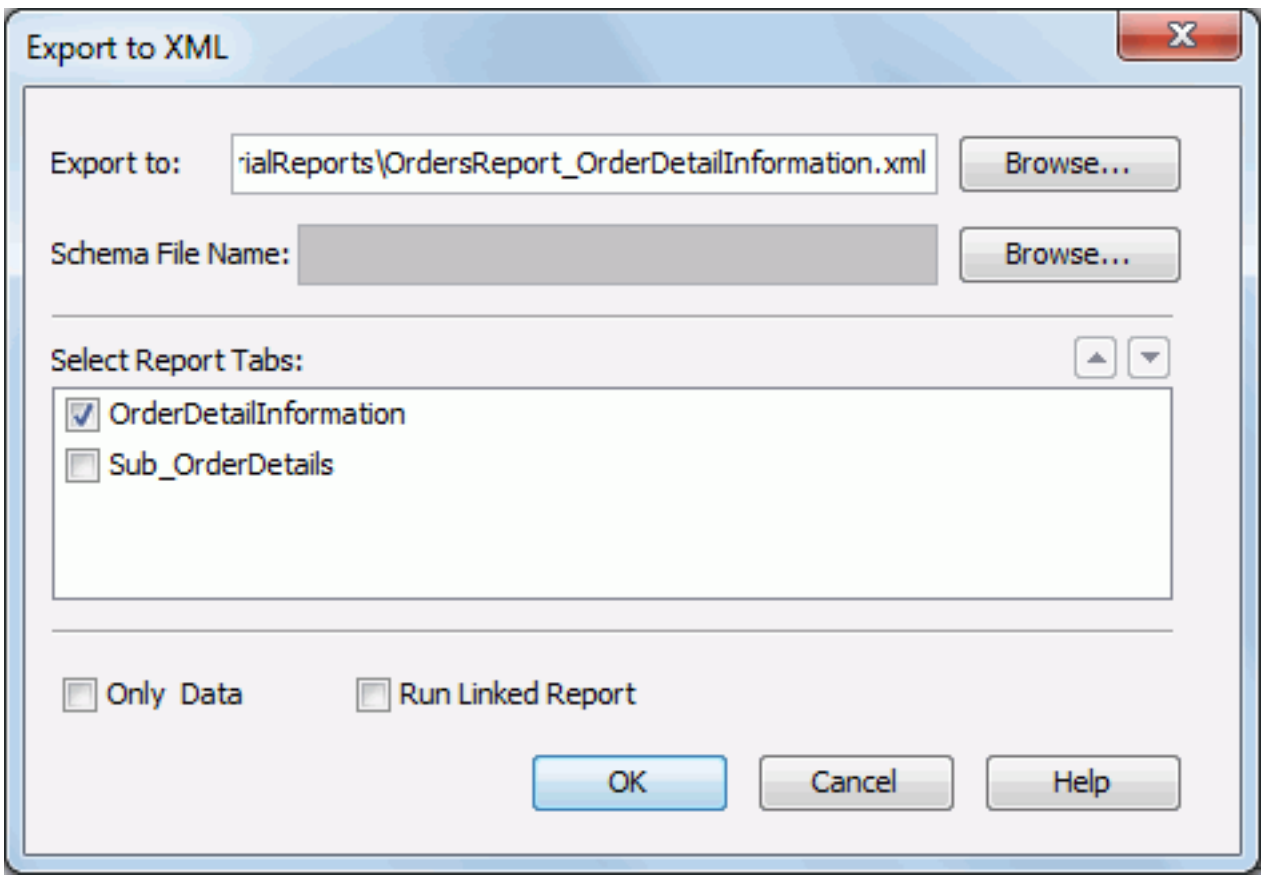
Displays the help document about this feature.

Related topics:

- Exporting report results - To Text

Export to XML dialog

This dialog appears when you click File > Export To > XML on the menu bar. It helps you to export a report to XML format.



The following are details about options in the dialog:

Export to

Specifies the file name and directory where you want to store the exported XML file.

Schema File Name

Specifies the directory and the name of an existing XML schema (.xsd) file. If you specify an existing XML schema file, the exported XML file, the generated XML file that is to be exported will be based on it. Otherwise, a new XML schema file will be generated to the directory where the exported XML file is to be exported. The new XML schema file and the exported XML file will have the same name but with different extensions.

Select Report Tabs

When exporting a page report, you can specify the report tabs in the page report you want to export to the XML file. The selected report tabs will be exported in the list order. If the report has only one report tab, it is selected by default.

Moves the specified report tab one step up.

Moves the specified report tab one step down.

Only Data

- If checked, the exported XML file will only contain the database column information. The exported XML schema file will only contain the structure information of the report.
- If unchecked, the exported XML file will also contain elements controlled by formulas, and the exported XML schema file will contain all the detailed information from the report, including all the property values of each report object.

Run Linked Report

If the report is linked with other reports, you can check whether or not to generate the linked reports in the exported XML file. If you are only interested in the primary report, leave this option unchecked. Generating linked reports at the same time, especially when the linked reports contain a large amount of data, will cause performance issue.

OK

Applies the changes and closes the dialog.

Cancel

Cancels the changes and exits the dialog.

Help

Displays the help document about this feature.

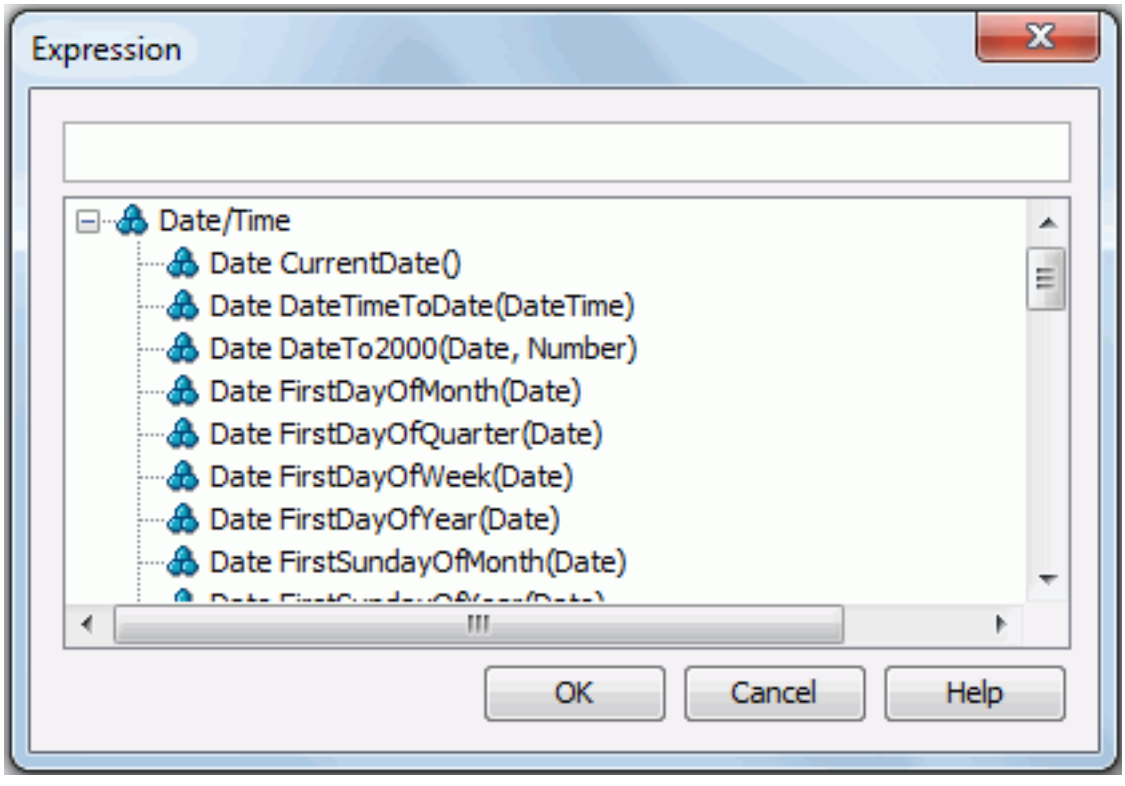
Related topics:

- Exporting report results - To XML

Expression dialog

JReport allows you to enter an expression for date parameters instead of choosing a specific date each time you run a report. For example, you may want to have a month to date report always has a begin date of the first day of the month and the end date as current date. By entering an expression, the defaults will always be the dates that you want to use.

The dialog appears when you click the button ... in the Calendar dialog. It provides some built-in functions of the Date/Time type for you to edit an expression.



The current expression is displayed at the top of the dialog. Edit it directly in the text field. If required, you can insert a function into the expression by double-clicking it from the built-in function list below. For details about the functions, see Date/Time.

OK

Saves the expression and closes the dialog.

Cancel

Does not retain any changes and closes the dialog.

Help

Displays the help document about this feature.

Related topics:

- Using built-in functions to set date and time parameter values

Fill Effects dialog

This dialog appears when you format a color schema for a chart. It helps you to specify either a gradient, a texture or an image as the fill effect, and consists of the following tabs:

- Gradient tab
- Texture tab
- Image tab

Default

Sets all the values in the dialog to be the default settings.

OK

Applies the changes and closes the dialog.

Cancel

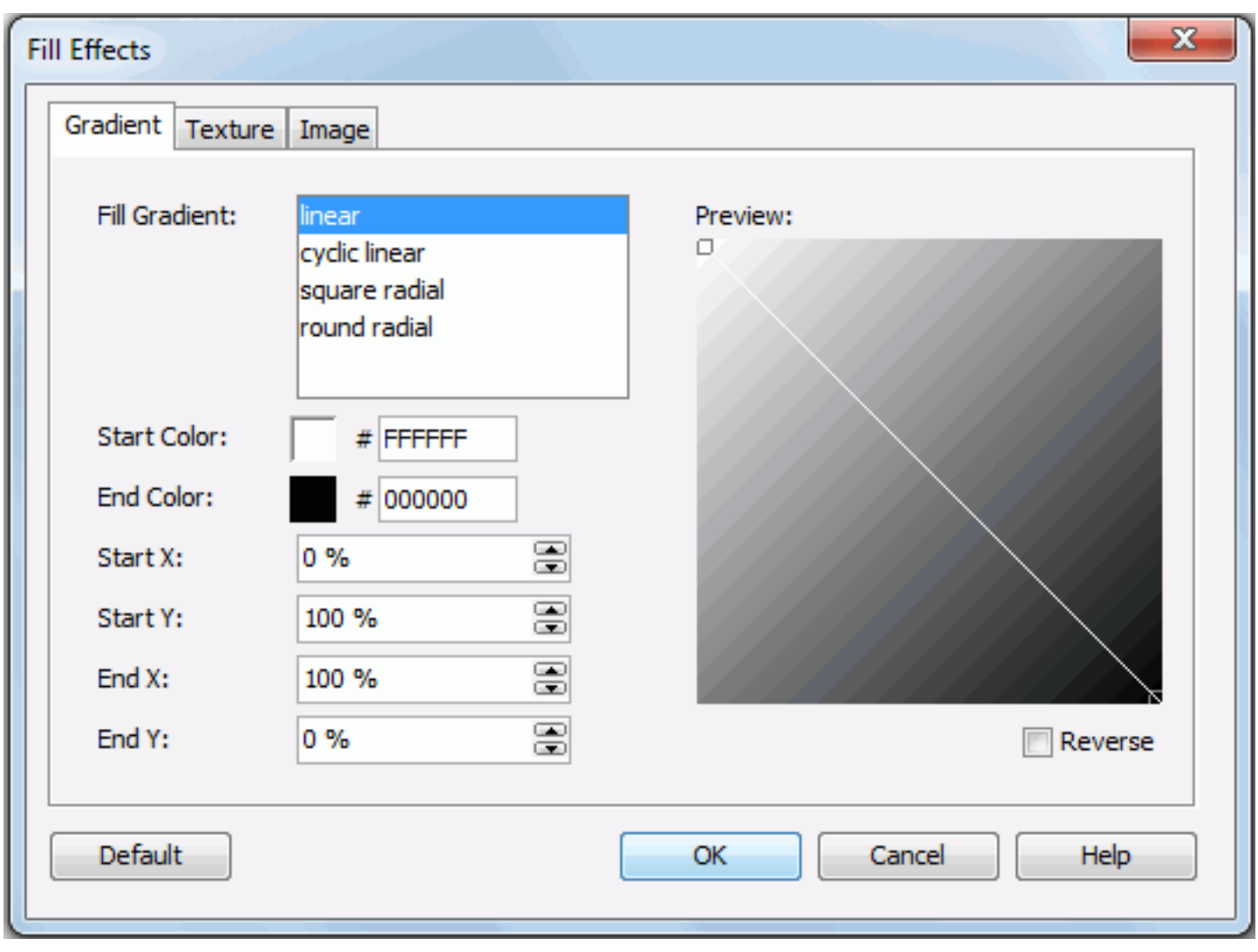
Does not retain any changes and closes the dialog.

Help

Displays the help document about this feature.

Gradient tab

Fills the object with specified gradient colors.



Fill Gradient

Specifies the type of gradient to fill the object. It can be one of the following: Linear, Cyclic Linear, Square Radial, and Round Radial.

Start Color

Specifies the color of the point where the gradient begins.

End Color

Specifies the color of the point where the gradient ends.

Start X

Specifies the horizontal position, measured in the percentage of the object's width, from the left edge of the object, where the gradient starts.

Start Y

Specifies the vertical position, measured in the percentage of the object's height, from the top edge of the object, where the gradient starts.

End X

Specifies the horizontal position, measured in the percentage of the object's width, from the left edge of the object, where the gradient ends.

End Y

Specifies the vertical position, measured in the percentage of the object's height, from the top edge of the object, where the gradient ends.

Preview

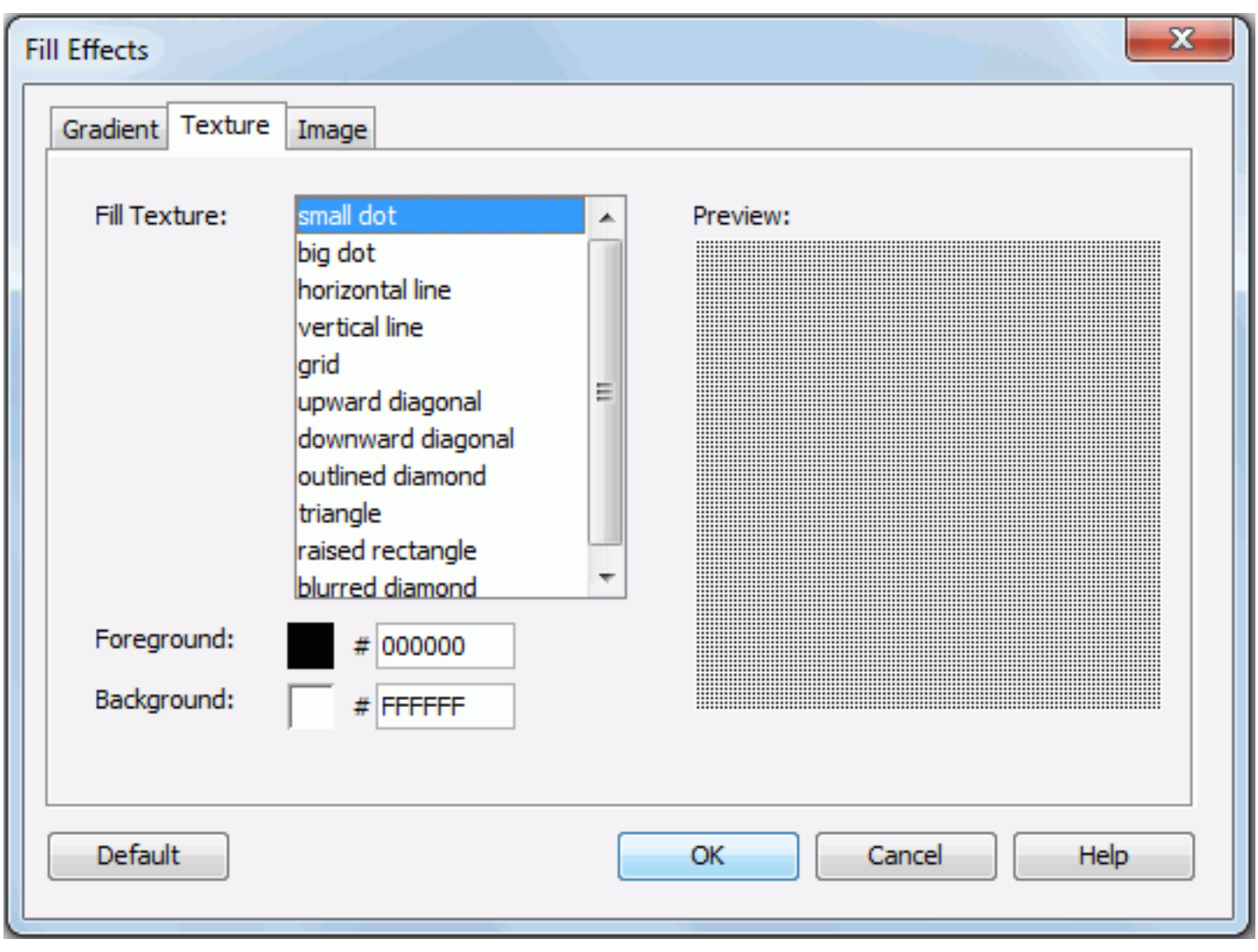
Displays a preview of your settings.

Reverse

Specifies whether to reverse values of Start X and End X, and reverse Start Y and End Y.

Texture tab

Fills the object with specified texture.



Fill Texture

Specifies the type of texture to fill the object.

Foreground

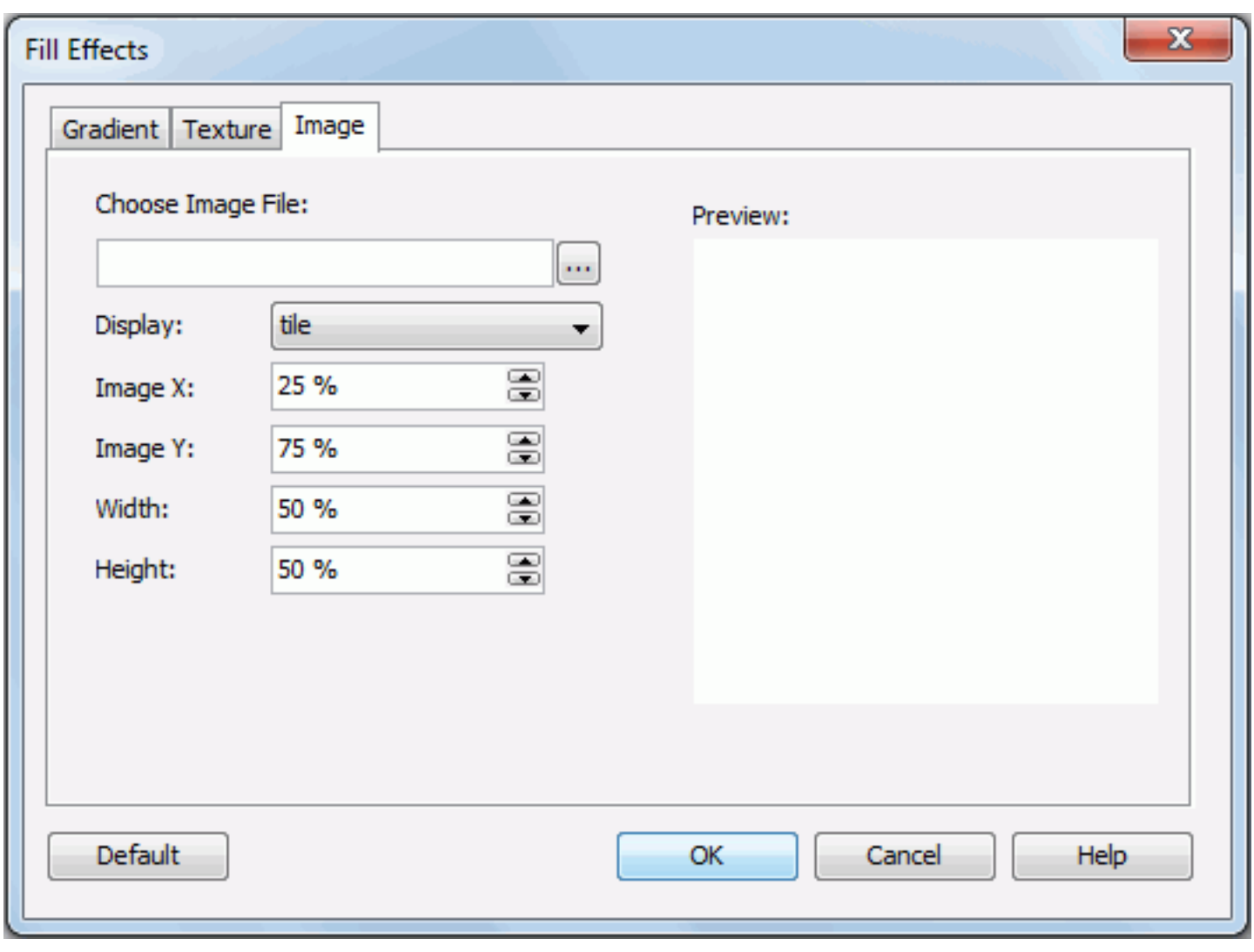
Specifies the foreground color of the texture.

Background

Specifies the background color of the texture.

Image tab

Fills the object with specified image. Available only for chart platform, chart paper and chart legend.



Choose Image File

Specifies the image file from the current catalog folder. Type the file path directly in the text field or click the Browse button to find the directory.

Display

Specifies the layout style of the image in the object. It can be on of the following: Tile, Centered and Scaled.

Image X

Specifies the left position of the area, measured in the percentage of the image's width, from the left edge of the image.

Image Y

Specifies the bottom position of the area, measured in the percentage of the image's width, from the bottom edge of the image.

Width

Specifies the width of the area, measured in the percentage of the image's width. Setting this value to less than 100 percent will cut the image from the right edge.

Height

Specifies the height of the area, measured in the percentage of the image's height. Setting this value to less than 100 percent will cut the image from the bottom edge.

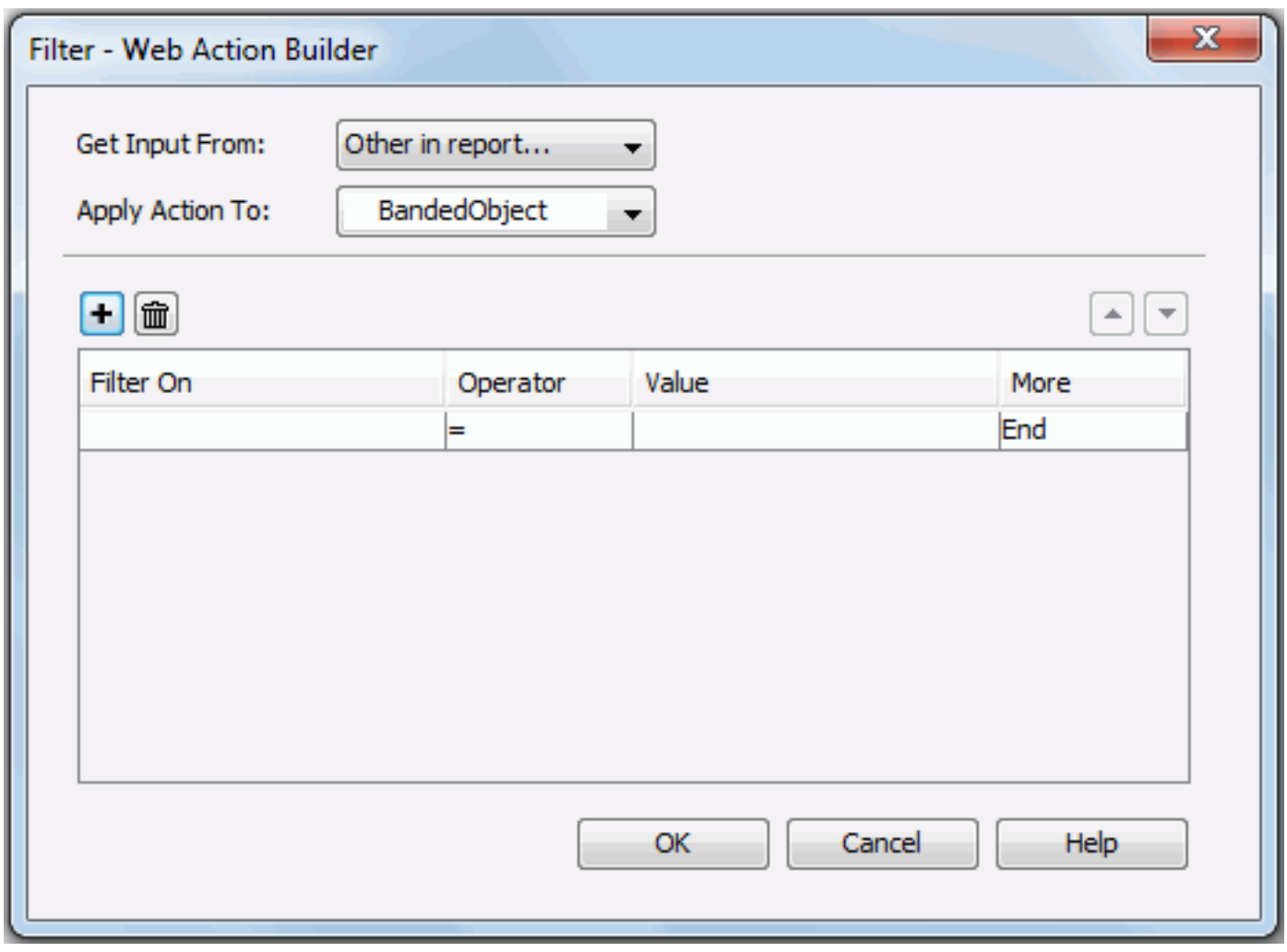
The above four options, namely Image X, Image Y, Width and Height, are used to define an area in the image and put the defined area of the image into the object.

Preview

Displays a preview of your settings.

Filter - Web Action Builder dialog

This dialog appears when you select *Filter and then click OK in the Web Action List dialog. It helps you to build a Filter web action on an object so as to filter data of the specified data components when the event bound with the action is triggered at runtime.



The following are details about options in the dialog:

Get Input From

Specifies where to get the object. You can select Form if you add some objects to a form. Otherwise, only the item Other in Report you can choose.

Apply Action To

Specifies the data component to which to apply the Filter action.

Adds a new filter condition.

Removes the selected filter condition.

Moves the selected filter condition up a step.

Moves the selected filter condition down a step.

Filter On

Specifies the field on which the filter is based. Which fields are available here depends on which object you choose from the Apply Action To drop-down list.

Operator

Specifies the operator to compose the filter expression.

- **=**
  Equal to
- **>**
  Greater than
- **<**
  Less than
- **>=**
  Greater than or equal to
- **<=**
  Less than or equal to
- **!=**
  Not equal to
- **in**
  Causes an enumerated list of values to appear in the WHERE clause predicate, used for evaluating for a true condition. For operator 'in', it is allowed to input multiple values in the Value input box and separate the values with comma (,).

Value

Specifies the value of how to filter the field. If multiple values are required, they should be separated with ",", and if "," or "\" is contained in the values, write it as "\," or "\\".

More

Lists some additional commands, which includes AND, OR, and End.

OK

Accepts the changes and closes this dialog.

Cancel

Does not retain any changes and closes the dialog.

Help

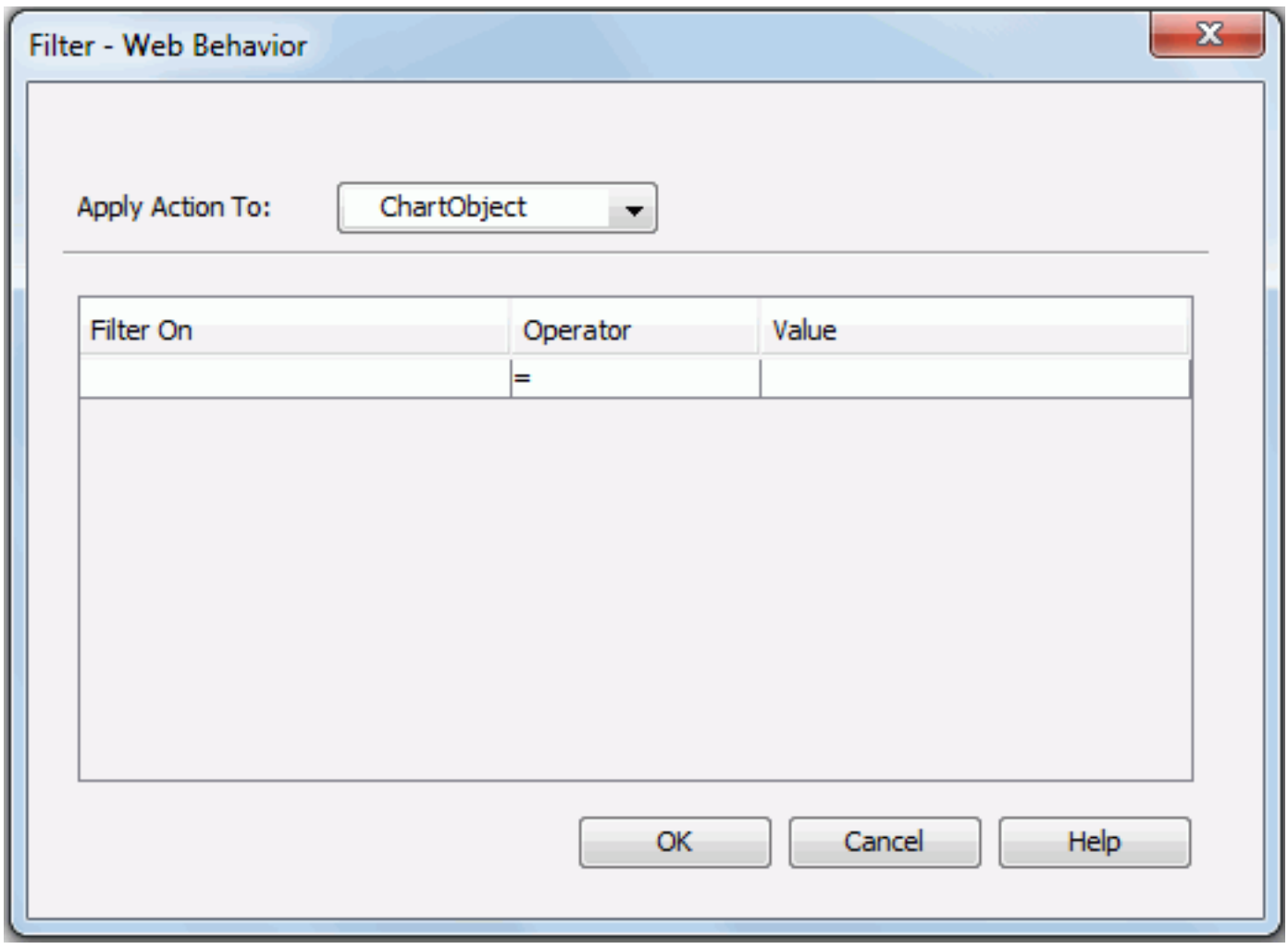
Displays the help document about this feature.

Related topics:

- Applying web actions to a label

Filter - Web Behavior dialog

This dialog appears when you right-click a web control in the configuration panel of a library component and select Web Behaviors > Filter from the shortcut menu. It helps you to build a Filter web behavior to the web control so as to filter data of the library component at runtime.



The following are details about options in the dialog:

Apply Action To

Specifies the data component to which to apply the filter action. Select the desired object from the drop-down list.

Filter On

Specifies the field on which the filter is based. Which fields are available here depends on which object you choose from the Apply Action To drop-down list.

Operator

Specifies the operator to compose the filter expression.

- **=**
  Equal to
- **>**
  Greater than
- **<**
  Less than
- **>=**
  Greater than or equal to
- **<=**
  Less than or equal to
- **!=**
  Not equal to
- **in**
  Causes an enumerated list of values to appear in the WHERE clause predicate, used for evaluating for a true condition. For operator 'in', it is allowed to input multiple values in the Value input box and separate the values with comma (,).

Value

Specifies the value of how to filter the field. The value may be input by yourself, or be specified by the value of a web control. If multiple values are required, they should be separated with ",", and if "," or "\" is contained in the values, write it as "\," or "\\".

OK

Accepts the changes and closes this dialog.

Cancel

Does not retain any changes and closes the dialog.

Help

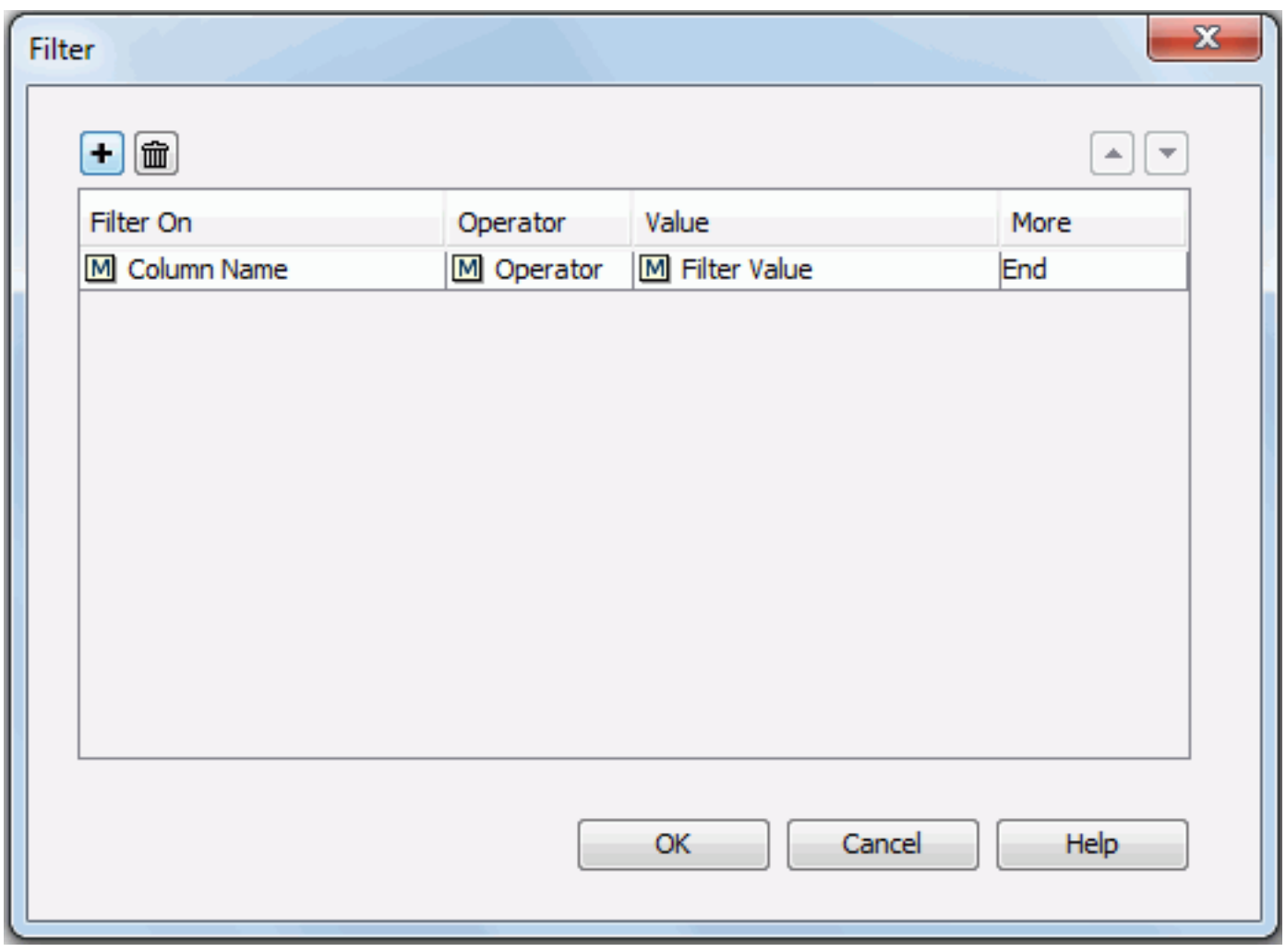
Displays the help document about this feature.

Related topics:

- Applying web behaviors to web controls in the panel

Filter dialog

This dialog helps you to define to filter data of the current library component as a response to the message the library component receives at runtime. It is displayed when you select 0001 - Filter from the drop-down list of the Message ID column and then click ... in the Actions column of the Receive Message dialog, or when you select *Filter and then click OK in the Web Action List dialog.



The following are details about options in the dialog:

Adds a new filter condition.

Removes the selected filter condition.

Moves the selected filter condition up a step.

Moves the selected filter condition down a step.

Filter On

Specifies the field on which the filter is based.

Operator

Specifies the operator to compose the filter expression.

- **=**
  Equal to
- **>**
  Greater than
- **<**
  Less than
- **>=**
  Greater than or equal to
- **<=**
  Less than or equal to
- **!=**
  Not equal to
- **in**
  Causes an enumerated list of values to appear in the WHERE clause predicate, used for evaluating for a true condition. For operator 'in', it is allowed to input multiple values in the Value input box and separate the values with comma (,).

Value

Specifies the value of how to filter the field. If multiple values are required, they should be separated with ",", and if "," or "\" is contained in the values, write it as "\," or "\\".

More

Lists some additional commands, which includes AND, OR, and End.

OK

Accepts the changes and closes this dialog.

Cancel

Does not retain any changes and closes the dialog.

Help

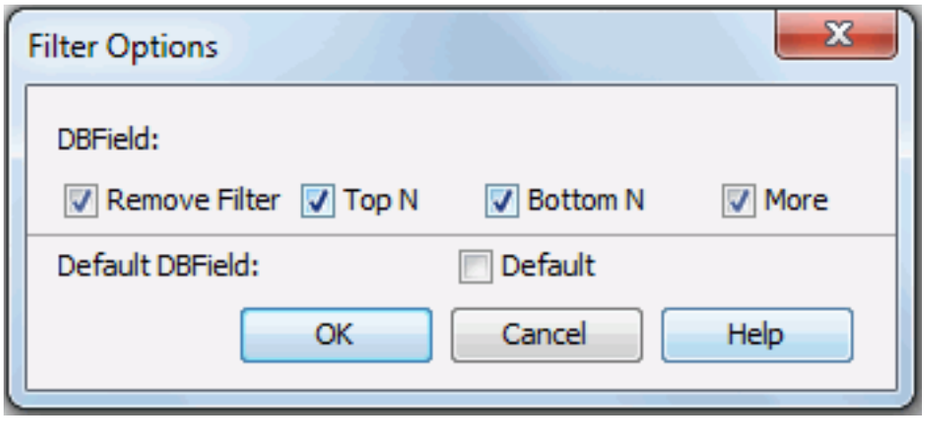
Displays the help document about this feature.

Related topics:

- Receiving messages

Filter Options dialog

This dialog appears when you click ... in the value cell to specify the Filter Options property for a field in the Report Inspector. It helps you to define whether or not the filter-related shortcut menu items will be displayed when you right-click an object in Page Report Studio.



The following are details about options in the dialog:

Remove Filter

Specifies whether to display the Remove Filter item on the shortcut menu when you right-click an object in Page Report Studio.

Top N

Specifies whether to display the Top N item on the shortcut menu when you right-click an object in Page Report Studio.

Bottom N

Specifies whether to display the Bottom N item on the shortcut menu when you right-click an object in Page Report Studio.

More

Specifies whether to display the More item on the shortcut menu when you right-click an object in Page Report Studio.

Default

Specifies whether to apply the default settings.

OK

Applies the changes and closes the dialog.

Cancel

Cancels the changes and exits the dialog.

Help

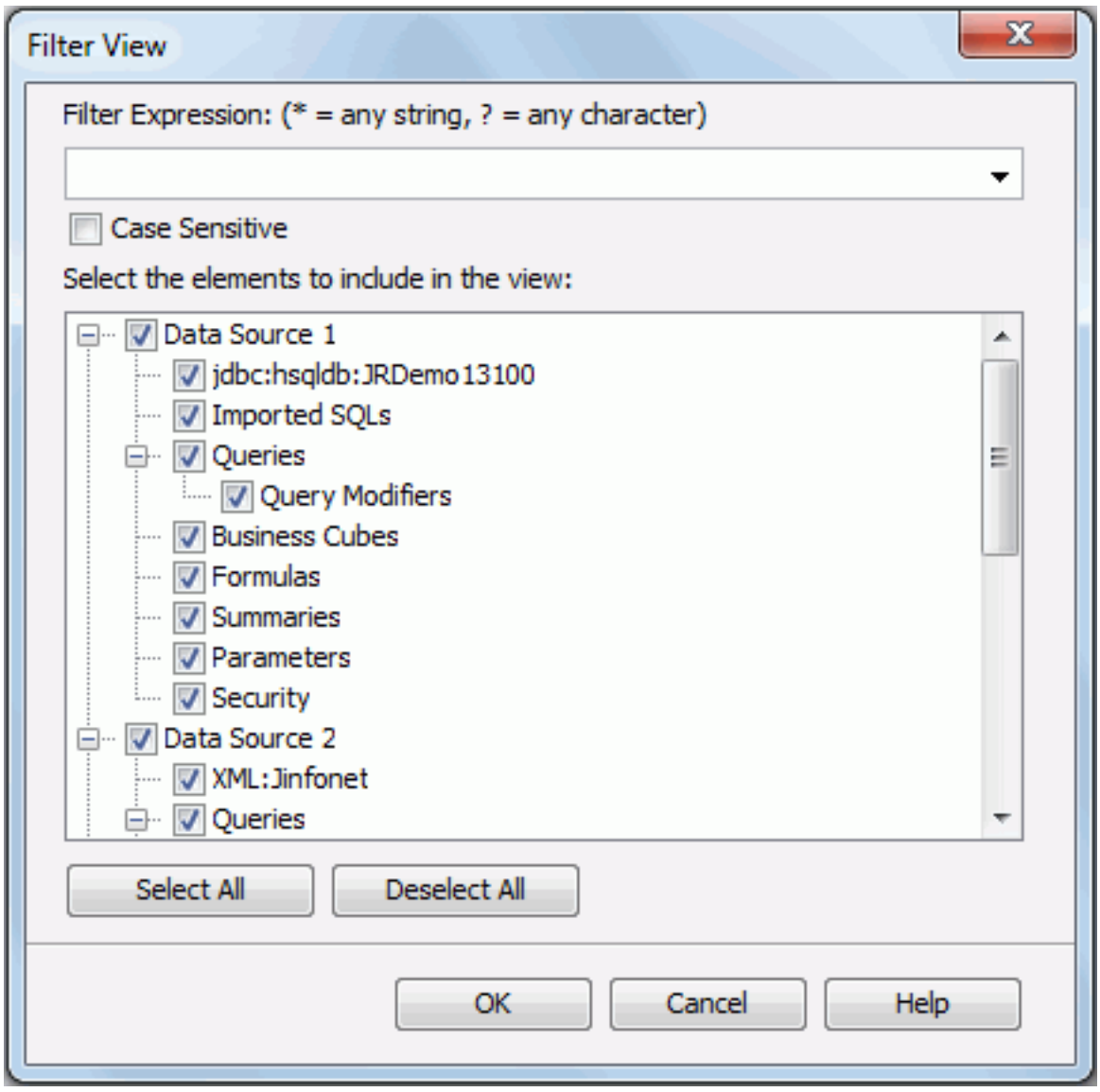
Displays the help document about this feature.

Related topics:

- Setting filter options for a field

Filter View dialog

This dialog appears when you click the Filter View button on the Catalog Browser toolbar. It enables you to select some elements to exclude from the view in the Catalog Browser, if you think they are not necessary to display.



The following are details about options in the dialog:

Filter Expression

Specifies the resource names that are to be displayed.

Case Sensitive

Specifies whether to match whole work only or match case.

Select the elements to include in the view

Lists all the elements in the Catalog Browser.

Select All

Selects all the elements in the list box to include them in the view.

Deselect All

Deselects all the elements in the list box to exclude them from the view.

OK

Applies all changes and closes the dialog.

Cancel

Does not retain any changes and closes the dialog.

Help

Displays the help document about this feature.

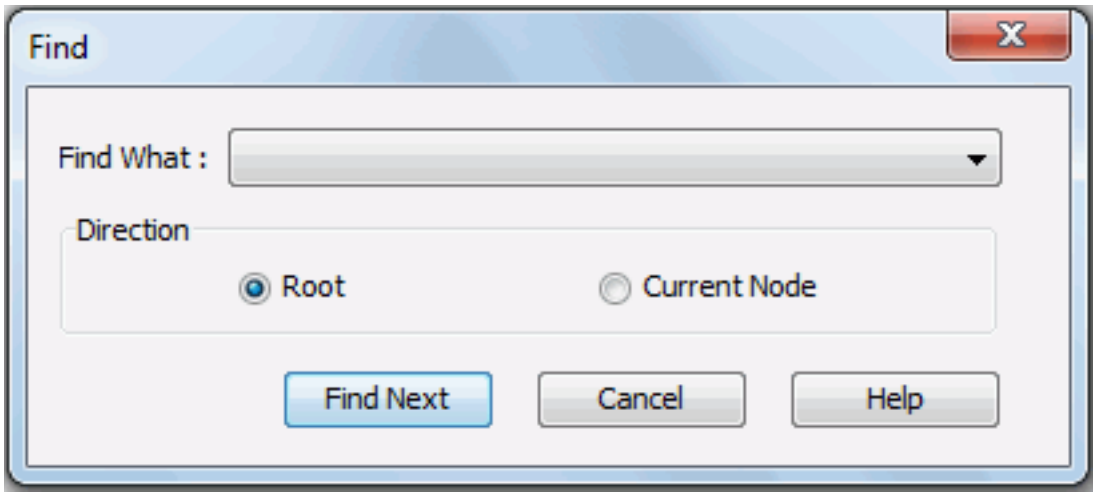
Related topics:

- Filtering resources

Find dialog

This dialog appears when you click Edit > Find DBField on the menu bar, or the Find DBField button on the Standard toolbar.

When a report has many fields, you may find it difficult to determine where a certain field is. This dialog helps you to find specific fields, including DBFields and formula fields, you have lost track of in your report.



The following are details about options in the dialog:

Find What

Specifies the field you want to find.

Direction

Specifies the criteria with which to find the field.

- **Root**
  Searches for the field from the top node.
- **Current Node**
  Searches for the field from the current node.

Find Next

Jumps to and selects the next occurrence of the searching field in the report.

Cancel

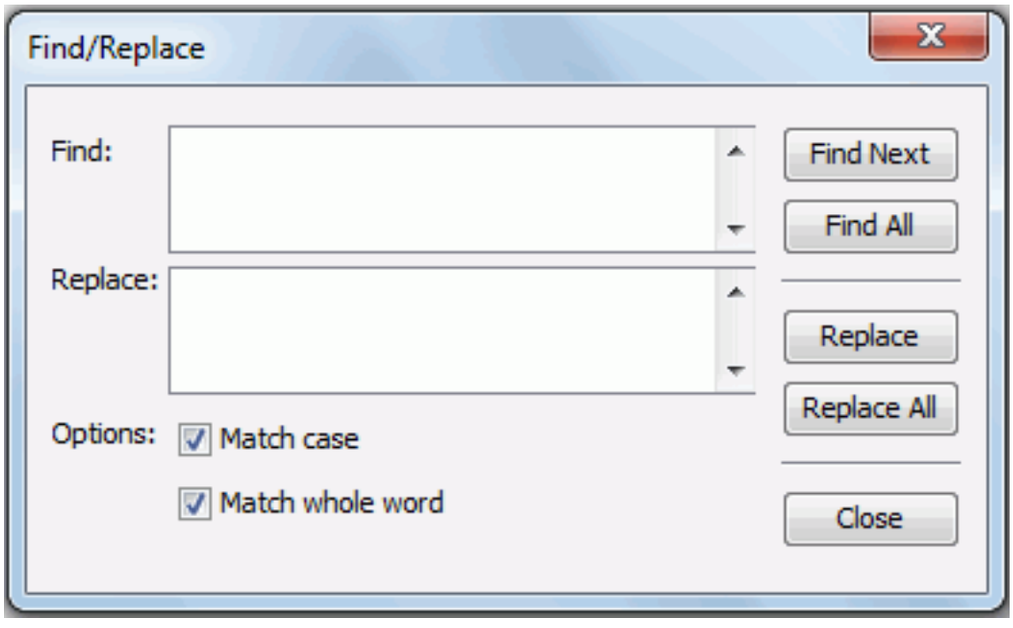
Ends the search and closes the dialog.

Help

Displays the help document about this feature.

Find/Replace dialog

This dialog appears when you click the Search button in the Formula Editor window. You can then search for text in the formula text, and replace the found text with different text.



The following are details about options in the dialog:

Find

Specifies what to find.

Replace

Specifies what to use to replace the found text.

Options

Specifies the finding limit.

- **Match case**
  Limits the search to text that exactly matches the case of the text you want to find.
- **Match whole word**
  Limits the search to text that matches one or more complete words.

Find Next

Jumps to and selects the next occurrence of the search words in the Formula Text.

Find All

Finds all occurrences of the search words in the Formula Text.

Replace

Replaces the found material one by one in the Formula Text.

Replace All

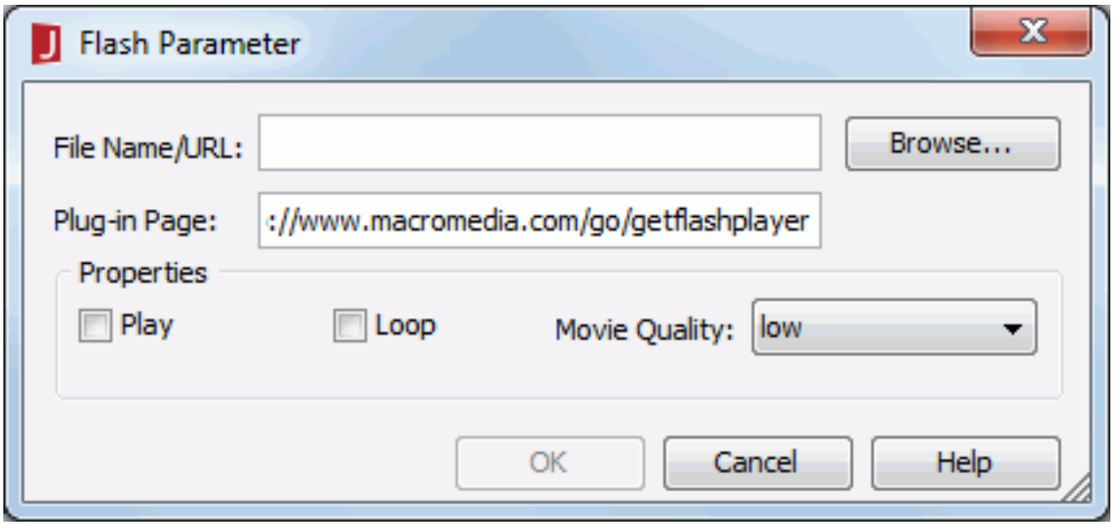
Replaces all of the found materials in the Formula Text.

Close

Closes the dialog.

Flash Parameter dialog

This dialog appears when you click Insert > Multimedia Objects > Flash on the menu bar, or drag Flash from the Toolbox panel to a report. It helps you to insert a flash object (using the OLE mode) into a report.



The following are details about options in the dialog:

File Name/URL

Specifies the URL of the plug-in page from which to download the player to play this flash file, if it isn't already installed on your local disk.

Plug-in Page

Specifies the URL of the plug-in page to play the flash file.

Properties

Specifies the properties for the flash file that is to be inserted into the report.

- **Play**
  Specifies whether to play the flash file.
- **Loop**
  Specifies whether to play the flash file repeatedly.
- **Movie Quality**
  Specifies the quality for the flash file from the drop-down list.

OK

Accepts all changes and closes the dialog.

Cancel

Does not retain any changes and closes the dialog.

Help

Displays the help document about this feature.

Related topics:

- Multimedia objects

Format Area dialog for library component

This dialog appears when you double-click an area of a chart in a library component, or right-click it and then select Format Area from the shortcut menu. It helps you to format areas in a chart, and consists of the following tabs:

- General tab
- Fill tab
- Border tab
- Data Label tab
- Behaviors tab

OK

Applies the changes and closes the dialog.

Cancel

Does not retain any changes and closes the dialog.

Apply

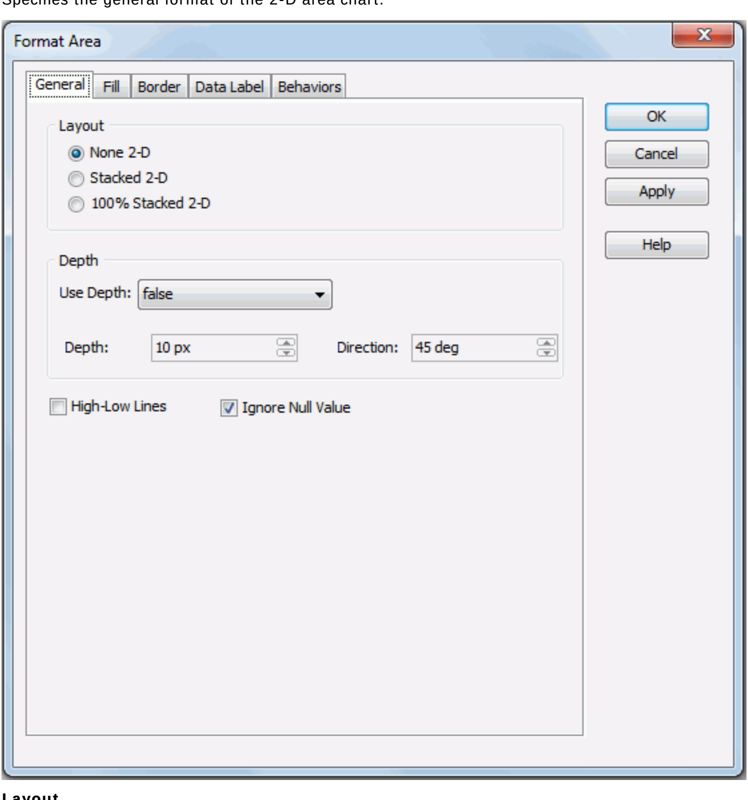
Applies all changes and leaves the dialog open.

Help

Displays the help document about this feature.

General tab

Specifies the general format of the 2-D area chart.



Layout

Specifies the layout for 2-D areas of the chart.

- **None 2-D**
  If selected, 2-D area will be applied to display the trend of the values over time or categories.
- **Stacked 2-D**
  If selected, stacked 2-D area will be applied to display the trend of the contribution of each data value over categories.
- **100% Stacked 2-D**
  If selected, 100% stacked 2-D area will be applied to display the trend of the percentage each data value contributes over categories.

Depth

Specifies the depth properties for areas of the chart.

- **Use Depth**
  Specifies whether to make the areas in the chart three-dimensional. You can also use a formula to control the property.
  - **Depth**
    Specifies the depth of the areas, in pixels.
  - **Direction**
    Specifies the direction for depth of the lines, in degrees.

High-Low Lines

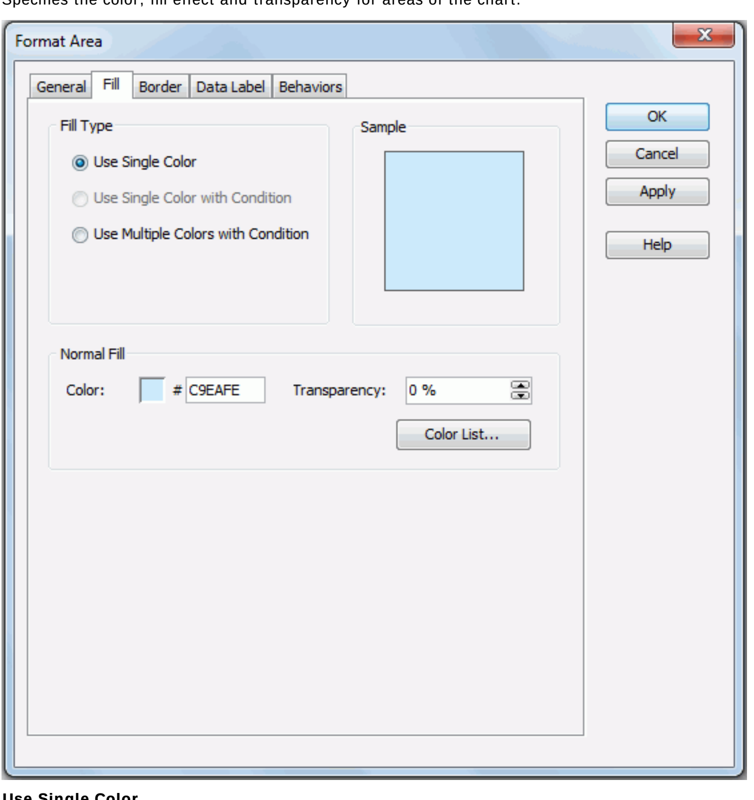
Specifies whether to show the high-low lines in areas of a chart.

Ignore Null Value

Specifies whether to ignore null data values when drawing areas in the chart. If checked, when a null data value appears, it is ignored and the area is drawn from the previous data value to the next data value directly, otherwise, the area is broken at the point of the null data value.

Fill tab

Specifies the color, fill effect and transparency for areas of the chart.



Use Single Color

If checked, the selected area will use the single color pattern.

- **Color**
  Specifies the color schema for the selected area.

  To change the color, click the color image button to display the color palette in which you can select a color, or click **More Colors** to customize the color within a wider range in the Pick a Color dialog, or click **More Fill Effects** to customize in the Fill Effects dialog. You can also input the hexadecimal value of a color directly in the text box after # if you have changed the default color.
- **Transparency**
  Specifies the transparency of the color schema.
- **Color List**
  Opens the Color List dialog to modify color pattern for each area.

Use Single Color with Condition

Not supported on area chart.

Use Multiple Colors with Condition

If checked, each area can have multiple color patterns based on defined conditions. You can divide each area into different parts based on different value ranges, along the direction of the value axis, and then specify different conditional colors to different value ranges for distinguishing.

- **Select Field**
  Lists the value field on which you can apply the conditional fill.
- **Start Value**
  Specifies the start value of the condition.
- **End Value**
  Specifies the end value of the condition.
- **Color**
  Specifies the color that will be applied to the values which meet the condition.

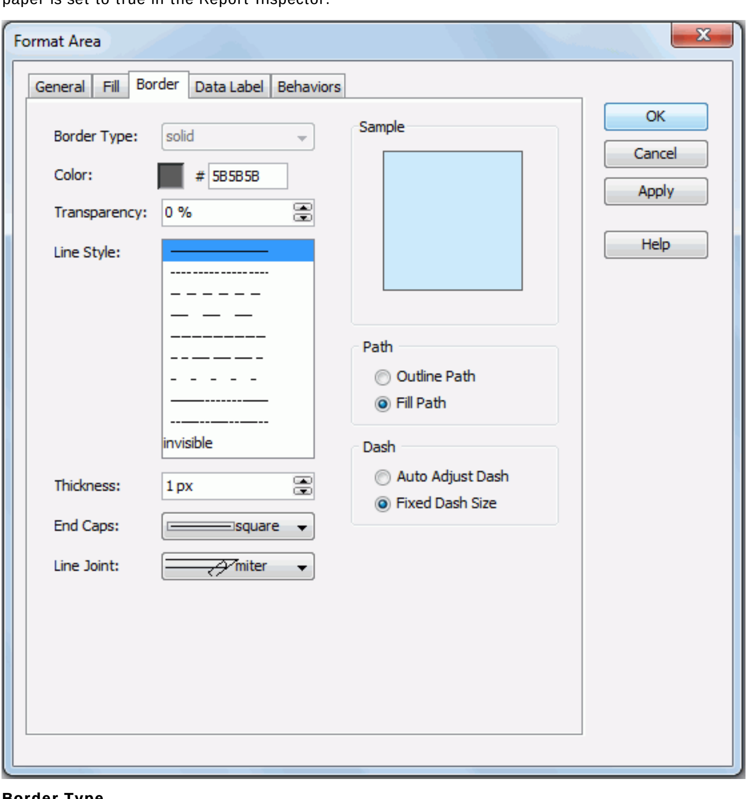
  To change the color, click the color image button to display the color palette in which you can select a color, or click **More Colors** to customize the color within a wider range in the Pick a Color dialog, or click **More Fill Effects** to customize in the Fill Effects dialog. You can also input the hexadecimal value of a color directly in the text box after #.
- **Transparency**
  Specifies the transparency for the color of the condition.
- Adds a new condition line.
- Removes the selected condition line.
- **Label**
  If checked, you can modify the condition expression of the selected condition line which will be shown as the legend entry label.

Sample

Displays a preview sample of your selection.

Border tab

Specifies properties for border of the areas, which take effect only when the Border property on chart paper is set to true in the Report Inspector.



Border Type

Displays the type for border of the areas. Its default value is solid, and cannot be changed.

Color

Specifies the color for border of the areas.

To change the color, click the color image button to display the color palette in which you can select a color, or click **More Colors** to customize the color within a wider range in the Pick a Color dialog. You can also input the hexadecimal value of a color directly in the text box after #.

Transparency

Specifies the transparency for color of the border.

Line Style

Specifies the line style to apply to border of the areas.

Thickness

Specifies the thickness of the border, in pixels.

End Caps

Specifies the ending style of the border line.

- **butt**
  Ends unclosed sub paths and dash segments with no added decoration.
- **round**
  Ends unclosed sub paths and dash segments with a round decoration that has a radius equal to half of the width of the pen.
- **square**
  Ends unclosed sub paths and dash segments with a square projection that extends beyond the end of the segment to a distance equal to half of the line width.

Line Joint

Specifies the line joint style for the border line.

- **miter**
  Joins path segments by extending their outside edges until they meet.
- **round**
  Joins path segments by rounding off the corner at a radius of half the line width.
- **bevel**
  Joins path segments by connecting the outer corners of their wide outlines with a straight segment.

Sample

Displays a preview sample of your selection.

Path

Specifies the fill pattern of the border line.

- **Outline Path**
  Specify the fill pattern of the border line to be outline path.
- **Fill Path**
  Specify the fill pattern of the border line to be whole path.

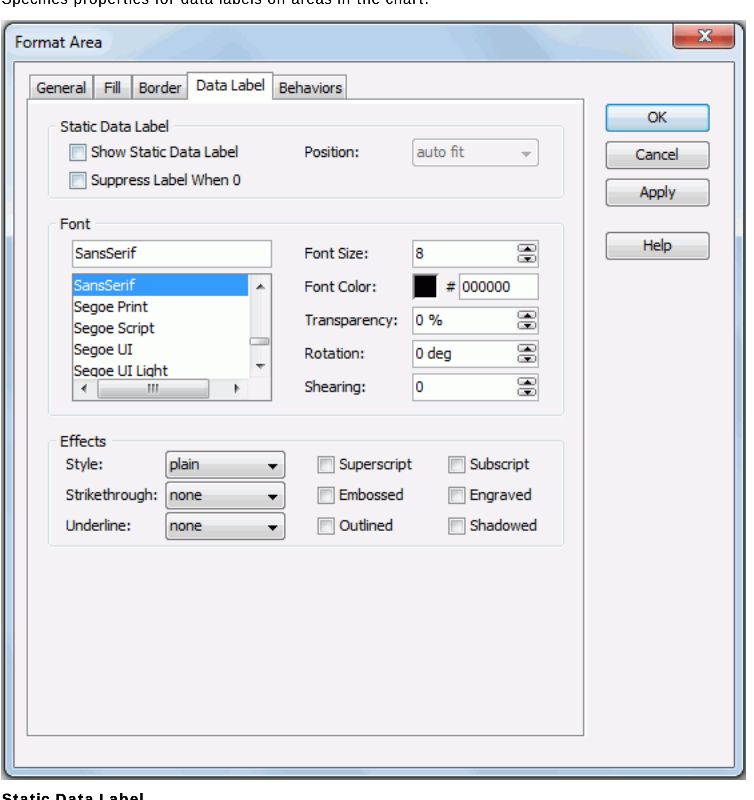
Dash

Specifies the dash size of the border line.

- **Auto Adjusted Dash**
  If selected, the dash size will be adjusted automatically.
- **Fixed Dash Size**
  If selected, the dash size will be fixed size.

Data Label tab

Specifies properties for data labels on areas in the chart.



Static Data Label

Specifies properties of the static data labels on the areas. Not supported on 3-D area charts.

- **Show Static Data Label**
  Specifies whether or not to show the static data labels. Only when it is checked can the following static data label related properties take effect.
- **Position**
  Specifies the position of the static data labels on the areas.
  - **auto fit**
    If selected, the static data labels will be displayed automatically.
  - **top center**
    If selected, the static data labels will be displayed in the top center of the nodes on the areas.
  - **top left**
    If selected, the static data labels will be displayed on the top left of the nodes on the areas.
  - **top right**
    If selected, the static data labels will be displayed on the top right of the nodes on the areas.
  - **bottom left**
    If selected, the static data labels will be displayed on the bottom left of the nodes on the areas.
  - **bottom center**
    If selected, the static data labels will be displayed in the bottom center of the nodes on the areas.
  - **bottom right**
    If selected, the static data labels will be displayed on the bottom right of the nodes on the areas.
- **Suppress Label When 0**
  If true, the static data label whose value is 0 will not be displayed on the chart.

Font

Specifies the font format of text in the data labels.

- **Font list**
  Lists all the available font faces that can be selected to apply to the text.
- **Font Size**
  Specifies the font size of the text.
- **Font Color**
  Specifies the font color of the text.

  To change the color, click the color image button to display the color palette in which you can select a color, or click **More Colors** to customize the color within a wider range in the Pick a Color dialog, or click **More Fill Effects** to customize in the Fill Effects dialog. You can also input the hexadecimal value of a color directly in the text box after #.
- **Transparency**
  Specifies the transparency of the text.
- **Rotation**
  Specifies the rotation angle of the text around its center, in degrees. The default value is 0.
- **Shearing**
  Specifies the gradient of the text.

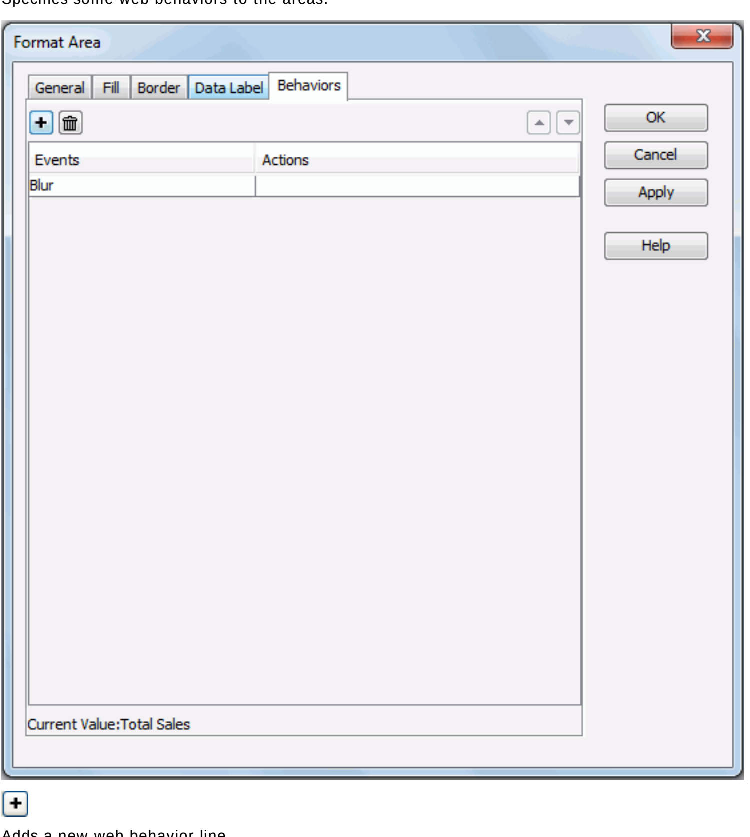
Effects

Specifies the special effects of text in the data labels.

- **Style**
  Specifies the font style of the text. It can be one of the following: plain, bold, italic, and bold italic.
- **Strikethrough**
  Specifies the style of the horizontal line with which the text is struck through. It can be one of the following: none, thin line, bold line, and double lines.
- **Underline**
  Specifies the style of the horizontal line under the text. It can be one of the following: none, single, single lower, bold line, bold lower, double lines, bold double, patterned line, and bold patterned.
- **Superscript**
  Raises the text above the baseline and changes the text to a smaller font size, if a smaller size is available.
- **Embossed**
  Makes the text appear to be raised off the page in relief.
- **Outlined**
  Displays the inner and outer borders of each character.
- **Subscript**
  Lowers the text below the baseline and changes the text to a smaller font size, if a smaller size is available.
- **Engraved**
  Makes the text appear to be imprinted or pressed into the page.
- **Shadowed**
  Adds a shadow beneath and to the right of the text.

Behaviors tab

Specifies some web behaviors to the areas.



Adds a new web behavior line.

Removes the selected web behavior.

Moves the selected web behavior up a step.

Moves the selected web behavior down a step.

Events

Specifies the trigger event.

Actions

Specifies the action you want the event to trigger.

- Opens the Web Action List dialog to bind a web action to the event.

Related topics:

- Formatting the areas

Format Area dialog for page report

This dialog appears when you double-click an area of a chart in a page report, or right-click it and then select Format Area from the shortcut menu. It helps you to format areas in a chart, and consists of the following tabs:

- General tab
- Fill tab
- Border tab
- Data Label tab

OK

Applies the changes and closes the dialog.

Cancel

Does not retain any changes and closes the dialog.

Apply

Applies all changes and leaves the dialog open.

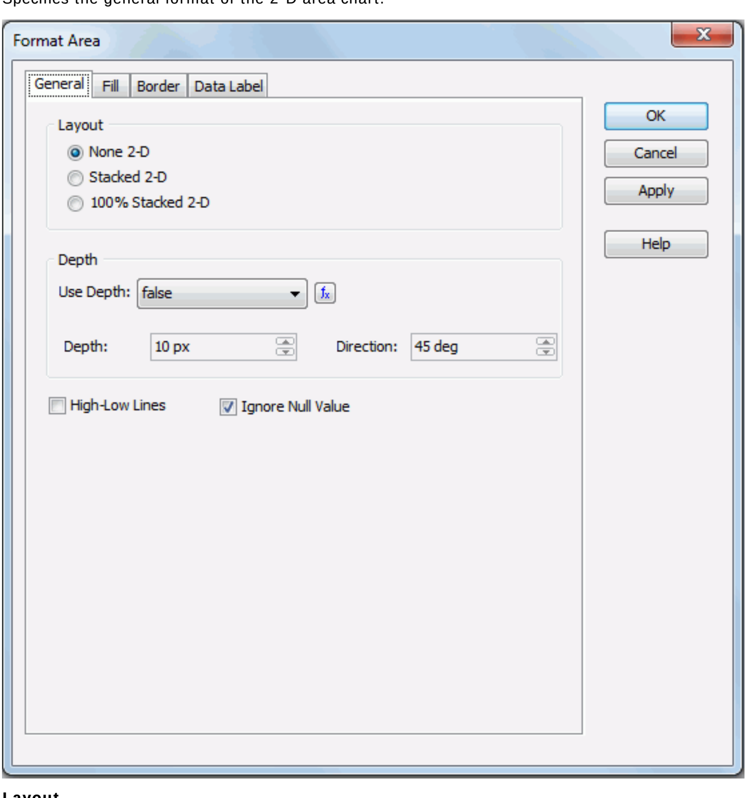
Help

Displays the help document about this feature.

General tab

For 2-D area charts

Specifies the general format of the 2-D area chart.



Layout

Specifies the layout for 2-D areas of the chart.

- **None 2-D**
  If selected, 2-D area will be applied to display the trend of the values over time or categories.
- **Stacked 2-D**
  If selected, stacked 2-D area will be applied to display the trend of the contribution of each data value over categories.
- **100% Stacked 2-D**
  If selected, 100% stacked 2-D area will be applied to display the trend of the percentage each data value contributes over categories.

Depth

Specifies the depth properties for areas of the chart.

- **Use Depth**
  Specifies whether to make the areas in the chart three-dimensional. You can also use a formula to control the property.
  - **Depth**
    Specifies the depth of the areas, in pixels.
  - **Direction**
    Specifies the direction for depth of the lines, in degrees.

High-Low Lines

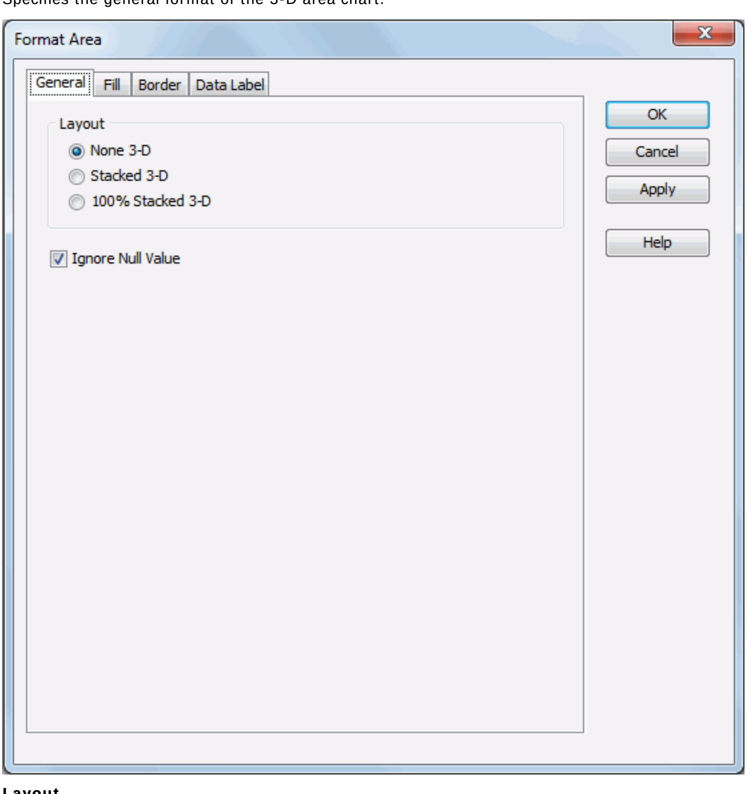
Specifies whether to show the high-low lines in areas of a chart.

Ignore Null Value

Specifies whether to ignore null data values when drawing areas in the chart. If checked, when a null data value appears, it is ignored and the area is drawn from the previous data value to the next data value directly, otherwise, the area is broken at the point of the null data value.

For 3-D area charts

Specifies the general format of the 3-D area chart.



Layout

Specifies the layout for 3-D areas of the chart.

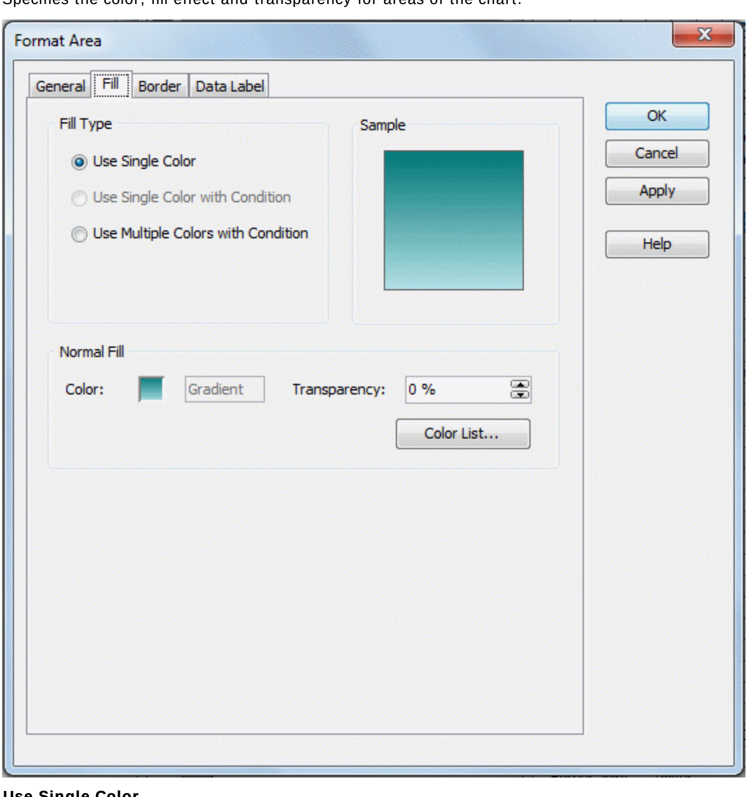
- **None 3-D**
  If selected, area with a 3-D visual effect will be applied.
- **Stacked 3-D**
  If selected, stacked area with a 3-D visual effect will be applied.
- **100% Stacked 3-D**
  If selected, 100% stacked area with a 3-D visual effect will be applied.

Ignore Null Value

Specifies whether to ignore null data values when drawing areas in the chart. If checked, when a null data value appears, it is ignored and the area is drawn from the previous data value to the next data value directly, otherwise, the area is broken at the point of the null data value.

Fill tab

Specifies the color, fill effect and transparency for areas of the chart.



Use Single Color

If checked, the selected area will use the single color pattern.

- **Color**
  Specifies the color schema for the selected area.

  To change the color, click the color image button to display the color palette in which you can select a color, or click **More Colors** to customize the color within a wider range in the Pick a Color dialog, or click **More Fill Effects** to customize in the Fill Effects dialog. You can also input the hexadecimal value of a color directly in the text box after # if you have changed the default color.
- **Transparency**
  Specifies the transparency of the color schema.
- **Color List**
  Opens the Color List dialog to modify color pattern for each area.

Use Single Color with Condition

Not supported on area chart.

Use Multiple Colors with Condition

If checked, each area can have multiple color patterns based on defined conditions. You can divide each area into different parts based on different value ranges, along the direction of the value axis, and then specify different conditional colors to different value ranges for distinguishing.

- **Select Field**
  Lists the value field on which you can apply the conditional fill.
- **Start Value**
  Specifies the start value of the condition.
- **End Value**
  Specifies the end value of the condition.
- **Color**
  Specifies the color that will be applied to the values which meet the condition.

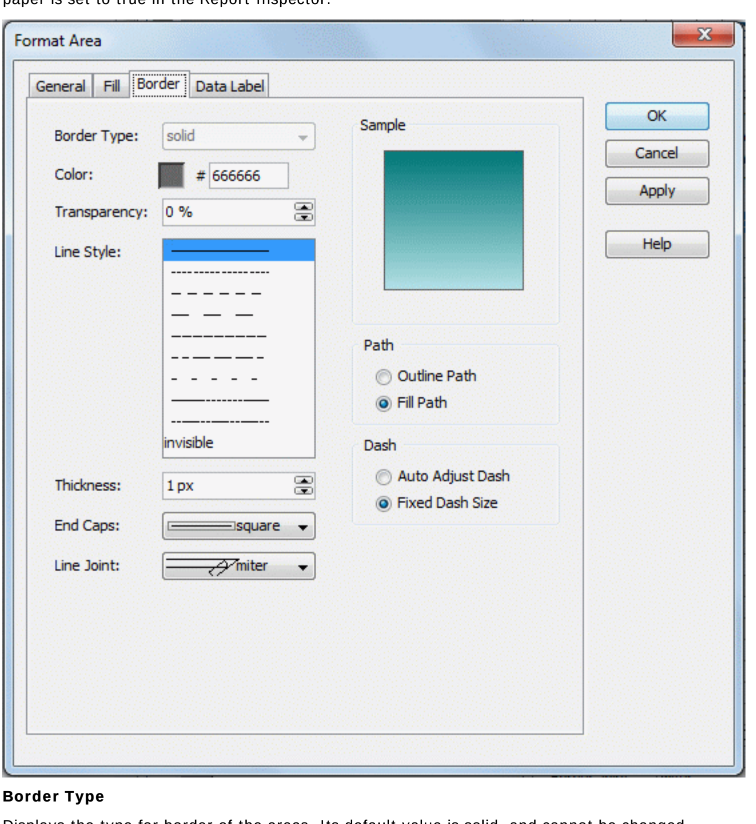
  To change the color, click the color image button to display the color palette in which you can select a color, or click **More Colors** to customize the color within a wider range in the Pick a Color dialog, or click **More Fill Effects** to customize in the Fill Effects dialog. You can also input the hexadecimal value of a color directly in the text box after #.
- **Transparency**
  Specifies the transparency for the color of the condition.
- [+]
  Adds a new condition line.
- [delete]
  Removes the selected condition line.
- **Label**
  If checked, you can modify the condition expression of the selected condition line which will be shown as the legend entry label.

Sample

Displays a preview sample of your selection.

Border tab

Specifies properties for border of the areas, which take effect only when the Border property on chart paper is set to true in the Report Inspector.



Border Type

Displays the type for border of the areas. Its default value is solid, and cannot be changed.

Color

Specifies the color for border of the areas.

To change the color, click the color image button to display the color palette in which you can select a color, or click **More Colors** to customize the color within a wider range in the Pick a Color dialog. You can also input the hexadecimal value of a color directly in the text box after #.

Transparency

Specifies the transparency for color of the border.

Line Style

Specifies the line style to apply to border of the areas.

Thickness

Specifies the thickness of the border, in pixels.

End Caps

Specifies the ending style of the border line.

- **butt**
  Ends unclosed sub paths and dash segments with no added decoration.
- **round**
  Ends unclosed sub paths and dash segments with a round decoration that has a radius equal to half of the width of the pen.
- **square**
  Ends unclosed sub paths and dash segments with a square projection that extends beyond the end of the segment to a distance equal to half of the line width.

Line Joint

Specifies the line joint style for the border line.

- **miter**
  Joins path segments by extending their outside edges until they meet.
- **round**
  Joins path segments by rounding off the corner at a radius of half the line width.
- **bevel**
  Joins path segments by connecting the outer corners of their wide outlines with a straight segment.

Sample

Displays a preview sample of your selection.

Path

Specifies the fill pattern of the border line.

- **Outline Path**
  Specify the fill pattern of the border line to be outline path.
- **Fill Path**
  Specify the fill pattern of the border line to be whole path.

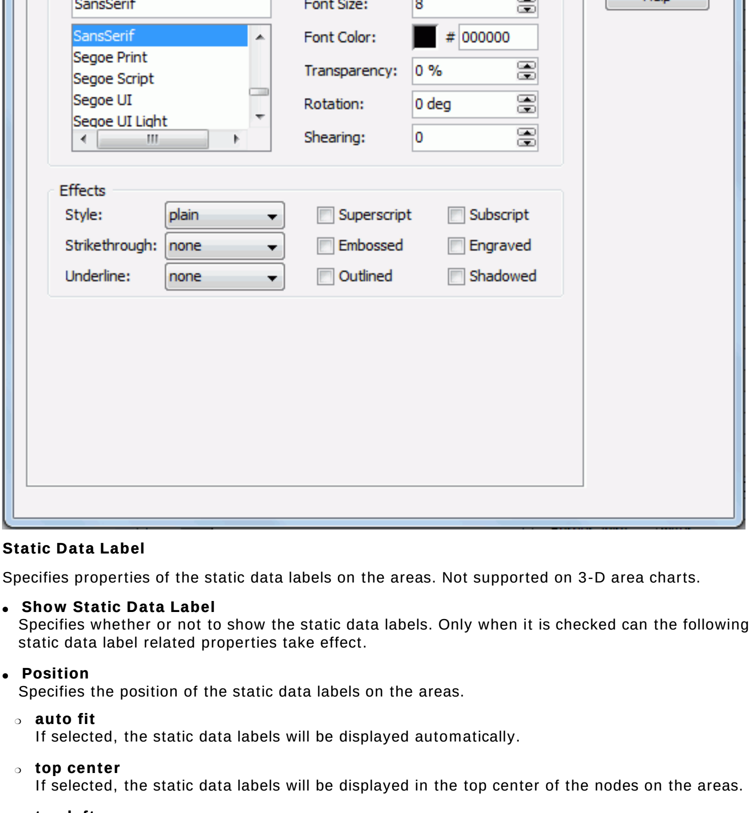
Dash

Specifies the dash size of the border line.

- **Auto Adjusted Dash**
  If selected, the dash size will be adjusted automatically.
- **Fixed Dash Size**
  If selected, the dash size will be fixed size.

Data Label tab

Specifies properties for data labels on areas in the chart.



Static Data Label

Specifies properties of the static data labels on the areas. Not supported on 3-D area charts.

- **Show Static Data Label**
  Specifies whether or not to show the static data labels. Only when it is checked can the following static data label related properties take effect.
- **Position**
  Specifies the position of the static data labels on the areas.
  - **auto fit**
    If selected, the static data labels will be displayed automatically.
  - **top center**
    If selected, the static data labels will be displayed in the top center of the nodes on the areas.
  - **top left**
    If selected, the static data labels will be displayed on the top left of the nodes on the areas.
  - **top right**
    If selected, the static data labels will be displayed on the top right of the nodes on the areas.
  - **bottom left**
    If selected, the static data labels will be displayed on the bottom left of the nodes on the areas.
  - **bottom center**
    If selected, the static data labels will be displayed in the bottom center of the nodes on the areas.
  - **bottom right**
    If selected, the static data labels will be displayed on the bottom right of the nodes on the areas.
- **Suppress Label When 0**
  If true, the static data label whose value is 0 will not be displayed on the chart.

Font

Specifies the font format of text in the data labels.

- **Font list**
  Lists all the available font faces that can be selected to apply to the text.
- **Font Size**
  Specifies the font size of the text.
- **Font Color**
  Specifies the font color of the text.

  To change the color, click the color image button to display the color palette in which you can select a color, or click **More Colors** to customize the color within a wider range in the Pick a Color dialog, or click **More Fill Effects** to customize in the Fill Effects dialog. You can also input the hexadecimal value of a color directly in the text box after #.
- **Transparency**
  Specifies the transparency of the text.
- **Rotation**
  Specifies the rotation angle of the text around its center, in degrees. The default value is 0.
- **Shearing**
  Specifies the gradient of the text.

Effects

Specifies the special effects of text in the data labels.

- **Style**
  Specifies the font style of the text. It can be one of the following: plain, bold, italic, and bold italic.
- **Strikethrough**
  Specifies the style of the horizontal line with which the text is struck through. It can be one of the following: none, thin line, bold line, and double lines.
- **Underline**
  Specifies the style of the horizontal line under the text. It can be one of the following: none, single, single lower, bold line, bold lower, double lines, bold double, patterned line, and bold patterned.
- **Superscript**
  Raises the text above the baseline and changes the text to a smaller font size, if a smaller size is available.
- **Embossed**
  Makes the text appear to be raised off the page in relief.
- **Outlined**
  Displays the inner and outer borders of each character.
- **Subscript**
  Lowers the text below the baseline and changes the text to a smaller font size, if a smaller size is available.
- **Engraved**
  Makes the text appear to be imprinted or pressed into the page.
- **Shadowed**
  Adds a shadow beneath and to the right of the text.

Related topics:

- Formatting the areas

Format Area dialog for web report

This dialog appears when you double-click an area of a chart in a web report, or right-click it and then select Format Area from the shortcut menu. It helps you to format areas in a chart, and consists of the following tabs:

- General tab
- Fill tab
- Border tab
- Data Label tab

OK

Applies the changes and closes the dialog.

Cancel

Does not retain any changes and closes the dialog.

Apply

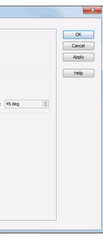
Applies all changes and leaves the dialog open.

Help

Displays the help document about this feature.

General tab

Specifies the general format of the 2-D area chart.



Layout

Specifies the layout for 2-D areas of the chart.

- **None 2-D**
  If selected, 2-D areas will be applied to display the trend of the values over time or categories.
- **Stacked 2-D**
  If selected, stacked 2-D area will be applied to display the trend of the contribution of each data value over categories.
- **100 % Stacked 2-D**
  If selected, 100% stacked 2-D area will be applied to display the trend of the percentage each data value contributes over categories.

Depth

Specifies the depth properties for areas of the chart.

- **Use Depth**
  Specifies whether to make the areas in the chart three-dimensional. You can also use a formula to control the property.
  - **Depth**
    Specifies the depth of the areas, in pixels.
  - **Direction**
    Specifies the direction for depth of the lines, in degrees.

High-Low Lines

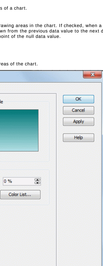
Specifies whether to show the high-low lines in areas of a chart.

Ignore Null Value

Specifies whether to ignore null data values when drawing areas in the chart. If checked, when a null data value appears, it is ignored and the area is drawn from the previous data value to the next data value directly, otherwise, the area is broken at the point of the null data value.

Fill tab

Specifies the color, fill effect and transparency for areas of the chart.



Use Single Color

If checked, the selected area will use the single color pattern.

- **Color**
  Specifies the color scheme for the selected area.

To change the color, click the color image button to display the color palette in which you can select a color, or click **More Colors** to customize the color within a wider range in the Pick a Color dialog, or click **More Fill Effects** to customize in the Fill Effects dialog. You can also input the hexadecimal value of a color directly in the text box after #.

- **Transparency**
  Specifies the transparency of the color scheme.
- **Color List**
  Opens the Color List dialog to modify color pattern for each area.

Use Single Color with Condition

Not supported on area chart.

Use Multiple Colors with Condition

If checked, chart area uses multiple color patterns based on defined conditions. You can divide each area into different parts based on different value ranges, along the direction of the value axis, and then specify different conditional colors to different value ranges for distinguishing.

- **Select Field**
  Lists the value field on which you can apply the conditional fill.
- **Start Value**
  Specifies the start value of the condition.
- **End Value**
  Specifies the end value of the condition.
- **Color**
  Specifies the color that will be applied to the values which meet the condition.

  To change the color, click the color image button to display the color palette in which you can select a color, or click **More Colors** to customize the color within a wider range in the Pick a Color dialog, or click **More Fill Effects** to customize in the Fill Effects dialog. You can also input the hexadecimal value of a color directly in the text box after #.
- **Transparency**
  Specifies the transparency for the color of the condition.

Adds a new condition line.

Removes the selected condition line.

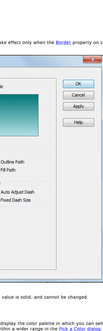
- **Label**
  If checked, you can modify the condition expression of the selected condition line which will be shown as the legend entry label.

Sample

Displays a preview sample of your selection.

Border tab

Specifies properties for border of the areas, which take effect only when the Border property on chart paper is set to true in the Report Inspector.



Border Type

Displays the type for border of the areas. Its default value is solid, and cannot be changed.

Color

Specifies the color for border of the areas.

To change the color, click the color image button to display the color palette in which you can select a color, or click **More Colors** to customize the color within a wider range in the Pick a Color dialog. You can also input the hexadecimal value of a color directly in the text box after #.

Transparency

Specifies the transparency for color of the border.

Line Style

Specifies the line style to apply to border of the areas.

Thickness

Specifies the thickness of the border, in pixels.

End Caps

Specifies the ending style of the border line.

- **butt**
  Ends unclosed sub paths and dash segments with no added decoration.
- **round**
  Ends unclosed sub paths and dash segments with a round decoration that has a radius equal to half of the width of the pen.
- **square**
  Ends unclosed sub paths and dash segments with a square projection that extends beyond the end of the segment to a distance equal to half of the line width.

Line Joint

Specifies the line joint style for the border.

- **miter**
  Joins path segments by extending their outside edges until they meet.
- **round**
  Joins path segments by rounding off the corner at a radius of half the line width.
- **bevel**
  Joins path segments by connecting the outer corners of their wide outlines with a straight segment.

Sample

Displays a preview sample of your selection.

Path

Specifies the fill pattern of the border.

- **Outline Path**
  Specify the fill pattern of the border line to be outline path.
- **Fill Path**
  Specify the fill pattern of the border line to be whole path.

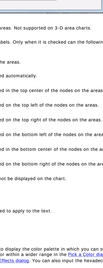
Dash

Specifies the dash size of the border line.

- **Auto Adjusted Dash**
  If checked, the dash size will be adjusted automatically.
- **Fixed Dash Size**
  If selected, the dash size will be fixed size.

Data Label tab

Specifies properties for data labels on areas in the chart.



Static Data Label

Specifies properties of the static data labels on the areas. Not supported on 3-D area charts.

- **Show Static Data Label**
  Specifies whether or not to show the static data labels. Only when it is checked can the following static data label related properties take effect.
- **Position**
  Specifies the position of the static data labels on the areas.
  - **auto fit**
    If selected, the static data labels will be displayed automatically.
  - **top center**
    If selected, the static data labels will be displayed in the top center of the nodes on the areas.
  - **top left**
    If selected, the static data labels will be displayed on the top left of the nodes on the areas.
  - **top right**
    If selected, the static data labels will be displayed on the top right of the nodes on the areas.
  - **bottom left**
    If selected, the static data labels will be displayed on the bottom left of the nodes on the areas.
  - **bottom center**
    If selected, the static data labels will be displayed in the bottom center of the nodes on the areas.
  - **bottom right**
    If selected, the static data labels will be displayed on the bottom right of the nodes on the areas.
- **Suppress Label When 0**
  If true, the static data label whose value is 0 will not be displayed on the chart.

Font

Specifies the font format of text in the data labels.

- **Font list**
  Lists all the available font faces that can be selected to apply to the text.
- **Font Size**
  Specifies the font size of the text.
- **Font Color**
  Specifies the font color of the text.

  To change the color, click the color image button to display the color palette in which you can select a color, or click **More Colors** to customize the color within a wider range in the Pick a Color dialog, or click **More Fill Effects** to customize in the Fill Effects dialog. You can also input the hexadecimal value of a color directly in the text box after #.
- **Transparency**
  Specifies the transparency of the text.
- **Rotation**
  Specifies the rotation angle of the text around its center, in degrees. The default value is 0.
- **Shearing**
  Specifies the gradient of the text.

Effects

Specifies the special effects of text in the data labels.

- **Style**
  Specifies the font style of the text. It can be one of the following: plain, bold, italic, and bold italic.
- **Strikethrough**
  Specifies the style of the horizontal line with which the text is struck through. It can be one of the following: none, thin, thin bold, bold line, and double lines.
- **Underline**
  Specifies the style of the horizontal line under the text. It can be one of the following: none, single, single lower, bold line, bold lower, double lines, bold double, patterned line, and bold patterned.
- **Superscript**
  Raises the text above the baseline and changes the text to a smaller font size. If a smaller size is available.
- **Embossed**
  Makes the text appear to be raised off the page in relief.
- **Outlined**
  Displays the inner and outer borders of each character.
- **Subscript**
  Lowers the text below the baseline and changes the text to a smaller font size. If a smaller size is available.
- **Engraved**
  Makes the text appear to be imprinted or pressed into the page.
- **Shadowed**
  Adds a shadow beneath and to the right of the text.

Related topics:

- Formatting the areas

Format Bar dialog for library component

This dialog appears when you double-click a bar of a chart in a library component, or right-click it and then select Format Bar from the context menu. It helps you to format the bars of a 2-D or a 3-D or 2-D chart, and consists of the following tabs:

- General, and
- Fill tab
- Border tab
- Data Label tab
- Behaviors tab

OK

Applies the changes and closes the dialog.

Cancel

Does not insert any changes and closes the dialog.

Apply

Applies all changes and leaves the dialog open.

Help

Displays the help document about this feature.

General tab

This dialog appears when you double-click a bar of a chart in a library component, or right-click it and then select Format Bar from the context menu. It helps you to format the bars of a 2-D or a 3-D or 2-D chart, and consists of the following tabs:

Style

Specifies the style of the bars in the chart. It can be Normal or Cylinder.

Size

Specifies the size of the bars in the chart.

Bars Width

Specifies bar width as a percentage of the unit width.

Bars Gap

Specifies the distance between two bars as a percentage of the bar width, only applying to clustered bar charts.

Layout

Specifies the layout of the chart.

- **Clustered - 2-D**
Displays and compares data values across categories.
- **Stacked - 2-D**
Displays and compares the contribution of each data value to a total across categories.
- **100% Stacked - 2-D**
Displays and compares the percentage that each data value contributes to a total across categories.

Specifies the depth properties for the chart.

Depth

Specifies whether to make the bars in the chart three-dimensional. You can also use a formula to control the property.

Depth Type

Specifies the depth of the bars, in pixels.

Direction

Specifies the direction for depth of the bars, in degrees.

Fill tab

Specifies the color, fill effect and transparency for bars of the chart.

Use Single Color

If checked, the bars will use the single color pattern.

Color

Specifies the color for the single color pattern.

Use Multiple Colors with Condition

To change the color, click the color image button to display the color palette in which you can select a color, or click **More Colors** to customize the color within a wider range in the Pick a Color dialog, or click **More Fill Effects** to customize in the Fill Effects dialog. You can also input the hexadecimal color value and directly in the text box after it. If you have changed the default color.

Transparency

Specifies the transparency of the color setting.

Vary Colors by Label Line

Specifies whether to vary the colors for bar colors vary. Available only for clustered bar types and when the chart has no series field. If checked, you can specify the color for each bar in the Color List.

Color List

Click the Color List dialog to modify color pattern for bars in the same data series.

Use Single Color with Condition

If checked, each bar can have its single color pattern based on defined conditions. You can click the conditions button to define the conditions and the corresponding color to apply with the conditions.

- **Normal**
Specifies the normal condition.
- **Select Field**
Specifies the available fields to which the conditional fill can be applied. Select the field on which you want to define the conditions from the drop-down list.
- **Start Value**
Specifies the start value of the condition. Available when you select a category or series field or a field whose return value is not a number from the Select Field drop-down list.
- **End Value**
Specifies the end value of the condition. Available when you select a value field or a field that returns a number from the Select Field drop-down list.
- **Color**
Specifies the color for the condition. Available only when you select a value field or a field that returns a number from the Select Field drop-down list.
- **Transparency**
Specifies the transparency for the color of the condition.
- **Delete**
Deletes a new condition line.
- **Add**
Adds a new condition line.
- **Delete All**
Deletes all the selected condition line.
- **Label**
If checked, you can modify the condition expression of the selected condition line which will be shown as the legend entry label.
- **Expression**
Allows to display data in the condition expression as value. Available only when you select a value field or a field that returns a number from the Select Field drop-down list.
- **Percent**
Allows to display data in the condition expression in percent. Available only when you select a value field or a field that returns a number from the Select Field drop-down list and takes effect only on 100% Stacked Bar 2-D chart and 100% Stacked Bar 3-D chart.

Use Multiple Colors with Condition

If checked, each bar can have multiple color patterns based on defined conditions. You can click each color in the Color List to define the conditions and the corresponding color to apply with the conditions, and specify different conditional colors to different value ranges for distinguishing.

- **Select Field**
Specifies the available fields to which the conditional fill can be applied. Select the field on which you want to define the conditions from the drop-down list.
- **Start Value**
Specifies the start value of the condition.
- **End Value**
Specifies the end value of the condition.
- **Color**
Specifies the color that will be applied to the values which meet the condition.
- **Transparency**
Specifies the transparency for the color of the condition.
- **Delete**
Deletes the Edit Conditions dialog to create or edit a condition.
- **Label**
If checked, you can modify the condition expression of the selected condition line which will be shown as the legend entry label.
- **Expression**
Allows to display data in the condition expression as value. Available only when you select a value field or a field that returns a number from the Select Field drop-down list.
- **Percent**
Allows to display data in the condition expression in percent. Available only when you select a value field or a field that returns a number from the Select Field drop-down list and takes effect only on 100% Stacked Bar 2-D chart and 100% Stacked Bar 3-D chart.

Transparency

Specifies the transparency for the color of the condition.

Color

Specifies the color that will be applied to the values which meet the condition.

To change the color, click the color image button to display the color palette in which you can select a color, or click **More Colors** to customize the color within a wider range in the Pick a Color dialog, or click **More Fill Effects** to customize in the Fill Effects dialog. You can also input the hexadecimal color value and directly in the text box after it.

Transparency

Specifies the transparency for the color of the condition.

Delete

Deletes the Edit Conditions dialog to create or edit a condition.

Label

If checked, you can modify the condition expression of the selected condition line which will be shown as the legend entry label.

Expression

Allows to display data in the condition expression as value. Available only when you select a value field or a field that returns a number from the Select Field drop-down list.

Percent

Allows to display data in the condition expression in percent. Available only when you select a value field or a field that returns a number from the Select Field drop-down list and takes effect only on 100% Stacked Bar 2-D chart and 100% Stacked Bar 3-D chart.

Border tab

Specifies the properties for border of the bars, which take effect only when the Border property on chart paper is set to true in the Report Inspector.

Border Type

Specifies the type for border of the bars. The default value is solid, and cannot be changed.

Color

Specifies the color for border of the bars.

To change the color, click the color image button to display the color palette in which you can select a color, or click **More Colors** to customize the color within a wider range in the Pick a Color dialog. You can also input the hexadecimal color value and directly in the text box after it.

Transparency

Specifies the transparency for color of the bars.

Line Style

Specifies the line style to apply to border of the bars.

Thickness

Specifies the thickness of the border, in pixels.

End Caps

Specifies the ending style of the border line.

- **None**
Sets unclosed sub paths and dash segments with no added decoration.
- **Round**
Ends unclosed sub paths and dash segments with a round decoration that has a radius equal to half of the pen width.
- **Square**
Ends unclosed sub paths and dash segments with a square projection that extends beyond the end of the segment to a distance equal to half of the pen width.

Line Joint

Specifies how the joint style for the border line

- **Miter**
Joins path segments by extending their outside edges until they meet.
- **Round**
Joins path segments by rounding off the corner at a radius of half the line width.
- **Bevel**
Joins path segments by connecting the outer corners of their wide outlines with a straight segment.

Sample

Displays a sample pattern of your selection.

Path

Specifies the fill pattern of the border line.

- **Outline Path**
Specifies the fill pattern of the border line to be outline path.
- **Fill Path**
Specifies the fill pattern of the border line to be whole path.

Dash

Specifies the dash style of the border line.

- **Auto Adjusted Dash**
Specifies the dash pattern will be adjusted automatically.
- **Fixed Dash Size**
If selected, the dash size will be fixed size.

Data Label tab

Specifies properties for data labels displayed on the bars.

Static Data Label

Specifies properties of the static data labels on the bars, first supported on 3-D bar charts.

- **Show Static Data Label**
Specifies whether or not to show the static data labels on the bars. Only when it is checked can the other fields of static data label related properties take effect.

- **Position**
Specifies the position of the static data labels on the bars.
- **auto**
In auto mode, the static data labels will be displayed automatically.
- **outside top**
If selected, the static data labels will be displayed on the outside top of the bars.
- **inside top**
If selected, the static data labels will be displayed on the inside top of the bars.
- **inside center**
If selected, the static data labels will be displayed at the inside center of the bars.
- **inside bottom**
If selected, the static data labels will be displayed at the inside bottom of the bars.

- **Auto Arrange**
Specifies the auto arrange of the static data labels inside the bars at the bar position. Available only when the static data label's position is set to one of the following: inside center, inside top and inside bottom.

The static data label will be displayed horizontally at the specified position if the bar has enough room horizontally; otherwise, it will be displayed vertically. If a bar does not have enough room horizontally or vertically, its static data label will not be displayed.

- **Text**
The static data labels will be displayed at the specified position, and if the labels get overlapping, some of them will not be displayed.

Font

Specifies the font format of text in the data labels.

- **Color**
Sets all the available font fields that can be selected to apply to the text.
- **Font Size**
Specifies the font size of the text.

Text Format

Specifies the text color of the text.

To change the color, click the color image button to display the color palette in which you can select a color, or click **More Colors** to customize the color within a wider range in the Pick a Color dialog, or click **More Fill Effects** to customize in the Fill Effects dialog. You can also input the hexadecimal color value and directly in the text box after it.

Transparency

Specifies the transparency of the text.

Rotation

Specifies the rotation angle of the text around its center, in degrees. The default value is 0.

- **Shearing**
Specifies the shearing of the text.

Effects

Specifies the special effects of text in the data labels.

- **Shadow**
If selected, the text is shadowed.
- **Outline**
Specifies the size of the horizontal line under the text. It can be one of the following: none, single, double, thick, dotted, dashed, long dashed, dash dot, dash dot dot, double dashed, patterned line, and bold patterned line.
- **Strike**
Places the text above the baseline and changes the text to a smaller font size, if a smaller size is available.
- **Embossed**
Makes the text appear to be raised off the page in relief.
- **Subscript**
Displays the inner and outer borders of each character.
- **Superscript**
Causes the text below the baseline and changes the text to a smaller font size, if a smaller size is available.
- **Engraved**
Makes the text appear to be imprinted or pressed into the page.
- **Shadowed**
Adds a shadow beneath and up to the right of the text.

Behaviors tab

Specifies some web behaviors to the bars of the chart.

Events

Lists all the available events that can be selected to apply to the text.

Actions

Lists the Web Action List dialog to bind a web action to the event.

Related topics:

- Formatting the bars

Format Bar dialog for page report

This dialog appears when you double-click a bar of a chart in a page report, or right-click a bar and then select Format Bar from the shortcut menu. It helps you to format the bars of a 2-D or a 3-D bar chart and contains of the following tabs:

- General tab
- Fill tab
- Border tab
- Data Label tab

OK
Applies the changes and closes the dialog.

Cancel
Does not retain any changes and closes the dialog.

Apply
Applies all changes and leaves the dialog open.

Help
Displays the help document about this feature.

General tab

For 2-D bar charts

Specifies the general format of the 2-D bar chart.

For 3-D bar charts

Specifies the general format of the 3-D bar chart.

Style
Specifies the style of the bars in the chart. It can be Normal or Cylinder.

Size
Specifies the size of the bars in the chart.

Bars Width
Specifies the bar width as a percentage of the unit width.

Bars Gap
Specifies the distance between two bars as a percentage of the bar width, only applying to clustered layout.

Layout
Specifies the layout of the chart.

- **Clustered - 2-D**
  Specifies to display clustered bar values across categories.
- **Stacked - 2-D**
  Specifies to display the contribution of each data value to a total across categories.
- **100% Stacked - 2-D**
  Displays each contribution as a percentage that each data value contributes to a total across categories.

Depth
Specifies the depth properties for bars of the chart.

Depth:
Specifies whether to make the bars in the chart three-dimensional. You can also use a formula to control the property.

Depth
Specifies the depth of the bars, in points.

Direction
Specifies the direction for depth of the bars, in degrees.

For 3-D bar charts

Specifies the general format of the 3-D bar chart.

Size
Specifies the size of bars in the chart.

Bars Width
Specifies the bar width as a percentage of the unit width.

Layout
Specifies the layout of the chart.

- **Clustered Bar 3-D**
  Specifies to display clustered bar with a 3-D visual effect to the chart.
- **Stacked Bar 3-D**
  Specifies to display stacked bar with a 3-D visual effect to the chart.
- **100% Stacked Bar 3-D**
  Specifies to display 100% Stacked bar with a 3-D visual effect to the chart.
- **Area 3-D**
  Specifies to apply area bar with a 3-D visual effect to the chart.

Fill tab

Specifies the color, fill effect and transparency for bars of the chart.

Use Single Color
If checked, the bars will use the single color pattern.

Color
Specifies the color scheme for the bars.

To change the color, click the color image button to display the color palette in which you can select a color. Click **More Colors** to customize the color within a wider range in the Pick a Color dialog, or click **More Fill Effects** to customize in the Fill Effects dialog. You can also input the hexadecimal value of a color directly in the text box after it if you have changed the default color.

Transparency
Specifies the transparency of the color scheme.

Turn a Colors Into Color List
Specifies whether or not to make the bar colors vary. Available only for clustered bar types and when **Use Single Color** is not checked. If checked, you can specify the color for each bar in the Color List dialog.

Color List
Displays the Color List dialog to modify color pattern for bars in the same data series.

Use Single Color Condition
If checked, each bar can have its single color pattern based on defined conditions. You can click the Condition button to select the condition to be not-sliding mode to edit the condition.

- **Normal**
  Specifies to display data in normal mode.
- **Condition**
  Specifies to display data in condition mode. Available only when you select a category or series field or a field whose input value is not a number from the Select Field drop down list.
- **Start Value**
  Specifies the start value of the condition. Available only when you select a value field or a field that can provide numbers from the drop down list.
- **End Value**
  Specifies the end value of the condition. Available only when you select a value field or a field that can provide numbers from the drop down list.
- **Transparency**
  Specifies the transparency for the color of the condition.
- **Color**
  Specifies the color for the condition.

To change the color, click the color image button to display the color palette in which you can select a color. Click **More Colors** to customize the color within a wider range in the Pick a Color dialog, or click **More Fill Effects** to customize in the Fill Effects dialog. You can also input the hexadecimal value of a color directly in the text box after it.

- **Transparency**
  Specifies the transparency for the color of the condition.
- **Color**
  Specifies a new condition line.
- **Delete**
  Deletes the selected condition line.
- **Value**
  Specifies the value for the condition. Available only when the selected condition line which will be shown as the legend entry label.

Use Multiple Colors with Condition
If checked, each bar can have multiple color patterns based on defined conditions. You can click each Edit button to select the condition to be not-sliding mode to edit the condition. Click the Value edit, and then select a condition color in different value ranges for distinguishing.

- **Select Field**
  Specifies to display fields to which the conditional fit can be applied. Select the field on which you want to define the conditions from the drop down list.
- **Start Value**
  Specifies the start value of the condition.
- **End Value**
  Specifies the end value of the condition.
- **Color**
  Specifies the color that will be applied to the values which meet the condition.

To change the color, click the color image button to display the color palette in which you can select a color. Click **More Colors** to customize the color within a wider range in the Pick a Color dialog, or click **More Fill Effects** to customize in the Fill Effects dialog. You can also input the hexadecimal value of a color directly in the text box after it.

- **Transparency**
  Specifies the transparency for the color of the condition.
- **Color**
  Specifies a new condition line.
- **Delete**
  Deletes the selected condition line.
- **Value**
  Specifies the value for the condition. Available only when the selected condition line which will be shown as the legend entry label.

Sample
Displays a preview sample of your selection.

Border tab

Specifies the properties for border of the bars, which take effect only when the Border property on chart object is set to true in the Report Designer.

Border Type
Displays the type for border of the bars. The default value is solid, and cannot be changed.

Color
Specifies the color for border of the bars.

To change the color, click the color image button to display the color palette in which you can select a color. Click **More Colors** to customize the color within a wider range in the Pick a Color dialog. You can also input the hexadecimal value of a color directly in the text box after it.

Transparency
Specifies the transparency for color of the border.

Line Style
Specifies the line style to apply to border of the bars.

Thickness
Specifies the thickness of the border, in points.

End Caps
Specifies the ending style of the border line.

- **Bevel**
  Bevels unclosed sub paths and dash segments with no added decoration.
- **round**
  Rounds unclosed sub paths and dash segments with a round decoration that has a radius equal to half of the width of the pen.
- **square**
  Extends unclosed sub paths and dash segments with a square projection that extends beyond the end of the segment to a distance equal to half of the line width.

Line Joint
Specifies the line joint style for the border line.

- **miter**
  Joins path segments by extending their outside edges until they meet.
- **round**
  Joins path segments by rounding off the corner at a radius of half the line width.
- **bevel**
  Joins path segments by connecting the outer corners of their wide outlines with a straight segment.

Sample
Displays a preview sample of your selection.

Path
Specifies the fill pattern of the border line.

- **Outline Path**
  Specifies the fill pattern of the border line to be outline path.
- **Fill Path**
  Specifies the fill pattern of the border line to be whole path.

Depth
Specifies the depth size of the border line.

- **Auto Adjusted Depth**
  If selected, the depth size will be adjusted automatically.
- **Fixed Depth Size**
  If selected, the depth size will be fixed.

Data Label tab

Specifies properties for data labels displayed on the bars.

Static Data Label
Specifies properties of the static data labels on the bars. Not supported on 3-D bar charts.

- **Show Static Data Label**
  If checked, the static data labels will be shown on the bars. Only when it is checked can the following static data label related properties take effect.
- **Position**
  Specifies the position of the static data labels on the bars.
  - **auto fit**
    If selected, the static data labels will be displayed automatically.
  - **outside**
    If selected, the static data labels will be displayed on the outside top of the bars.
  - **inside**
    If selected, the static data labels will be displayed on the inside top of the bars.
  - **inside center**
    If selected, the static data labels will be displayed at the inside center of the bars.
  - **inside bottom**
    If selected, the static data labels will be displayed at the inside bottom of the bars.
- **Suppress Label When 0**
  If true, the static data label whose value is 0 will not be displayed on the chart.
- **Auto Arrange**
  Specifies whether to display the static data labels inside the bars at the best position. Available only when the static data label position is set to **auto fit**. If the static data label position is set to **inside center**, **inside** and **inside bottom**, the value is ignored.
  The static data label will be displayed horizontally at the specified position if the bar has enough space to display the static data label. Otherwise, the static data label will be displayed at the specified position vertically and horizontally. So static data label will be displayed.
- **Text**
  The static data labels will be displayed at the specified position, and if the labels get overlapping, some of them will not be displayed.

Series Data Label
Specifies properties of the series data labels on the bars. Not supported on 3-D bar charts.

- **Show Series Data Label**
  If checked, the series data labels will be shown on the bars. Only when it is checked can the following series data label related properties take effect.
- **Position**
  Specifies the position of the series data labels on the bars.
  - **auto fit**
    If selected, the series data labels will be displayed automatically.
  - **outside**
    If selected, the series data labels will be displayed on the outside top of the bars.
  - **inside**
    If selected, the series data labels will be displayed on the inside top of the bars.
  - **inside center**
    If selected, the series data labels will be displayed at the inside center of the bars.
  - **inside bottom**
    If selected, the series data labels will be displayed at the inside bottom of the bars.
- **Suppress Label When 0**
  If true, the series data label whose value is 0 will not be displayed on the chart.
- **Auto Arrange**
  Specifies whether to display the series data labels inside the bars at the best position. Available only when the series data label position is set to **auto fit**. If the series data label position is set to **inside center**, **inside** and **inside bottom**, the value is ignored.
  The series data label will be displayed horizontally at the specified position if the bar has enough space to display the series data label. Otherwise, the series data label will be displayed at the specified position vertically and horizontally. So series data label will be displayed.
- **Transparency**
  Specifies the transparency for the text.
- **Rotation**
  Specifies the rotation angle of the text around its center, in degrees. The default value is 0.

Drawing
Specifies the grabbed of the text.

Effects
Specifies the special effects of text in the data labels.

- **Style**
  Specifies the font style of the text. It can be one of the following: plain, bold, italic, and bold italic.
- **Strikethrough**
  Specifies the type of the horizontal line with which the text is struck through. It can be one of the following: none, thin line, medium line, and double line.
- **Underline**
  Specifies the style of the horizontal line under the text. It can be one of the following: none, single line, bold line, bold lower, double line, bold double, patterned line, and bold patterned.
- **Superscript**
  If true, the text above the baseline and changes the text to a smaller font size, if a smaller size is available.
- **Embossed**
  Makes the text appear to be raised off the page in relief.
- **Outline**
  Specifies a thin inner and outer borders of each character.
- **Subscript**
  Lowers the text below the baseline and changes the text to a smaller font size, if a smaller size is available.
- **Engraved**
  Makes the text appear to be imprinted or pressed into the page.
- **Shadowed**
  Adds a shadow beneath and to the right of the text.

Related topics:
- Formatting the bars

Format Bar dialog for web report

This dialog appears when you double-click a bar of a chart in a web report, or right-click it and then select Format Bar from the shortcut menu. It helps you to format the bars of a 2-D or a 3-D bar chart, and consists of the following tabs:

- General tab
- Fill tab
- Border tab
- Data Label tab

OK

Closes the dialog and saves the changes.

Cancel

Does not remain any changes and closes the dialog.

Apply

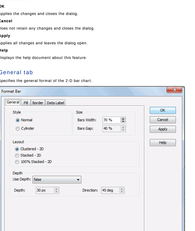
Applies all changes and leaves the dialog open.

Help

Displays the help document about this feature.

General tab

Specifies the general format of the 2-D bar chart.



Style

Specifies the style of the bars in the chart. It can be Normal or Cylinder.

Size

Specifies the size of the bars in the chart.

- **Bars Width**
  Specifies the width as a percentage of the unit width.
- **Bars Gap**
  Specifies the distance between two bars as a percentage of the bar width, only applying to clustered bar charts.

Layout

Specifies the layout of the chart.

- **Clustered - 2-D**
  Displays and compares data values across categories.
- **Stacked - 2-D**
  Displays and compares the contribution of each data value to a total across categories.
- **100% Stacked - 2-D**
  Displays and compares the percentage that each data value contributes to a total across categories.

Depth

Specifies the depth properties for the bars of a chart.

- **Use Depth**
  Specifies whether to make the bars in the chart three-dimensional. You can also use a formula to control the property.
- **Depth**
  Specifies the depth of the bars, in pixels.
- **Direction**
  Specifies the direction for depth of the bars, in degrees.

Fill tab

Specifies the color, fill effect and transparency for bars of the chart.

Use Single Color

If checked, the chart will use the single color pattern.

- **Color**
  Specifies the color for the bars.
  To change the color, click the color image button to display the color palette in which you can select a color, or click **More Colors** to customize the color within a wider range in the Pick a Color dialog, or click **More Fill Effects** to customize in the Fill Effects dialog. You can also input the hexadecimal value of a color directly in the text box after # if you have changed the default color.
- **Transparency**
  Specifies the transparency of the color schema.

Vary Colors by Color List

Specifies whether or not to make the bar colors vary. Available only for clustered bar types and when the Use Single Color is unchecked. If checked, you can specify the color for each bar in the Color List dialog.

- **Color List**
  Opens the Color List dialog to modify color patterns for bars in the same data series.

Use Single Color with Condition

If checked, the chart will use the single color pattern based on defined conditions. You can click the Advanced or Normal button to switch between the two editing modes to edit the conditions.

- **Normal**
  - **Select Field**
    Specifies the attribute/metric fields to which the conditional fill can be applied. Select the field on which you want to define the conditions from the drop-down list.
  - **Values**
    Specifies the value of the condition. Available when you select a category or series field or a field whose return value is a number from the Select Field drop-down list.
  - **Start Value**
    Specifies the start value of the condition. Available only when you select a value field or a field that returns a number from the Select Field drop-down list.
  - **End Value**
    Specifies the end value of the condition. Available only when you select a value field or a field that returns a number from the Select Field drop-down list and takes effect only on 100% Stacked Bar 2-D chart and 100% Stacked Bar 3-D chart.
  - **Transparency**
    Specifies the transparency for the color of the condition.
  - **Color**
    Specifies a new condition line.
  - **Delete**
    Removes the selected condition line.
- **Advanced**
  - **Label**
    If checked, you can modify the condition expression of the selected condition line which will be shown as the legend entry.
  - **Value**
    Specifies to display data in the condition expression as value. Available only when you select a value field or a field that returns a number from the Select Field drop-down list.
  - **Percent**
    Specifies to display data in the condition expression in percent. Available only when you select a value field or a field that returns a number from the Select Field drop-down list and takes effect only on 100% Stacked Bar 2-D chart and 100% Stacked Bar 3-D chart.
  - **Transparency**
    Specifies the transparency for the color of the condition.
  - **Color**
    Opens the Edit Conditions dialog to create or edit a condition.
  - **Delete**
    Removes the selected condition line.

Use Multiple Colors with Condition

If checked, each bar can have multiple color patterns based on defined conditions. You can divide each bar into different parts based on different value ranges, along the direction of the value axis, and then specify different conditional colors to different value ranges for distinguishing.

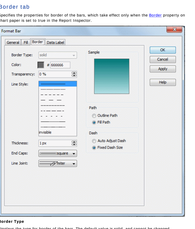
- **Select Field**
  Lists all the attributes/metrics to which the conditional fill can be applied. Select the field on which you want to define the conditions from the drop-down list.
- **Start Value**
  Specifies the start value of the condition.
- **End Value**
  Specifies the end value of the condition.
- **Color**
  Specifies the color that will be applied to the values which meet the condition.
  To change the color, click the color image button to display the color palette in which you can select a color, or click **More Colors** to customize the color within a wider range in the Pick a Color dialog, or click **More Fill Effects** to customize in the Fill Effects dialog. You can also input the hexadecimal value of a color directly in the text box after #.
- **Transparency**
  Specifies the transparency for the color of the condition.
- **Delete**
  Removes a new condition line.
- **Clear**
  Removes the selected condition line.
- **Label**
  If checked, you can modify the condition expression of the selected condition line which will be shown as the legend entry label.

Sample

Displays a preview sample of your selection.

Border tab

Specifies the properties for border of the bars, which take effect only when the Border property on chart paper is set to true in the Report Inspector.



Border Type

Displays the type for border of the bars. The default value is solid, and cannot be changed.

Color

Specifies the color for border of the bars.

To change the color, click the color image button to display the color palette in which you can select a color, or click **More Colors** to customize the color within a wider range in the Pick a Color dialog. You can also input the hexadecimal value of a color directly in the text box after #.

Transparency

Specifies the transparency for color of the border.

Line Style

Specifies the line style to apply to border of the bars.

Thickness

Specifies the thickness of the border, in pixels.

End Caps

Specifies the ending style of the border line.

- **butt**
  Ends enclosed sub paths and dash segments with no added decoration.
- **round**
  Ends enclosed sub paths and dash segments with a round decoration that has a radius equal to half of the width of the pen.
- **square**
  Ends enclosed sub paths and dash segments with a square projection that extends beyond the end of the segment to a distance equal to half of the line width.

Line Joint

Specifies the line joint style for the border.

- **miter**
  Joins path segments by extending their outside edges until they meet.
- **bevel**
  Joins path segments by rounding off the corner at a radius of half the line width.
- **round**
  Joins path segments by connecting the outer corners of their wide outlines with a straight segment.

Sample

Displays a preview sample of your selection.

Path

Specifies the fill pattern of the bar.

- **Outline Path**
  Specifies the fill pattern of the border line to be outline path.
- **Fill Path**
  Specifies the fill pattern of the border line to be whole path.

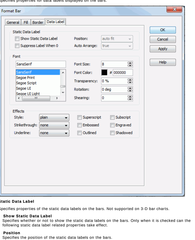
Dash

Specifies the dash size of the border line.

- **Auto Adjusted Dash**
  If selected, the dash size will be adjusted automatically.
- **Fixed Dash Size**
  If selected, the dash size will be fixed.

Data Label tab

Specifies properties for data labels displayed on the bars.



Static Data Label

Specifies properties of the static data labels on the bars. Not supported in 3-D bar charts.

- **Show Static Data Label**
  Specifies whether to show the static data labels on the bars. Only when it is checked can the following static data label related properties take effect.
- **Position**
  Specifies the position of the static data labels on the bars.
  - **auto fit**
    If selected, the static data labels will be displayed automatically.
  - **outside top**
    If selected, the static data labels will be displayed on the outside top of the bars.
  - **inside top**
    If selected, the static data labels will be displayed on the inside top of the bars.
  - **inside center**
    If selected, the static data labels will be displayed at the inside center of the bars.
  - **inside bottom**
    If selected, the static data labels will be displayed at the inside bottom of the bars.
- **Suppress Label When 0**
  If selected, the static data label whose value is 0 will not be displayed on the chart.
- **Auto Arrange**
  Specifies whether to display the static data labels inside the bars at the best position. Available only when the static data label position is set to one of the following: inside center, inside top and inside bottom.
  - **true**
    The static data label will be displayed horizontally at the specified position if the bar has enough room horizontally; otherwise, it will be displayed vertically. If a bar does not have enough room horizontally and horizontally, its static data label will not be displayed.
  - **false**
    The static data labels will be displayed at the specified position, and if the labels get overlapping, some of them will not be displayed.

Font

Specifies the font format of text in the data labels.

- **Font list**
  Lists all the available fonts that can be selected to apply to the text.
- **Font Size**
  Specifies the font size of the text.
- **Font Color**
  Specifies the font color of the text.
  To change the color, click the color image button to display the color palette in which you can select a color, or click **More Colors** to customize the color within a wider range in the Pick a Color dialog.
- **Transparency**
  Specifies the transparency of the text.
- **Rotation**
  Specifies the rotation angle of the text around its center, in degrees. The default value is 0.
- **Shearing**
  Specifies the gradient of the text.

Effects

Specifies the special effects of text in the data labels.

- **Style**
  Specifies the font style of the text. It can be one of the following: plain, bold, italic, and bold italic.
- **Strikethrough**
  Specifies the style of the horizontal line with which the text is struck through. It can be one of the following: none, thin line, bold line, and double lines.
- **Underline**
  Specifies the style of the horizontal line under the text. It can be one of the following: none, single, single word, bold line, bold word, double lines, bold double, patterned line, and bold patterned.
- **Superscript**
  Raises the text above the baseline and changes the text to a smaller font size, if a smaller size is available.
- **Subscript**
  Lowers the text below the baseline and changes the text to a smaller font size, if a smaller size is available.
- **Embossed**
  Raises the text appear to be raised off the page in relief.
- **Outlined**
  Displays the inner and outer borders of each character.
- **Engraved**
  Makes the text appear to be impressed or pressed into the page.
- **Shadowed**
  Adds a shadow beneath and to the right of the text.

Related topics:

- Formatting the bars

Format Bar Gauge dialog for library component

This dialog appears when you double-click a bar gauge of a chart in a library component, or right-click it and select Format Bar Gauge from the shortcut menu. It helps you to format bar gauges in a chart, and consists of the following tabs:

- General tab
- Fill tab
- Border tab
- Data Label tab
- Behaviors tab

OK

Applies the changes and closes the dialog.

Cancel

Does not retain any changes and closes the dialog.

Apply

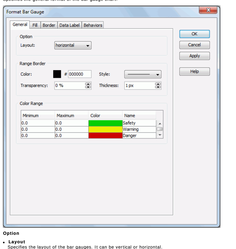
Applies all changes and leaves the dialog open.

Help

Displays the help document about this feature.

General tab

Specifies the general format of the bar gauge chart.



Option

- **Layout**
  Specifies the layout of the bar gauges. It can be vertical or horizontal.

Range Border

Specifies properties of the range borders.

- **Color**
  Specifies the color schema of the range borders.
  To change the color, click the color image button to display the color palette in which you can select a color, or click **More Colors** to customize the color within a wider range in the Pick a Color dialog. You can also input the hexadecimal value of a color directly in the text box after #.
- **Style**
  Specifies the style of the range borders.
- **Transparency**
  Specifies the transparency of the color schema.
- **Thickness**
  Specifies the thickness of the range borders, in pixels.

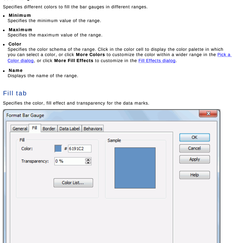
Color Range

Specifies different colors to fill the bar gauges in different ranges.

- **Minimum**
  Specifies the minimum value of the range.
- **Maximum**
  Specifies the maximum value of the range.
- **Color**
  Specifies the color schema of the range. Click in the color cell to display the color palette in which you can select a color, or click **More Colors** to customize the color within a wider range in the Pick a Color dialog, or click **More Fill Effects** to customize in the Fill Effects dialog.
- **Name**
  Displays the name of the range.

Fill tab

Specifies the color, fill effect and transparency for the data marks.



Color

Specifies the color schema for the selected data marks in the same data series.

To change the color, click the color image button to display the color palette in which you can select a color, or click **More Colors** to customize the color within a wider range in the Pick a Color dialog, or click **More Fill Effects** to customize in the Fill Effects dialog. You can also input the hexadecimal value of a color directly in the text box after # if you have changed the default color

Transparency

Specifies the transparency of the color schema.

Color List

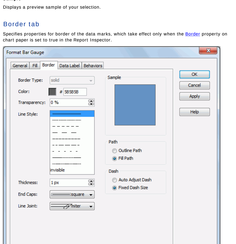
Opens the Color List dialog to modify color pattern for data marks in the same data series.

Sample

Displays a preview sample of your selection.

Border tab

Specifies properties for border of the data marks, which take effect only when the Border property on chart paper is set to true in the Report Inspector.



Border Type

Displays the type of the border. The default value is solid, and cannot be changed.

Color

Specifies the color of the border.

To change the color, click the color image button to display the color palette in which you can select a color, or click **More Colors** to customize the color within a wider range in the Pick a Color dialog. You can also input the hexadecimal value of a color directly in the text box after #

Transparency

Specifies the transparency for color of the border.

Line Style

Specifies the line style of the border.

Thickness

Specifies the thickness of the border, in pixels.

End Caps

Specifies the ending style of the border line.

- **butt**
  Ends unclosed sub paths and dash segments with no added decoration.
- **round**
  Ends unclosed sub paths and dash segments with a round decoration that has a radius equal to half of the width of the pen.
- **square**
  Ends unclosed sub paths and dash segments with a square projection that extends beyond the end of the segment to a distance equal to half of the line width.

Line Joint

Specifies the line joint style of the border line.

- **miter**
  Joins path segments by extending their outside edges until they meet.
- **round**
  Joins path segments by rounding off the corner at a radius of half the line width.
- **bevel**
  Joins path segments by connecting the outer corners of their wide outlines with a straight segment.

Sample

Displays a preview sample of your selection.

Path

Specifies the fill pattern of the border line.

- **Outline Path**
  Specifies the fill pattern of the border line to be outline path.
- **Fill Path**
  Specifies the fill pattern of the border line to be whole path.

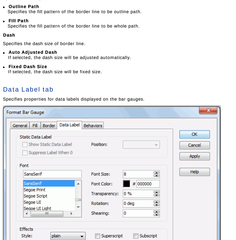
Dash

Specifies the dash size of border line.

- **Auto Adjusted Dash**
  If selected, the dash size will be adjusted automatically.
- **Fixed Dash Size**
  If selected, the dash size will be fixed size.

Data Label tab

Specifies properties for data labels displayed on the bar gauges.



Static Data Label

Specifies properties of the static data labels. Not supported on gauge chart.

Font

Specifies the font format of text in the data labels.

- **Font list**
  Lists all the available font faces that can be selected to apply to the text.
- **Font Size**
  Specifies the font size of the text.
- **Font Color**
  Specifies the font color of the text.
  To change the color, click the color image button to display the color palette in which you can select a color, or click **More Colors** to customize the color within a wider range in the Pick a Color dialog, or click **More Fill Effects** to customize in the Fill Effects dialog. You can also input the hexadecimal value of a color directly in the text box after #.
- **Transparency**
  Specifies the transparency of the text.
- **Rotation**
  Specifies the rotation angle of the text around its center, in degrees. The default value is 0.
- **Shearing**
  Specifies the gradient of the text.

Effects

Specifies the special effects of text in the data labels.

- **Style**
  Specifies the font style of the text. It can be one of the following: plain, bold, italic, and bold italic.
- **Strikethrough**
  Specifies the style of the horizontal line with which the text is struck through. It can be one of the following: none, thin line, bold line, and double lines.
- **Underline**
  Specifies the style of the horizontal line under the text. It can be one of the following: none, single, single lower, bold line, bold lower, double lines, bold double, patterned line, and bold patterned.
- **Superscript**
  Raises the text above the baseline and changes the text to a smaller font size, if a smaller size is available.
- **Embossed**
  Makes the text appear to be raised off the page in relief.
- **Outlined**
  Displays the inner and outer borders of each character.
- **Subscript**
  Lowers the text below the baseline and changes the text to a smaller font size, if a smaller size is available.
- **Engraved**
  Makes the text appear to be imprinted or pressed into the page.
- **Shadowed**
  Adds a shadow beneath and to the right of the text.

Behaviors tab

Specifies some web behaviors to the bar gauge.

Adds a new web behavior line.

Removes the selected web behavior.

Moves the selected web behavior up a step.

Moves the selected web behavior down a step.

Events

Specifies the trigger event.

Actions

Specifies the action you want the event to trigger.

Opens the Web Action List dialog to bind a web action to the event.

Related topics :

- Formatting the gauge

Format Bar Gauge dialog for report

This dialog appears when you double-click a bar gauge of a chart in a page report or web report, or right-click it and select Format Bar Gauge from the shortcut menu. It helps you to format bar gauges in a chart, and consists of the following tabs:

- General tab
- Fill tab
- Border tab
- Data Label tab

OK
Applies the changes and closes the dialog.

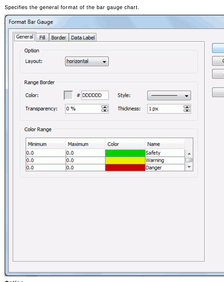
Cancel
Does not retain any changes and closes the dialog.

Apply
Applies all changes and leaves the dialog open.

Help
Displays the help document about this feature.

General tab

Specifies the general format of the bar gauge chart.



Option

- **Layout**
  Specifies the layout of the bar gauges. It can be vertical or horizontal.

Range Border

Specifies properties of the range borders.

- **Color**
  Specifies the color schema of the range borders.

  To change the color, click the color image button to display the color palette in which you can select a color, or click **More Colors** to customize the color within a wider range in the Pick a Color dialog. You can also input the hexadecimal value of a color directly in the text box after #.

- **Style**
  Specifies the style of the range borders.
- **Transparency**
  Specifies the transparency of the color schema.
- **Thickness**
  Specifies the thickness of the range borders, in pixels.

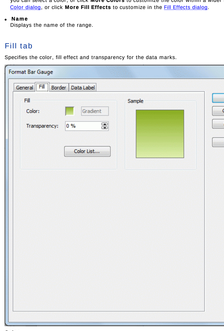
Color Range

Specifies different colors to fill the bar gauges in different ranges.

- **Minimum**
  Specifies the minimum value of the range.
- **Maximum**
  Specifies the maximum value of the range.
- **Color**
  Specifies the color schema of the range. Click in the color cell to display the color palette in which you can select a color, or click **More Colors** to customize the color within a wider range in the Pick a Color dialog, or click **More Fill Effects** to customize in the Fill Effects dialog.
- **Name**
  Displays the name of the range.

Fill tab

Specifies the color, fill effect and transparency for the data marks.



Color

Specifies the color schema for the selected data marks in the same data series.

To change the color, click the color image button to display the color palette in which you can select a color, or click **More Colors** to customize the color within a wider range in the Pick a Color dialog, or click **More Fill Effects** to customize in the Fill Effects dialog. You can also input the hexadecimal value of a color directly in the text box after # if you have changed the default color.

Transparency

Specifies the transparency of the color schema.

Color List

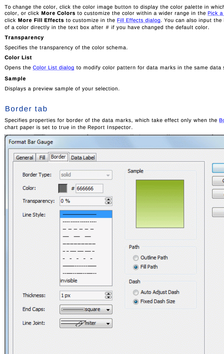
Opens the Color List dialog to modify color pattern for data marks in the same data series.

Sample

Displays a preview sample of your selection.

Border tab

Specifies properties for border of the data marks, which take effect only when the Border property on chart paper is set to true in the Report Inspector.



Border Type

Displays the type of the border. The default value is solid, and cannot be changed.

Color

Specifies the color of the border.

To change the color, click the color image button to display the color palette in which you can select a color, or click **More Colors** to customize the color within a wider range in the Pick a Color dialog. You can also input the hexadecimal value of a color directly in the text box after #

Transparency

Specifies the transparency for color of the border.

Line Style

Specifies the line style of the border.

Thickness

Specifies the thickness of the border, in pixels.

End Caps

Specifies the ending style of the border line.

- **butt**
  Ends unclosed sub paths and dash segments with no added decoration.
- **round**
  Ends unclosed sub paths and dash segments with a round decoration that has a radius equal to half of the width of the pen.
- **square**
  Ends unclosed sub paths and dash segments with a square projection that extends beyond the end of the segment to a distance equal to half of the line width.

Line Joint

Specifies the line joint style of the border line.

- **miter**
  Joins path segments by extending their outside edges until they meet.
- **round**
  Joins path segments by rounding off the corner at a radius of half the line width.
- **bevel**
  Joins path segments by connecting the outer corners of their wide outlines with a straight segment.

Sample

Displays a preview sample of your selection.

Path

Specifies the fill pattern of the border line.

- **Outline Path**
  Specifies the fill pattern of the border line to be outline path.
- **Fill Path**
  Specifies the fill pattern of the border line to be whole path.

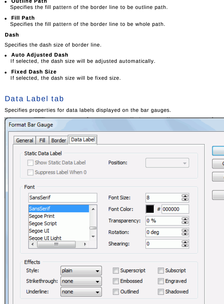
Dash

Specifies the dash size of border line.

- **Auto Adjusted Dash**
  If selected, the dash size will be adjusted automatically.
- **Fixed Dash Size**
  If selected, the dash size will be fixed size.

Data Label tab

Specifies properties for data labels displayed on the bar gauges.



Static Data Label

Specifies properties of the static data labels. Not supported on gauge chart.

Font

Specifies the font format of text in the data labels.

- **Font list**
  Lists all the available font faces that can be selected to apply to the text.
- **Font Size**
  Specifies the font size of the text.
- **Font Color**
  Specifies the font color of the text.

  To change the color, click the color image button to display the color palette in which you can select a color, or click **More Colors** to customize the color within a wider range in the Pick a Color dialog, or click **More Fill Effects** to customize in the Fill Effects dialog. You can also input the hexadecimal value of a color directly in the text box after #.

- **Transparency**
  Specifies the transparency of the text.
- **Rotation**
  Specifies the rotation angle of the text around its center, in degrees. The default value is 0.
- **Shearing**
  Specifies the gradient of the text.

Effects

Specifies the special effects of text in the data labels.

- **Style**
  Specifies the font style of the text. It can be one of the following: plain, bold, italic, and bold italic.
- **Strikethrough**
  Specifies the style of the horizontal line with which the text is struck through. It can be one of the following: none, thin line, bold line, and double lines.
- **Underline**
  Specifies the style of the horizontal line under the text. It can be one of the following: none, single, single lower, bold line, bold lower, double lines, bold double, patterned line, and bold patterned.
- **Superscript**
  Raises the text above the baseline and changes the text to a smaller font size, if a smaller size is available.
- **Embossed**
  Makes the text appear to be raised off the page in relief.
- **Outlined**
  Displays the inner and outer borders of each character.
- **Subscript**
  Lowers the text below the baseline and changes the text to a smaller font size, if a smaller size is available.
- **Engraved**
  Makes the text appear to be imprinted or pressed into the page.
- **Shadowed**
  Adds a shadow beneath and to the right of the text.

Related topics:

- Formatting the gauge

Format Bubble dialog

This dialog appears when you double-click a bubble of a chart, or right-click it and select Format Bubble from the shortcut menu. It helps you to format the bubbles in a bubble chart, and consists of the following tabs:

- General tab
- Fill tab
- Border tab
- Data Label tab

OK
Applies the changes and closes the dialog.

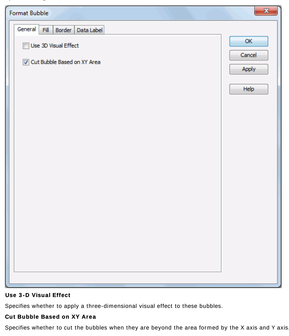
Cancel
Does not retain any changes and closes the dialog.

Apply
Applies all changes and leaves the dialog open.

Help
Displays the help document about this feature.

General tab

Specifies the general format of the bubble chart.



Use 3-D Visual Effect

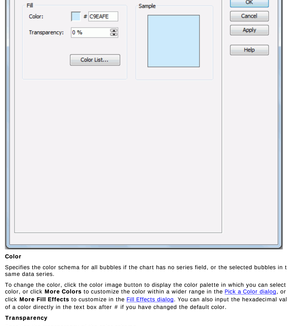
Specifies whether to apply a three-dimensional visual effect to these bubbles.

Cut Bubble Based on XY Area

Specifies whether to cut the bubbles when they are beyond the area formed by the X axis and Y axis.

Fill tab

Specifies the color, fill effect and transparency of the bubbles.



Color

Specifies the color schema for all bubbles if the chart has no series field, or the selected bubbles in the same data series.

To change the color, click the color image button to display the color palette in which you can select a color, or click **More Colors** to customize the color within a wider range in the Pick a Color dialog, or click **More Fill Effects** to customize in the Fill Effects dialog. You can also input the hexadecimal value of a color directly in the text box after # if you have changed the default color.

Transparency

Specifies the transparency of the color schema.

Color List

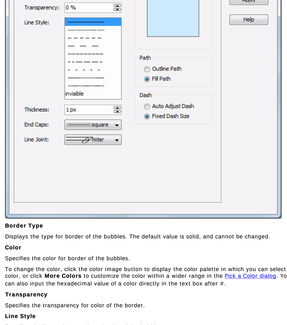
Opens the Color List dialog to modify color pattern for bubbles in the same data series.

Sample

Displays a preview sample of your selection.

Border tab

Specifies the properties for border of the bubbles, which take effect only when the Border property on chart paper is set to true in the Report Inspector.



Border Type

Displays the type for border of the bubbles. The default value is solid, and cannot be changed.

Color

Specifies the color for border of the bubbles.

To change the color, click the color image button to display the color palette in which you can select a color, or click **More Colors** to customize the color within a wider range in the Pick a Color dialog. You can also input the hexadecimal value of a color directly in the text box after #.

Transparency

Specifies the transparency for color of the border.

Line Style

Specifies the line style to apply to border of the bubbles.

Thickness

Specifies the thickness of the border, in pixels.

End Caps

Specifies the ending style of the border line.

- **butt**
  Ends unclosed sub paths and dash segments with no added decoration.
- **round**
  Ends unclosed sub paths and dash segments with a round decoration that has a radius equal to half of the width of the pen.
- **square**
  Ends unclosed sub paths and dash segments with a square projection that extends beyond the end of the segment to a distance equal to half of the line width.

Line Joint

Specifies the line joint style for the border line.

- **miter**
  Joins path segments by extending their outside edges until they meet.
- **round**
  Joins path segments by rounding off the corner at a radius of half the line width.
- **bevel**
  Joins path segments by connecting the outer corners of their wide outlines with a straight segment.

Sample

Displays a preview sample of your selection.

Path

Specifies the fill pattern of the border line.

- **Outline Path**
  Specifies the fill pattern of the border line to be outline path.
- **Fill Path**
  Specifies the fill pattern of the border line to be whole path.

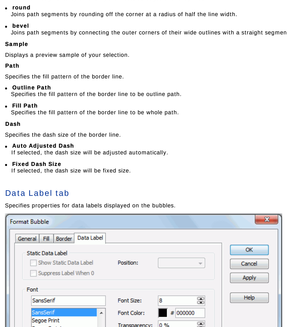
Dash

Specifies the dash size of the border line.

- **Auto Adjusted Dash**
  If selected, the dash size will be adjusted automatically.
- **Fixed Dash Size**
  If selected, the dash size will be fixed size.

Data Label tab

Specifies properties for data labels displayed on the bubbles.



Static Data Label

Specifies properties of the static data labels. Not supported on bubble chart.

Font

Specifies the font format of text in the data labels.

- **Font list**
  Lists all the available font faces that can be selected to apply to the text.

Font Size

Specifies the font size of the text.

Font Color

Specifies the font color of the text.

To change the color, click the color image button to display the color palette in which you can select a color, or click **More Colors** to customize the color within a wider range in the Pick a Color dialog, or click **More Fill Effects** to customize in the Fill Effects dialog. You can also input the hexadecimal value of a color directly in the text box after #.

Transparency

Specifies the transparency of the text.

Rotation

Specifies the rotation angle of the text around its center, in degrees. The default value is 0.

Shearing

Specifies the gradient of the text.

Effects

Specifies the special effects of text in the data labels.

- **Style**
  Specifies the font style of the text. It can be one of the following: plain, bold, italic, and bold italic.
- **Strikethrough**
  Specifies the style of the horizontal line with which the text is struck through. It can be one of the following: none, thin line, bold line, and double lines.
- **Underline**
  Specifies the style of the horizontal line under the text. It can be one of the following: none, single, single lower, bold line, bold lower, double lines, bold double, patterned line, and bold patterned.
- **Superscript**
  Raises the text above the baseline and changes the text to a smaller font size, if a smaller size is available.
- **Embossed**
  Makes the text appear to be raised off the page in relief.
- **Outlined**
  Displays the inner and outer borders of each character.
- **Subscript**
  Lowers the text below the baseline and changes the text to a smaller font size, if a smaller size is available.
- **Engraved**
  Makes the text appear to be imprinted or pressed into the page.
- **Shadowed**
  Adds a shadow beneath and to the right of the text.

Related topics:

Formatting the bubbles

Format Bubble Gauge dialog for library component

This dialog appears when you double-click a bubble gauge of a chart in a library component, or right-click it and then select Format Bubble Gauge from the shortcut menu. It enables you to format bubble gauges in a gauge chart, and consists of the following tabs:

- General tab
- Border tab
- Data Label tab
- Behaviors tab

OK
Applies the changes and closes the dialog.

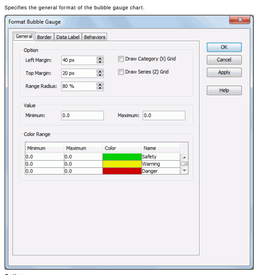
Cancel
Does not retain any changes and closes the dialog.

Apply
Applies all changes and leaves the dialog open.

Help
Displays the help document about this feature.

General tab

Specifies the general format of the bubble gauge chart.



Option

- **Left Margin**
  Specifies the gap between the left labels and left bubbles, in pixels.
- **Top Margin**
  Specifies the gap between top labels and top bubbles, in pixels.
- **Range Radius**
  Specifies the relative size of a bubble in a percentage of total bubble size.
- **Draw Category (X) Grid**
  Specifies whether to draw category grid.
- **Draw Series (Z) Grid**
  Specifies whether to draw series grid.

Value

Specifies the minimum and maximum values for the color range. The values will be equally divided into three ranges, each of which will be filled with the color you specify in the Color Range box automatically. Values beyond the minimum and maximum will be filled with grey color by default. Applied only when you do not specify ranges for the values in the Color Range box.

- **Minimum**
  Specifies the minimum value of the color range.
- **Maximum**
  Specifies the maximum value of the color range.

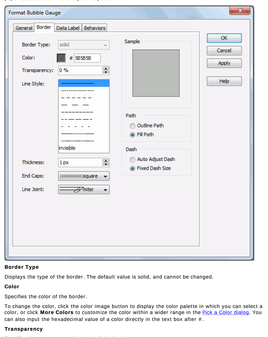
Color Range

Specifies different colors to fill the bubble gauges in different ranges.

- **Minimum**
  Specifies the minimum value of the range.
- **Maximum**
  Specifies the maximum value of the range.
- **Color**
  Specifies the color schema of the range. Click in the color cell to display the color palette in which you can select a color, or click **More Colors** to customize the color within a wider range in the Pick a Color dialog, or click **More Fill Effects** to customize in the Fill Effects dialog.
- **Name**
  Displays the name of the range.

Border tab

Specifies properties for border of the bubbles, which take effect only when the Border property on chart paper is set to true in the Report Inspector.



Border Type
Displays the type of the border. The default value is solid, and cannot be changed.

Color
Specifies the color of the border.
To change the color, click the color image button to display the color palette in which you can select a color, or click **More Colors** to customize the color within a wider range in the Pick a Color dialog. You can also input the hexadecimal value of a color directly in the text box after #.

Transparency
Specifies the transparency for color of the border.

Line Style
Specifies the line style to apply to the border.

Thickness
Specifies the thickness of the border, in pixels.

End Caps
Specifies the ending style of the border line.

- **butt**
  Ends unclosed sub paths and dash segments with no added decoration.
- **round**
  Ends unclosed sub paths and dash segments with a round decoration that has a radius equal to half of the width of the pen.
- **square**
  Ends unclosed sub paths and dash segments with a square projection that extends beyond the end of the segment to a distance equal to half of the line width.

Line Joint
Specifies the line joint style for the border line.

- **miter**
  Joins path segments by extending their outside edges until they meet.
- **round**
  Joins path segments by rounding off the corner at a radius of half the line width.
- **bevel**
  Joins path segments by connecting the outer corners of their wide outlines with a straight segment.

Sample
Displays a preview sample of your selection.

Path
Specifies the fill pattern of the border line.

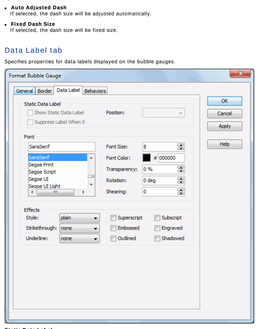
- **Outline Path**
  Specifies the fill pattern of the border line to be outline path.
- **Fill Path**
  Specifies the fill pattern of the border line to be whole path.

Dash
Specifies the dash size of border line.

- **Auto Adjusted Dash**
  If selected, the dash size will be adjusted automatically.
- **Fixed Dash Size**
  If selected, the dash size will be fixed size.

Data Label tab

Specifies properties for data labels displayed on the bubble gauges.



Static Data Label
Specifies properties of the static data labels. Not supported on gauge chart.

Font
Specifies the font format of text in the data labels.

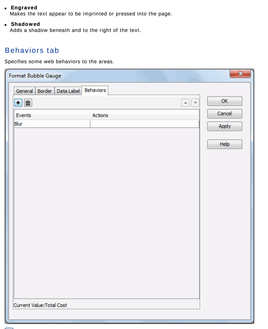
- **Font list**
  Lists all the available font faces that can be selected to apply to the text.
- **Font Size**
  Specifies the font size of the text.
- **Font Color**
  Specifies the font color of the text.
  To change the color, click the color image button to display the color palette in which you can select a color, or click **More Colors** to customize the color within a wider range in the Pick a Color dialog, or click **More Fill Effects** to customize in the Fill Effects dialog. You can also input the hexadecimal value of a color directly in the text box after #.
- **Transparency**
  Specifies the transparency of the text.
- **Rotation**
  Specifies the rotation angle of the text around its center, in degrees. The default value is 0.
- **Shearing**
  Specifies the gradient of the text.

Effects
Specifies the special effects of text in the data labels.

- **Style**
  Specifies the font style of the text. It can be one of the following: plain, bold, italic, and bold italic.
- **Strikethrough**
  Specifies the style of the horizontal line with which the text is struck through. It can be one of the following: none, thin line, bold line, and double lines.
- **Underline**
  Specifies the style of the horizontal line under the text. It can be one of the following: none, single, single lines, bold line, bold lower, double lines, bold double, patterned line, and bold patterned.
- **Superscript**
  Raises the text above the baseline and changes the text to a smaller font size, if a smaller size is available.
- **Embossed**
  Makes the text appear to be raised off the page in relief.
- **Outlined**
  Displays the inner and outer borders of each character.
- **Subscript**
  Lowers the text below the baseline and changes the text to a smaller font size, if a smaller size is available.
- **Engraved**
  Makes the text appear to be imprinted or pressed into the page.
- **Shadowed**
  Adds a shadow beneath and to the right of the text.

Behaviors tab

Specifies some web behaviors to the areas.



Adds a new web behavior line.

Removes the selected web behavior.

Moves the selected web behavior up a step.

Moves the selected web behavior down a step.

Events
Specifies the trigger event.

Actions
Specifies the action you want the event to trigger.

Opens the Web Action List dialog to bind a web action to the event.

Related topics:

- Formatting the gauge

Format Bubble Gauge dialog for report

This dialog appears when you double-click a bubble gauge of a chart in a page report or web report, or right-click it and then select Format Bubble Gauge from the shortcut menu. It enables you to format bubble gauges in a gauge chart, and consists of the following tabs:

- General tab
- Border tab
- Data Label tab

OK
Applies the changes and closes the dialog.

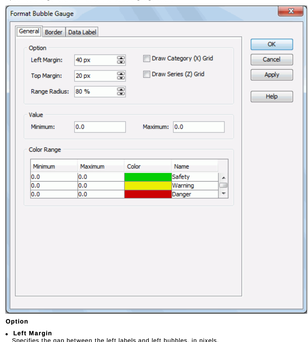
Cancel
Does not retain any changes and closes the dialog.

Apply
Applies all changes and leaves the dialog open.

Help
Displays the help document about this feature.

General tab

Specifies the general format of the bubble gauge chart.



Option

- **Left Margin**
  Specifies the gap between the left labels and left bubbles, in pixels.
- **Top Margin**
  Specifies the gap between top labels and top bubbles, in pixels.
- **Range Radius**
  Specifies the relative size of a bubble in a percentage of total bubble size.
- **Draw Category (X) Grid**
  Specifies whether to draw category grid.
- **Draw Series (Z) Grid**
  Specifies whether to draw series grid.

Value

Specifies the minimum and maximum values for the color range. The values will be equally divided into three ranges, each of which will be filled with the color you specify in the Color Range box automatically. Values beyond the minimum and maximum will be filled with grey color by default. Applied only when you do not specify ranges for the values in the Color Range box.

- **Minimum**
  Specifies the minimum value of the color range.
- **Maximum**
  Specifies the maximum value of the color range.

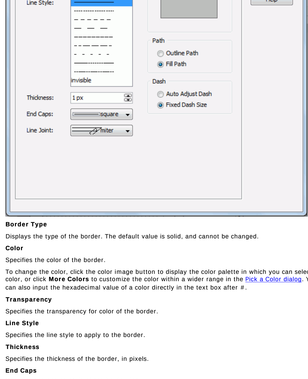
Color Range

Specifies different colors to fill the bubble gauges in different ranges.

- **Minimum**
  Specifies the minimum value of the range.
- **Maximum**
  Specifies the maximum value of the range.
- **Color**
  Specifies the color schema of the range. Click in the color cell to display the color palette in which you can select a color, or click **More Colors** to customize the color within a wider range in the Pick a Color dialog, or click **More Fill Effects** to customize in the Fill Effects dialog.
- **Name**
  Displays the name of the range.

Border tab

Specifies properties for border of the bubbles, which take effect only when the Border property on chart paper is set to true in the Report Inspector.



Border Type

Displays the type of the border. The default value is solid, and cannot be changed.

Color

Specifies the color of the border.

To change the color, click the color image button to display the color palette in which you can select a color, or click **More Colors** to customize the color within a wider range in the Pick a Color dialog. You can also input the hexadecimal value of a color directly in the text box after #.

Transparency

Specifies the transparency for color of the border.

Line Style

Specifies the line style to apply to the border.

Thickness

Specifies the thickness of the border, in pixels.

End Caps

Specifies the ending style of the border line.

- **butt**
  Ends unclosed sub paths and dash segments with no added decoration.
- **round**
  Ends unclosed sub paths and dash segments with a round decoration that has a radius equal to half of the width of the pen.
- **square**
  Ends unclosed sub paths and dash segments with a square projection that extends beyond the end of the segment to a distance equal to half of the line width.

Line Joint

Specifies the line joint style for the border line.

- **miter**
  Joins path segments by extending their outside edges until they meet.
- **round**
  Joins path segments by rounding off the corner at a radius of half the line width.
- **bevel**
  Joins path segments by connecting the outer corners of their wide outlines with a straight segment.

Sample

Displays a preview sample of your selection.

Path

Specifies the fill pattern of the border line.

- **Outline Path**
  Specifies the fill pattern of the border line to be outline path.
- **Fill Path**
  Specifies the fill pattern of the border line to be whole path.

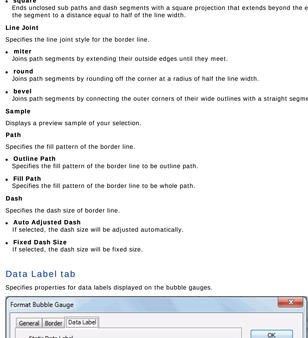
Dash

Specifies the dash size of border line.

- **Auto Adjusted Dash**
  If selected, the dash size will be adjusted automatically.
- **Fixed Dash Size**
  If selected, the dash size will be fixed size.

Data Label tab

Specifies properties for data labels displayed on the bubble gauges.



Static Data Label

Specifies properties of the static data labels. Not supported on gauge chart.

Font

Specifies the font format of text in the data labels.

- **Font list**
  Lists all the available font faces that can be selected to apply to the text.
- **Font Size**
  Specifies the font size of the text.
- **Font Color**
  Specifies the font color of the text.

  To change the color, click the color image button to display the color palette in which you can select a color, or click **More Colors** to customize the color within a wider range in the Pick a Color dialog, or click **More Fill Effects** to customize in the Fill Effects dialog. You can also input the hexadecimal value of a color directly in the text box after #.
- **Transparency**
  Specifies the transparency of the text.
- **Rotation**
  Specifies the rotation angle of the text around its center, in degrees. The default value is 0.
- **Shearing**
  Specifies the gradient of the text.

Effects

Specifies the special effects of text in the data labels.

- **Style**
  Specifies the font style of the text. It can be one of the following: plain, bold, italic, and bold italic.
- **Strikethrough**
  Specifies the style of the horizontal line with which the text is struck through. It can be one of the following: none, thin line, bold line, and double lines.
- **Underline**
  Specifies the style of the horizontal line under the text. It can be one of the following: none, single, single lower, bold line, bold lower, double lines, bold double, patterned line, and bold patterned.
- **Superscript**
  Raises the text above the baseline and changes the text to a smaller font size, if a smaller size is available.
- **Embossed**
  Makes the text appear to be raised off the page in relief.
- **Outlined**
  Displays the inner and outer borders of each character.
- **Subscript**
  Lowers the text below the baseline and changes the text to a smaller font size, if a smaller size is available.
- **Engraved**
  Makes the text appear to be imprinted or pressed into the page.
- **Shadowed**
  Adds a shadow beneath and to the right of the text.

Related topics:

- Formatting the gauge

Format Bullet dialog for library component

The dialog appears when you double-click a bullet of a chart in a library component, or right-click it and select Format Bullet from the shortcut menu. It helps you to format the bullets of a chart, and contains of the four following tabs:

- General tab
- Fill tab
- Border tab
- Data Label tab
- Behaviors tab

OK

Applies the changes and closes the dialog.

Cancel

Does not retain any changes and closes the dialog.

Apply

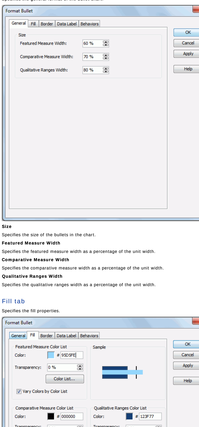
Applies all changes and leaves the dialog open.

Help

Displays the help document about this feature.

General tab

Specifies the general format of the bullet chart.



Size

Specifies the size of the bullets in the chart.

Featured Measure Width

Specifies the featured measure width as a percentage of the unit width.

Comparative Measure Width

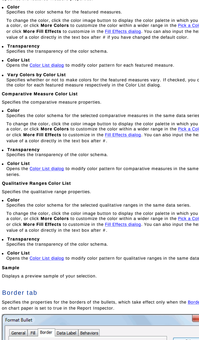
Specifies the comparative measure width as a percentage of the unit width.

Qualitative Ranges Width

Specifies the qualitative ranges width as a percentage of the unit width.

Fill tab

Specifies the fill properties.



Featured Measure Color List

Specifies the featured measure properties.

- **Color**
  Specifies the color scheme for the featured measures.
  To change the color, click the color image button to display the color palette in which you can select a color, or click **More Colors** to customize the color within a wider range in the Pick a Color dialog, or click **More Fill Effects** to customize in the Fill Effects dialog. You can also input the hexadecimal value of a color directly in the text box after #. If you have changed the default color.
- **Transparency**
  Specifies the transparency of the color scheme.
- **Color List**
  Opens the Color List dialog to modify color pattern for each featured measure.
- **Very Colors by Color List**
  Specifies whether or not to make colors for the featured measures vary. If checked, you can specify the color for each featured measure respectively in the Color List dialog.

Comparative Measure Color List

Specifies the comparative measure properties.

- **Color**
  Specifies the color scheme for the selected comparative measures in the same data series.
  To change the color, click the color image button to display the color palette in which you can select a color, or click **More Colors** to customize the color within a wider range in the Pick a Color dialog, or click **More Fill Effects** to customize in the Fill Effects dialog. You can also input the hexadecimal value of a color directly in the text box after #.
- **Transparency**
  Specifies the transparency of the color scheme.
- **Color List**
  Opens the Color List dialog to modify color pattern for comparative measures in the same data series.

Qualitative Ranges Color List

Specifies the qualitative range properties.

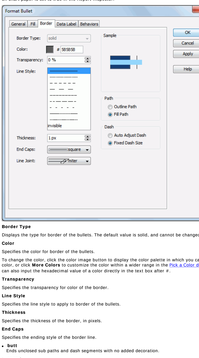
- **Color**
  Specifies the color scheme for the selected qualitative ranges in the same data series.
  To change the color, click the color image button to display the color palette in which you can select a color, or click **More Colors** to customize the color within a wider range in the Pick a Color dialog, or click **More Fill Effects** to customize in the Fill Effects dialog. You can also input the hexadecimal value of a color directly in the text box after #.
- **Transparency**
  Specifies the transparency of the color scheme.
- **Color List**
  Opens the Color List dialog to modify color pattern for qualitative ranges in the same data series.

Sample

Displays a preview sample of your selection.

Border tab

Specifies the properties for the borders of the bullets, which take effect only when the Border property on chart paper is set to true in the Report Inspector.



Border Type

Displays the type for border of the bullets. The default value is solid, and cannot be changed.

Color

Specifies the color for border of the bullets.

To change the color, click the color image button to display the color palette in which you can select a color, or click **More Colors** to customize the color within a wider range in the Pick a Color dialog. You can also input the hexadecimal value of a color directly in the text box after #.

Transparency

Specifies the transparency for color of the border.

Line Style

Specifies the line style to apply to border of the bullets.

Thickness

Specifies the thickness of the border, in pixels.

End Caps

Specifies the ending style of the border line.

- **butt**
  Ends unclosed sub paths and dash segments with no added decoration.
- **round**
  Ends unclosed sub paths and dash segments with a round decoration that has a radius equal to half of the width of the pen.
- **square**
  Ends unclosed sub paths and dash segments with a square projection that extends beyond the end of the segment to a distance equal to half of the line width.

Line Joint

Specifies the line joint style for the border line.

- **miter**
  Joins path segments by extending their outside edges until they meet.
- **round**
  Joins path segments by rounding off the corner at a radius of half the line width.
- **bevel**
  Joins path segments by connecting the outer corners of their wide outlines with a straight segment.

Sample

Displays a preview sample of your selection.

Path

Specifies the fill pattern of the border line.

- **Outline Path**
  Specifies the fill pattern of the border line to be outline path.
- **Fill Path**
  Specifies the fill pattern of the border line to be whole path.

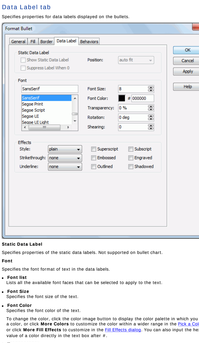
Dash

Specifies the dash size of the border line.

- **Auto Adjusted Dash**
  If selected, the dash size will be adjusted automatically.
- **Fixed Dash Size**
  If selected, the dash size will be fixed size.

Data Label tab

Specifies properties for data labels displayed on the bullets.



Static Data Label

Specifies properties of the static data labels. Not supported on bullet chart.

Font

Specifies the text format of text in the data labels.

- **Font list**
  Lists all the available font faces that can be selected to apply to the text.
- **Font Size**
  Specifies the font size of the text.
- **Font Color**
  Specifies the font color of the text.
  To change the color, click the color image button to display the color palette in which you can select a color, or click **More Colors** to customize the color within a wider range in the Pick a Color dialog, or click **More Fill Effects** to customize in the Fill Effects dialog. You can also input the hexadecimal value of a color directly in the text box after #.
- **Transparency**
  Specifies the transparency of the text.
- **Rotation**
  Specifies the rotation angle of the text around its center, in degrees. The default value is 0.
- **Shearing**
  Specifies the gradient of the text.

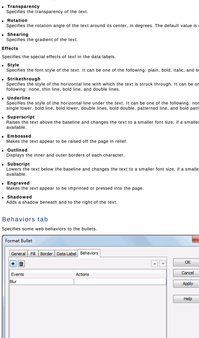
Effects

Specifies the special effects of text in the data labels.

- **Style**
  Specifies the font style of the text. It can be one of the following: plain, bold, italic, and bold italic.
- **Strikethrough**
  Specifies the style of the horizontal line with which the text is struck through. It can be one of the following: none, thin line, bold line, and double lines.
- **Underline**
  Specifies the style of the horizontal line under the text. It can be one of the following: none, single, single lower, bold line, bold lower, double lines, bold double, patterned line, and bold patterned.
- **Superscript**
  Raises the text above the baseline and changes the text to a smaller font size, if a smaller size is available.
- **Embossed**
  Makes the text appear to be raised off the page in relief.
- **Outlined**
  Displays the inner and outer borders of each character.
- **Subscript**
  Lowers the text below the baseline and changes the text to a smaller font size, if a smaller size is available.
- **Engraved**
  Makes the text appear to be imprinted or pressed into the page.
- **Shadowed**
  Adds a shadow beneath and to the right of the text.

Behaviors tab

Specifies some web behaviors to the bullets.



Adds a new web behavior line.

Removes the selected web behavior.

Moves the selected web behavior up a step.

Moves the selected web behavior down a step.

Events

Specifies the trigger event.

Actions

Specifies the action you want the event to trigger.

Opens the Web Action List dialog to bind a web action to the event.

Related topics:

- Formatting the bullets

Format Bullet dialog for report

The dialog appears when you double-click a bullet of a chart in a page report or web report, or right-click it and select Format Bullet from the shortcut menu. It helps you to format the bullets of a chart, and consists of the four following tabs:

- General tab
- Fill tab
- Border tab
- Data Label tab

OK
Applies the changes and closes the dialog.

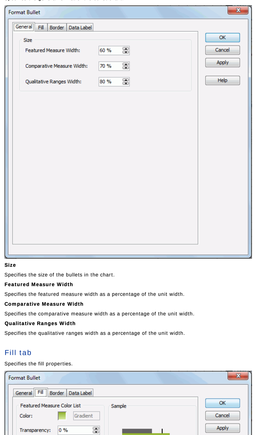
Cancel
Does not retain any changes and closes the dialog.

Apply
Applies all changes and leaves the dialog open.

Help
Displays the help document about this feature.

General tab

Specifies the general format of the bullet chart.



Size
Specifies the size of the bullets in the chart.

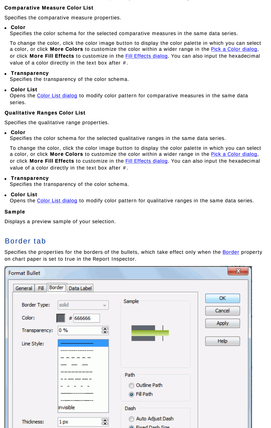
Featured Measure Width
Specifies the featured measure width as a percentage of the unit width.

Comparative Measure Width
Specifies the comparative measure width as a percentage of the unit width.

Qualitative Ranges Width
Specifies the qualitative ranges width as a percentage of the unit width.

Fill tab

Specifies the fill properties.



Featured Measure Color List
Specifies the featured measure properties.

- **Color**
  Specifies the color schema for the featured measures.
  To change the color, click the color image button to display the color palette in which you can select a color, or click **More Colors** to customize the color within a wider range in the Pick a Color dialog, or click **More Fill Effects** to customize in the Fill Effects dialog. You can also input the hexadecimal value of a color directly in the text box after # if you have changed the default color.
- **Transparency**
  Specifies the transparency of the color schema.
- **Color List**
  Opens the Color List dialog to modify color pattern for each featured measure.
- **Vary Colors by Color List**
  Specifies whether or not to make colors for the featured measures vary. If checked, you can specify the color for each featured measure respectively in the Color List dialog.

Comparative Measure Color List
Specifies the comparative measure properties.

- **Color**
  Specifies the color schema for the selected comparative measures in the same data series.
  To change the color, click the color image button to display the color palette in which you can select a color, or click **More Colors** to customize the color within a wider range in the Pick a Color dialog, or click **More Fill Effects** to customize in the Fill Effects dialog. You can also input the hexadecimal value of a color directly in the text box after #.
- **Transparency**
  Specifies the transparency of the color schema.
- **Color List**
  Opens the Color List dialog to modify color pattern for comparative measures in the same data series.

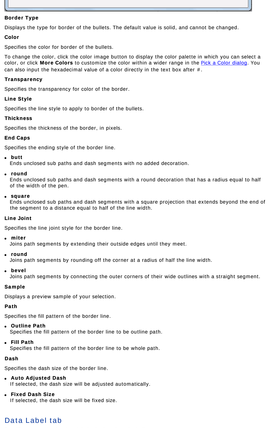
Qualitative Ranges Color List
Specifies the qualitative range properties.

- **Color**
  Specifies the color schema for the selected qualitative ranges in the same data series.
  To change the color, click the color image button to display the color palette in which you can select a color, or click **More Colors** to customize the color within a wider range in the Pick a Color dialog, or click **More Fill Effects** to customize in the Fill Effects dialog. You can also input the hexadecimal value of a color directly in the text box after #.
- **Transparency**
  Specifies the transparency of the color schema.
- **Color List**
  Opens the Color List dialog to modify color pattern for qualitative ranges in the same data series.

Sample
Displays a preview sample of your selection.

Border tab

Specifies the properties for the borders of the bullets, which take effect only when the Border property on chart paper is set to true in the Report Inspector.



Border Type
Displays the type for border of the bullets. The default value is solid, and cannot be changed.

Color
Specifies the color for border of the bullets.
To change the color, click the color image button to display the color palette in which you can select a color, or click **More Colors** to customize the color within a wider range in the Pick a Color dialog. You can also input the hexadecimal value of a color directly in the text box after #.

Transparency
Specifies the transparency for color of the border.

Line Style
Specifies the line style to apply to border of the bullets.

Thickness
Specifies the thickness of the border, in pixels.

End Caps
Specifies the ending style of the border line.

- **butt**
  Ends unclosed sub paths and dash segments with no added decoration.
- **round**
  Ends unclosed sub paths and dash segments with a round decoration that has a radius equal to half of the width of the pen.
- **square**
  Ends unclosed sub paths and dash segments with a square projection that extends beyond the end of the segment to a distance equal to half of the line width.

Line Joint
Specifies the line joint style for the border line.

- **miter**
  Joins path segments by extending their outside edges until they meet.
- **round**
  Joins path segments by rounding off the corner at a radius of half the line width.
- **bevel**
  Joins path segments by connecting the outer corners of their wide outlines with a straight segment.

Sample
Displays a preview sample of your selection.

Path
Specifies the fill pattern of the border line.

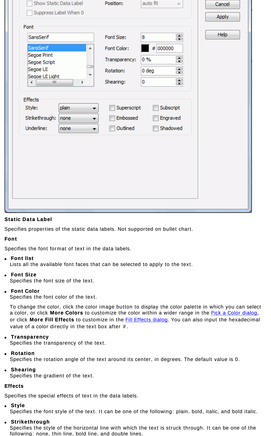
- **Outline Path**
  Specifies the fill pattern of the border line to be outline path.
- **Fill Path**
  Specifies the fill pattern of the border line to be whole path.

Dash
Specifies the dash size of the border line.

- **Auto Adjusted Dash**
  If selected, the dash size will be adjusted automatically.
- **Fixed Dash Size**
  If selected, the dash size will be fixed size.

Data Label tab

Specifies properties for data labels displayed on the bullets.



Static Data Label
Specifies properties of the static data labels. Not supported on bullet chart.

Font
Specifies the font format of text in the data labels.

- **Font list**
  Lists all the available font faces that can be selected to apply to the text.
- **Font Size**
  Specifies the font size of the text.
- **Font Color**
  Specifies the font color of the text.
  To change the color, click the color image button to display the color palette in which you can select a color, or click **More Colors** to customize the color within a wider range in the Pick a Color dialog, or click **More Fill Effects** to customize in the Fill Effects dialog. You can also input the hexadecimal value of a color directly in the text box after #.
- **Transparency**
  Specifies the transparency of the text.
- **Rotation**
  Specifies the rotation angle of the text around its center, in degrees. The default value is 0.
- **Shearing**
  Specifies the gradient of the text.

Effects
Specifies the special effects of text in the data labels.

- **Style**
  Specifies the font style of the text. It can be one of the following: plain, bold, italic, and bold italic.
- **Strikethrough**
  Specifies the style of the horizontal line with which the text is struck through. It can be one of the following: none, thin line, bold line, and double lines.
- **Underline**
  Specifies the style of the horizontal line under the text. It can be one of the following: none, single, single lower, bold line, bold lower, double lines, bold double, patterned line, and bold patterned.
- **Superscript**
  Raises the text above the baseline and changes the text to a smaller font size, if a smaller size is available.
- **Embossed**
  Makes the text appear to be raised off the page in relief.
- **Outlined**
  Displays the inner and outer borders of each character.
- **Subscript**
  Lowers the text below the baseline and changes the text to a smaller font size, if a smaller size is available.
- **Engraved**
  Makes the text appear to be imprinted or pressed into the page.
- **Shadowed**
  Adds a shadow beneath and to the right of the text.

Related topics:

- Formatting the bullets

Format Category (X) Axis dialog for library component

This dialog appears when you double-click the category (X) axis of a chart in a library component, or right-click it and then select Format Axes > Format Category (X) Axis from the shortcut menu. It helps you to format the category (X) axis of a chart, and consists of the following tabs:

- Axis tab
- Tick Mark tab
- Font tab
- Orientation tab
- Format tab
- Behaviors tab

OK

Applies the changes and closes the dialog.

Cancel

Does not retain any changes and closes the dialog.

Apply

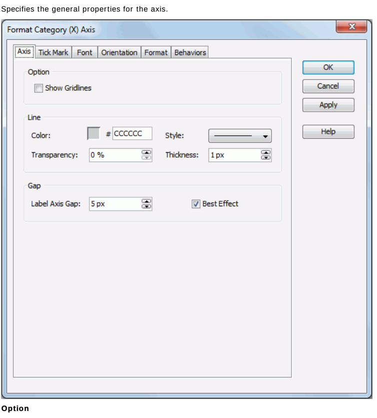
Applies all changes and leaves the dialog open.

Help

Displays the help document about this feature.

Axis tab

Specifies the general properties for the axis.



Option

Specifies the options for the axis.

- **Minimum Value**
  Specifies the minimum value that is to be displayed on the axis. Available only to bubble charts which use the axis to show numeric data and scatter charts.
- **Maximum Value**
  Specifies the maximum value that is to be displayed on the axis. Available only to bubble charts which use the axis to show numeric data and scatter charts.
- **Increment**
  Specifies the difference between two adjacent values on the axis. Available only to bubble charts which use the axis to show numeric data and scatter charts.
- **Number of Tick Marks**
  Specifies how many tick marks to be displayed on the axis. Available only to bubble charts which use the axis to show numeric data and scatter charts.
- **Show Gridlines**
  Specifies whether to show gridlines on the axis.

Line

Specifies the line style for the axis.

- **Color**
  Specifies the color of the axis.

  To change the color, click the color image button to display the color palette in which you can select a color, or click **More Colors** to customize the color within a wider range in the Pick a Color dialog. You can also input the hexadecimal value of a color directly in the text box after #.
- **Style**
  Specifies the style for the line of the axis.
- **Transparency**
  Specifies the transparency for the color of the axis.
- **Thickness**
  Specifies the thickness for the line of the axis, in pixels.

Gap

Specifies the gap properties for the labels on the axis.

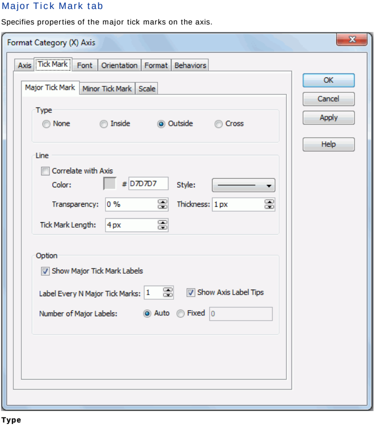
- **Label Axis Gap**
  Specifies the distance between the label and the axis, in pixels.
- **Best Effect**
  Specifies whether to adjust the labels automatically to make them placed best.

Tick Mark tab

The tab consists of three sub tabs: major tick mark tab, minor tick mark tab, and scale tab.

Major Tick Mark tab

Specifies properties of the major tick marks on the axis.



Type

Specifies the type of the major tick marks on the axis.

- **None**
  If selected, major tick marks will not be shown on the axis and it will be meaningless to specify all the other major tick mark related properties.
- **Inside**
  If selected, major tick marks will be inside the chart.
- **Outside**
  If selected, major tick marks will be outside the chart.
- **Cross**
  If selected, major tick marks will be across the axis.

Line

Specifies the line properties of the major tick marks on the axis.

- **Correlate with Axis**
  If checked, the line properties of the major tick marks will correlate with that of the axis automatically.
  - **Color**
    Specifies the color of the major tick marks.

    To change the color, click the color image button to display the color palette in which you can select a color, or click **More Colors** to customize the color within a wider range in the Pick a Color dialog. You can also input the hexadecimal value of a color directly in the text box after #.
  - **Style**
    Specifies the type of the major tick marks.
  - **Transparency**
    Specifies the transparency for the color of the major tick marks.
  - **Thickness**
    Specifies the thickness of the major tick marks, in pixels.
- **Tick Mark Length**
  Specifies the length of the major tick marks, in pixels.

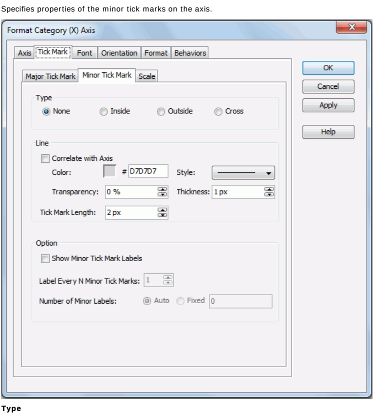
Option

Specifies the other properties of the major tick mark labels on the axis.

- **Show Major Tick Mark Labels**
  Specifies whether to display the labels of the major tick marks on the axis. If checked, the following properties will be enabled.
- **Label Every N Major Tick Marks**
  Specifies the frequency at which the major tick marks will be labeled.
- **Show Axis Label Tips**
  Specifies whether to display the complete label text when the mouse pointer points at a label on the axis.
- **Number of Major Labels**
  Specifies how many major tick mark labels to be displayed on the axis.
  - **Auto**
    If checked, all major tick mark labels will be shown.
  - **Fixed**
    If checked, you can specify the number of the major tick mark labels to be displayed on the axis.

Minor Tick Mark tab

Specifies properties of the minor tick marks on the axis.



Type

Specifies the type of the minor tick marks on the axis.

- **None**
  If selected, minor tick marks will not be shown on the axis and it will be meaningless to specify all the other minor tick mark related properties.
- **Inside**
  If selected, minor tick marks will be inside the chart.
- **Outside**
  If selected, minor tick marks will be outside the chart.
- **Cross**
  If selected, minor tick marks will be across the axis.

Line

Specifies the line properties of the minor tick marks on the axis.

- **Correlate with Axis**
  If checked, the line properties of the minor tick marks will correlate with that of the axis automatically.
  - **Color**
    Specifies the color of the minor tick marks.

    To change the color, click the color image button to display the color palette in which you can select a color, or click **More Colors** to customize the color within a wider range in the Pick a Color dialog. You can also input the hexadecimal value of a color directly in the text box after #.
  - **Style**
    Specifies the type of the minor tick marks.
  - **Transparency**
    Specifies the transparency for the color of the minor tick marks.
  - **Thickness**
    Specifies the thickness of the minor tick marks, in pixels.
- **Tick Mark Length**
  Specifies the length of the minor tick marks, in pixels.

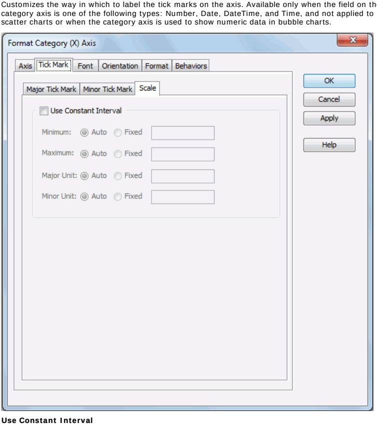
Option

Specifies the other properties of the minor tick marks on the axis.

- **Show Minor Tick Mark Labels**
  Specifies whether to display the labels of the minor tick marks on the axis. If checked, the following properties will be enabled.
- **Label Every N Minor Tick Marks**
  Specifies the frequency at which the minor tick marks will be labeled.
- **Number of Minor Labels**
  Specifies how many minor tick mark labels to be displayed on the axis.
  - **Auto**
    If checked, all minor tick mark labels will be shown.
  - **Fixed**
    If checked, you can specify the number of the minor tick mark labels to be displayed on the axis.

Scale tab

Customizes the way in which to label the tick marks on the axis. Available only when the field on the category axis is one of the following types: Number, Date, DateTime, and Time, and not applied to scatter charts or when the category axis is used to show numeric data in bubble charts.



Use Constant Interval

Specifies whether to use a constant interval to label the tick marks. If checked, the values of the tick marks will be increased continually on the axis based on the following properties, instead of just using the data values.

Minimum

Specifies the minimum value which will be used to label the tick marks.

- **Auto**
  If checked, the minimum value will be defined by JReport automatically.
- **Fixed**
  If checked, you can define the minimum value as required. Input the value in the text box, or specify it in the calendar if the field on the category axis is of Date, DateTime or Time type.

Maximum

Specifies the maximum value which will be used to label the tick marks.

- **Auto**
  If checked, the maximum value will be defined by JReport automatically.
- **Fixed**
  If checked, you can define the maximum value as required. Input the value in the text box, or specify it in the calendar if the field on the category axis is of Date, DateTime or Time type.

Major Unit

Specifies the unit between two adjacent major tick marks.

- **Auto**
  If checked, the unit will be defined by JReport automatically.
- **Fixed**
  If checked, you can define the unit as required. Input the value in the text box, or choose the desired one from the drop-down list if the field on the category axis is of Date, DateTime or Time type.

Minor Unit

Specifies the unit between two adjacent minor tick marks.

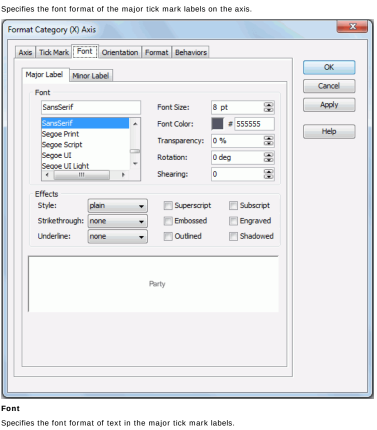
- **Auto**
  If checked, the unit will be defined by JReport automatically.
- **Fixed**
  If checked, you can define the unit as required. Input the value in the text box, or choose the desired one from the drop-down list if the field on the category axis is of the Date, DateTime or Time type.

Font tab

The tab consists of two sub tabs: major label tab and minor label tab.

Major Label tab

Specifies the font format of the major tick mark labels on the axis.



Font

Specifies the font format of text in the major tick mark labels.

- **Font list**
  Lists all the available font faces that can be selected to apply to the text.
- **Font Size**
  Specifies the font size of the text.
- **Font Color**
  Specifies the font color of the text.

  To change the color, click the color image button to display the color palette in which you can select a color, or click **More Colors** to customize the color within a wider range in the Pick a Color dialog, or click **More Fill Effects** to customize in the Fill Effects dialog. You can also input the hexadecimal value of a color directly in the text box after #.
- **Transparency**
  Specifies the transparency of the text.
- **Rotation**
  Specifies the rotation angle of the text around its center, in degrees. The default value is 0.
- **Shearing**
  Specifies the gradient of the text.

Effects

Specifies the special effects of text in the major tick mark labels.

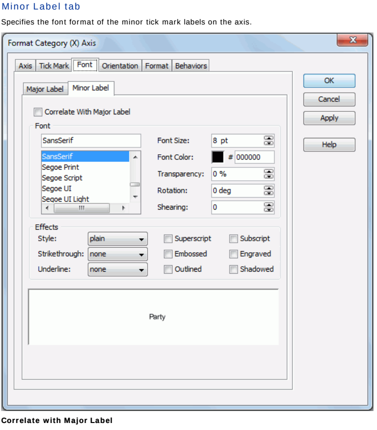
- **Style**
  Specifies the font style of the text. It can be one of the following: plain, bold, italic, and bold italic.
- **Strikethrough**
  Specifies the style of the horizontal line with which the text is struck through. It can be one of the following: none, thin line, bold line, and double lines.
- **Underline**
  Specifies the style of the horizontal line under the text. It can be one of the following: none, single, single lower, bold line, bold lower, double lines, bold double, patterned line, and bold patterned.
- **Superscript**
  Raises the text above the baseline and changes the text to a smaller font size, if a smaller size is available.
- **Embossed**
  Makes the text appear to be raised off the page in relief.
- **Outlined**
  Displays the inner and outer borders of each character.
- **Subscript**
  Lowers the text below the baseline and changes the text to a smaller font size, if a smaller size is available.
- **Engraved**
  Makes the text appear to be imprinted or pressed into the page.
- **Shadowed**
  Adds a shadow beneath and to the right of the text.

Sample

Displays a sample of the specified font and any text effects.

Minor Label tab

Specifies the font format of the minor tick mark labels on the axis.



Correlate with Major Label

If checked, the font format of the minor tick mark labels will correlate with that of the major tick mark labels automatically. Only when it is unchecked can the font properties of the minor tick mark labels take effect.

Font

Specifies the font format of text in the minor tick mark labels.

- **Font list**
  Lists all the available font faces that can be selected to apply to the text.
- **Font Size**
  Specifies the font size of the text.
- **Font Color**
  Specifies the font color of the text.

  To change the color, click the color image button to display the color palette in which you can select a color, or click **More Colors** to customize the color within a wider range in the Pick a Color dialog, or click **More Fill Effects** to customize in the Fill Effects dialog. You can also input the hexadecimal value of a color directly in the text box after #.
- **Transparency**
  Specifies the transparency of the text.
- **Rotation**
  Specifies the rotation angle of the text around its center, in degrees. The default value is 0.
- **Shearing**
  Specifies the gradient of the text.

Effects

Specifies the special effects of text in the minor tick mark labels.

- **Style**
  Specifies the font style of the text. It can be one of the following: plain, bold, italic, and bold italic.
- **Strikethrough**
  Specifies the style of the horizontal line with which the text is struck through. It can be one of the following: none, thin line, bold line, and double lines.
- **Underline**
  Specifies the style of the horizontal line under the text. It can be one of the following: none, single, single lower, bold line, bold lower, double lines, bold double, patterned line, and bold patterned.
- **Superscript**
  Raises the text above the baseline and changes the text to a smaller font size, if a smaller size is available.
- **Embossed**
  Makes the text appear to be raised off the page in relief.
- **Outlined**
  Displays the inner and outer borders of each character.
- **Subscript**
  Lowers the text below the baseline and changes the text to a smaller font size, if a smaller size is available.
- **Engraved**
  Makes the text appear to be imprinted or pressed into the page.
- **Shadowed**
  Adds a shadow beneath and to the right of the text.

Sample

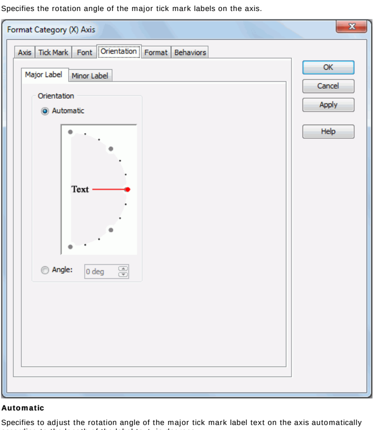
Displays a sample of the specified font and any text effects.

Orientation tab

The tab consists of two sub tabs: major label tab and minor label tab.

Major Label tab

Specifies the rotation angle of the major tick mark labels on the axis.



Automatic

Specifies to adjust the rotation angle of the major tick mark label text on the axis automatically according to the length of the label text, in degrees.

When this option is checked by default:

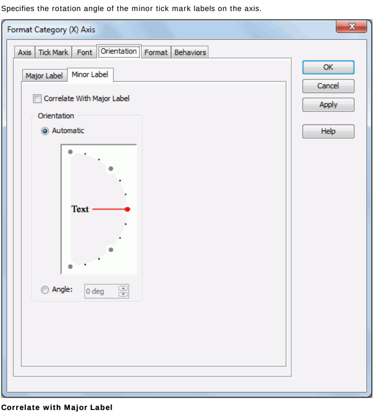
- If the text can be completely displayed horizontally, the default rotation angle will be 0.
- If the text cannot be completely displayed horizontally, the default rotation angle will be 30 anticlockwise, and the cut off part will be shown as suspension points.

Angle

Specifies to customize the rotation angle of the major tick mark label text on the axis. When you change the rotation angle in the text box, the red line in the spin box will change correspondingly, and vice versa.

Minor Label tab

Specifies the rotation angle of the minor tick mark labels on the axis.



Correlate with Major Label

If checked, the orientation setting of the minor tick mark labels will correlate with that of the major tick mark labels automatically. Only when it is unchecked can the orientation properties of the minor tick mark labels take effect.

Automatic

Specifies to adjust the rotation angle of the minor tick mark label text on the axis automatically according to the length of the label text, in degrees.

When this option is checked by default:

- If the text can be completely displayed horizontally, the default rotation angle will be 0.
- If the text cannot be completely displayed horizontally, the default rotation angle will be 30 anticlockwise, and the cut off part will be shown as suspension points.

Angle

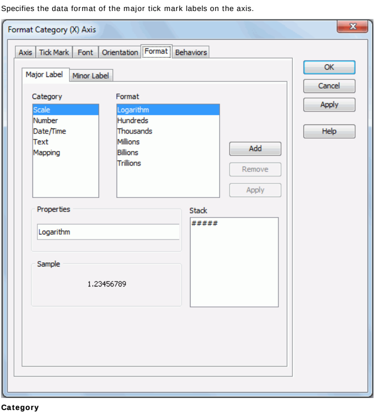
Specifies to customize the rotation angle of the minor tick mark label text on the axis. When you change the rotation angle in the text box, the red line in the spin box will change correspondingly, and vice versa.

Format tab

The tab consists of two sub tabs: major label tab and minor label tab.

Major Label tab

Specifies the data format of the major tick mark labels on the axis.



Category

Lists the category types. Select one to customize its format.

Format

Displays all the formats of the selected category. Select the required one and click Add to add it as the format of the specified category. You can add only one format for each category. For more details about the formats, see A detailed chart property reference.

Properties

Displays the properties of the format you select. If the formats listed in the Format box cannot meet your requirement, define the format in the text field and then click Add to add it as the format of the specified category.

Sample

Displays the selected format effects.

Stack

Lists all the formats you select from different categories.

Add

Adds a format to the Stack list box.

Remove

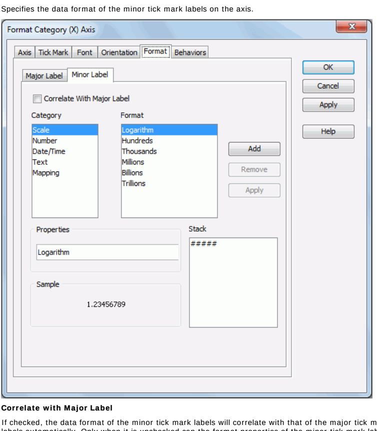
Removes a format from the Stack list box.

Apply

Applies the specified format in the Stack list box to the major tick mark labels.

Minor Label tab

Specifies the data format of the minor tick mark labels on the axis.



Correlate with Major Label

If checked, the data format of the minor tick mark labels will correlate with that of the major tick mark labels automatically. Only when it is unchecked can the format properties of the minor tick mark labels take effect.

Category

Lists the category types. Select one to customize its format.

Format

Displays all the formats of the selected category. Select the required one and click Add to add it as the format of the specified category. You can add only one format for each category. For more details about the formats, see A detailed chart property reference.

Properties

Displays the properties of the format you select. If the formats listed in the Format box cannot meet your requirement, define the format in the text field and then click Add to add it as the format of the specified category.

Sample

Displays the selected format effects.

Stack

Lists all the formats you select from different categories.

Add

Adds a format to the Stack list box.

Remove

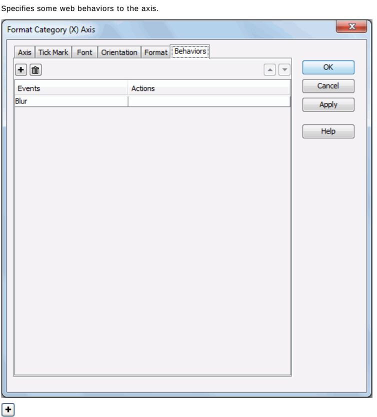
Removes a format from the Stack list box.

Apply

Applies the specified format in the Stack list box to the minor tick mark labels.

Behaviors tab

Specifies some web behaviors to the axis.



Adds a new web behavior line.

Removes the selected web behavior.

Moves the selected web behavior up a step.

Moves the selected web behavior down a step.

Events

Specifies the trigger event.

Actions

Specifies the action you want the event to trigger.

- Opens the Web Action List dialog to bind a web action to the event.

Related topics:

- Formatting the category axis

Format Category (X) Axis dialog for report

This dialog appears when you double-click the category (X) axis of a chart in a report, or right-click it and then select Format Axes > Format Category (X) Axis from the shortcut menu. It helps you to format the category (X) axis of a chart, and consists of the following tabs:

- Axis tab
- Tick Mark tab
- Font tab
- Orientation tab
- Format tab

OK

Applies the changes and closes the dialog.

Cancel

Does not retain any changes and closes the dialog.

Apply

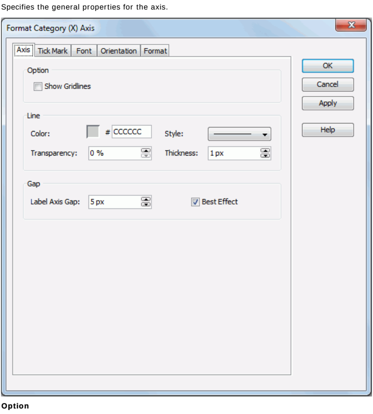
Applies all changes and leaves the dialog open.

Help

Displays the help document about this feature.

Axis tab

Specifies the general properties for the axis.



Option

Specifies the options for the axis.

- **Minimum Value**
  Specifies the minimum value that is to be displayed on the axis. Available only to bubble charts which use the axis to show numeric data and scatter charts.
- **Maximum Value**
  Specifies the maximum value that is to be displayed on the axis. Available only to bubble charts which use the axis to show numeric data and scatter charts.
- **Increment**
  Specifies the difference between two adjacent values on the axis. Available only to bubble charts which use the axis to show numeric data and scatter charts.
- **Number of Tick Marks**
  Specifies how many tick marks to be displayed on the axis. Available only to bubble charts which use the axis to show numeric data and scatter charts.
- **Show Gridlines**
  Specifies whether to show gridlines on the axis.

Line

Specifies the line style for the axis.

- **Color**
  Specifies the color of the axis.

  To change the color, click the color image button to display the color palette in which you can select a color, or click **More Colors** to customize the color within a wider range in the Pick a Color dialog. You can also input the hexadecimal value of a color directly in the text box after #.
- **Style**
  Specifies the style for the line of the axis.
- **Transparency**
  Specifies the transparency for the color of the axis.
- **Thickness**
  Specifies the thickness for the line of the axis, in pixels.

Gap

Specifies the gap properties for the labels on the axis.

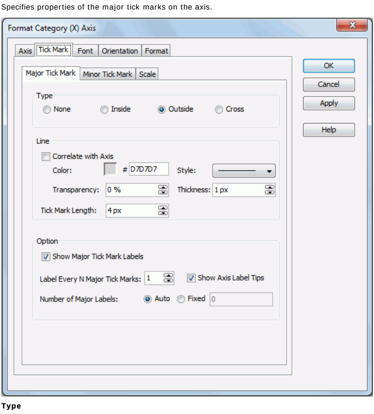
- **Label Axis Gap**
  Specifies the distance between the label and the axis, in pixels.
- **Best Effect**
  Specifies whether to adjust the labels automatically to make them placed best.

Tick Mark tab

The tab consists of three sub tabs: major tick mark tab, minor tick mark tab, and scale tab.

Major Tick Mark tab

Specifies properties of the major tick marks on the axis.



Type

Specifies the type of the major tick marks on the axis.

- **None**
  If selected, major tick marks will not be shown on the axis and it will be meaningless to specify all the other major tick mark related properties.
- **Inside**
  If selected, major tick marks will be inside the chart.
- **Outside**
  If selected, major tick marks will be outside the chart.
- **Cross**
  If selected, major tick marks will be across the axis.

Line

Specifies the line properties of the major tick marks on the axis.

- **Correlate with Axis**
  If checked, the line properties of the major tick marks will correlate with that of the axis automatically.
  - **Color**
    Specifies the color of the major tick marks.

    To change the color, click the color image button to display the color palette in which you can select a color, or click **More Colors** to customize the color within a wider range in the Pick a Color dialog. You can also input the hexadecimal value of a color directly in the text box after #.
  - **Style**
    Specifies the type of the major tick marks.
  - **Transparency**
    Specifies the transparency for the color of the major tick marks.
  - **Thickness**
    Specifies the thickness of the major tick marks, in pixels.
- **Tick Mark Length**
  Specifies the length of the major tick marks, in pixels.

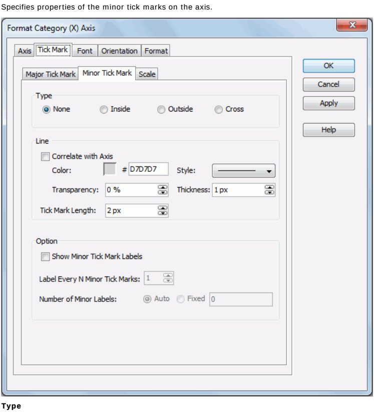
Option

Specifies the other properties of the major tick mark labels on the axis.

- **Show Major Tick Mark Labels**
  Specifies whether to display the labels of the major tick marks on the axis. If checked, the following properties will be enabled.
- **Label Every N Major Tick Marks**
  Specifies the frequency at which the major tick marks will be labeled.
- **Show Axis Label Tips**
  Specifies whether to display the complete label text when the mouse pointer points at a label on the axis.
- **Number of Major Labels**
  Specifies how many major tick mark labels to be displayed on the axis.
  - **Auto**
    If checked, all major tick mark labels will be shown.
  - **Fixed**
    If checked, you can specify the number of the major tick mark labels to be displayed on the axis.

Minor Tick Mark tab

Specifies properties of the minor tick marks on the axis.



Type

Specifies the type of the minor tick marks on the axis.

- **None**
  If selected, minor tick marks will not be shown on the axis and it will be meaningless to specify all the other minor tick mark related properties.
- **Inside**
  If selected, minor tick marks will be inside the chart.
- **Outside**
  If selected, minor tick marks will be outside the chart.
- **Cross**
  If selected, minor tick marks will be across the axis.

Line

Specifies the line properties of the minor tick marks on the axis.

- **Correlate with Axis**
  If checked, the line properties of the minor tick marks will correlate with that of the axis automatically.
  - **Color**
    Specifies the color of the minor tick marks.

    To change the color, click the color image button to display the color palette in which you can select a color, or click **More Colors** to customize the color within a wider range in the Pick a Color dialog. You can also input the hexadecimal value of a color directly in the text box after #.
  - **Style**
    Specifies the type of the minor tick marks.
  - **Transparency**
    Specifies the transparency for the color of the minor tick marks.
  - **Thickness**
    Specifies the thickness of the minor tick marks, in pixels.
- **Tick Mark Length**
  Specifies the length of the minor tick marks, in pixels.

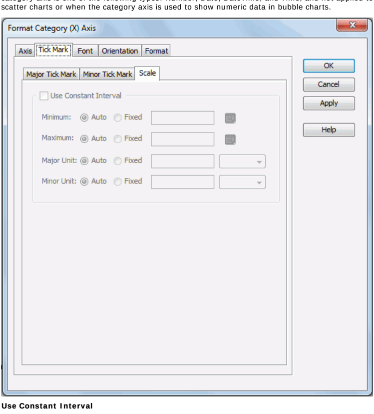
Option

Specifies the other properties of the minor tick marks on the axis.

- **Show Minor Tick Mark Labels**
  Specifies whether to display the labels of the minor tick marks on the axis. If checked, the following properties will be enabled.
- **Label Every N Minor Tick Marks**
  Specifies the frequency at which the minor tick marks will be labeled.
- **Number of Minor Labels**
  Specifies how many minor tick mark labels to be displayed on the axis.
  - **Auto**
    If checked, all minor tick mark labels will be shown.
  - **Fixed**
    If checked, you can specify the number of the minor tick mark labels to be displayed on the axis.

Scale tab

Customizes the way in which to label the tick marks on the axis. Available only when the field on the category axis is one of the following types: Number, Date, DateTime, and Time, and not applied to scatter charts or when the category axis is used to show numeric data in bubble charts.



Use Constant Interval

Specifies whether to use a constant interval to label the tick marks. If checked, the values of the tick marks will be increased continually on the axis based on the following properties, instead of just using the data values.

Minimum

Specifies the minimum value which will be used to label the tick marks.

- **Auto**
  If checked, the minimum value will be defined by JReport automatically.
- **Fixed**
  If checked, you can define the minimum value as required. Input the value in the text box, or specify it in the calendar if the field on the category axis is of Date, DateTime or Time type.

Maximum

Specifies the maximum value which will be used to label the tick marks.

- **Auto**
  If checked, the maximum value will be defined by JReport automatically.
- **Fixed**
  If checked, you can define the maximum value as required. Input the value in the text box, or specify it in the calendar if the field on the category axis is of Date, DateTime or Time type.

Major Unit

Specifies the unit between two adjacent major tick marks.

- **Auto**
  If checked, the unit will be defined by JReport automatically.
- **Fixed**
  If checked, you can define the unit as required. Input the value in the text box, or choose the desired one from the drop-down list if the field on the category axis is of Date, DateTime or Time type.

Minor Unit

Specifies the unit between two adjacent minor tick marks.

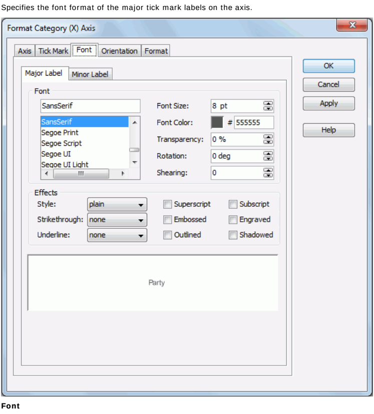
- **Auto**
  If checked, the unit will be defined by JReport automatically.
- **Fixed**
  If checked, you can define the unit as required. Input the value in the text box, or choose the desired one from the drop-down list if the field on the category axis is of the Date, DateTime or Time type.

Font tab

The tab consists of two sub tabs: major label tab and minor label tab.

Major Label tab

Specifies the font format of the major tick mark labels on the axis.



Font

Specifies the font format of text in the major tick mark labels.

- **Font list**
  Lists all the available font faces that can be selected to apply to the text.
- **Font Size**
  Specifies the font size of the text.
- **Font Color**
  Specifies the font color of the text.

  To change the color, click the color image button to display the color palette in which you can select a color, or click **More Colors** to customize the color within a wider range in the Pick a Color dialog, or click **More Fill Effects** to customize in the Fill Effects dialog. You can also input the hexadecimal value of a color directly in the text box after #.
- **Transparency**
  Specifies the transparency of the text.
- **Rotation**
  Specifies the rotation angle of the text around its center, in degrees. The default value is 0.
- **Shearing**
  Specifies the gradient of the text.

Effects

Specifies the special effects of text in the major tick mark labels.

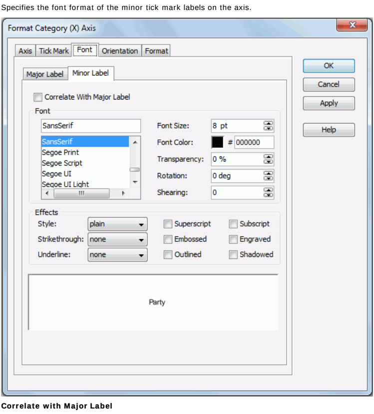
- **Style**
  Specifies the font style of the text. It can be one of the following: plain, bold, italic, and bold italic.
- **Strikethrough**
  Specifies the style of the horizontal line with which the text is struck through. It can be one of the following: none, thin line, bold line, and double lines.
- **Underline**
  Specifies the style of the horizontal line under the text. It can be one of the following: none, single, single lower, bold line, bold lower, double lines, bold double, patterned line, and bold patterned.
- **Superscript**
  Raises the text above the baseline and changes the text to a smaller font size, if a smaller size is available.
- **Embossed**
  Makes the text appear to be raised off the page in relief.
- **Outlined**
  Displays the inner and outer borders of each character.
- **Subscript**
  Lowers the text below the baseline and changes the text to a smaller font size, if a smaller size is available.
- **Engraved**
  Makes the text appear to be imprinted or pressed into the page.
- **Shadowed**
  Adds a shadow beneath and to the right of the text.

Sample

Displays a sample of the specified font and any text effects.

Minor Label tab

Specifies the font format of the minor tick mark labels on the axis.



Correlate with Major Label

If checked, the font format of the minor tick mark labels will correlate with that of the major tick mark labels automatically. Only when it is unchecked can the font properties of the minor tick mark labels take effect.

Font

Specifies the font format of text in the minor tick mark labels.

- **Font list**
  Lists all the available font faces that can be selected to apply to the text.
- **Font Size**
  Specifies the font size of the text.
- **Font Color**
  Specifies the font color of the text.

  To change the color, click the color image button to display the color palette in which you can select a color, or click **More Colors** to customize the color within a wider range in the Pick a Color dialog, or click **More Fill Effects** to customize in the Fill Effects dialog. You can also input the hexadecimal value of a color directly in the text box after #.
- **Transparency**
  Specifies the transparency of the text.
- **Rotation**
  Specifies the rotation angle of the text around its center, in degrees. The default value is 0.
- **Shearing**
  Specifies the gradient of the text.

Effects

Specifies the special effects of text in the minor tick mark labels.

- **Style**
  Specifies the font style of the text. It can be one of the following: plain, bold, italic, and bold italic.
- **Strikethrough**
  Specifies the style of the horizontal line with which the text is struck through. It can be one of the following: none, thin line, bold line, and double lines.
- **Underline**
  Specifies the style of the horizontal line under the text. It can be one of the following: none, single, single lower, bold line, bold lower, double lines, bold double, patterned line, and bold patterned.
- **Superscript**
  Raises the text above the baseline and changes the text to a smaller font size, if a smaller size is available.
- **Embossed**
  Makes the text appear to be raised off the page in relief.
- **Outlined**
  Displays the inner and outer borders of each character.
- **Subscript**
  Lowers the text below the baseline and changes the text to a smaller font size, if a smaller size is available.
- **Engraved**
  Makes the text appear to be imprinted or pressed into the page.
- **Shadowed**
  Adds a shadow beneath and to the right of the text.

Sample

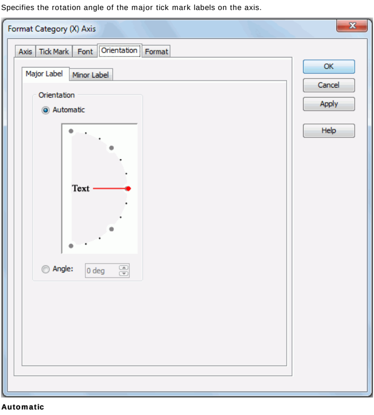
Displays a sample of the specified font and any text effects.

Orientation tab

The tab consists of two sub tabs: major label tab and minor label tab.

Major Label tab

Specifies the rotation angle of the major tick mark labels on the axis.



Automatic

Specifies to adjust the rotation angle of the major tick mark label text on the axis automatically according to the length of the label text, in degrees.

When this option is checked by default:

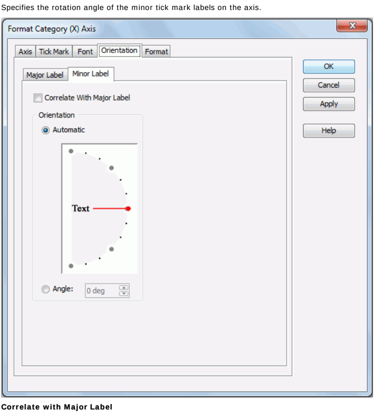
- If the text can be completely displayed horizontally, the default rotation angle will be 0.
- If the text cannot be completely displayed horizontally, the default rotation angle will be 30 anticlockwise, and the cut off part will be shown as suspension points.

Angle

Specifies to customize the rotation angle of the major tick mark label text on the axis. When you change the rotation angle in the text box, the red line in the spin box will change correspondingly, and vice versa.

Minor Label tab

Specifies the rotation angle of the minor tick mark labels on the axis.



Correlate with Major Label

If checked, the orientation setting of the minor tick mark labels will correlate with that of the major tick mark labels automatically. Only when it is unchecked can the orientation properties of the minor tick mark labels take effect.

Automatic

Specifies to adjust the rotation angle of the minor tick mark label text on the axis automatically according to the length of the label text, in degrees.

When this option is checked by default:

- If the text can be completely displayed horizontally, the default rotation angle will be 0.
- If the text cannot be completely displayed horizontally, the default rotation angle will be 30 anticlockwise, and the cut off part will be shown as suspension points.

Angle

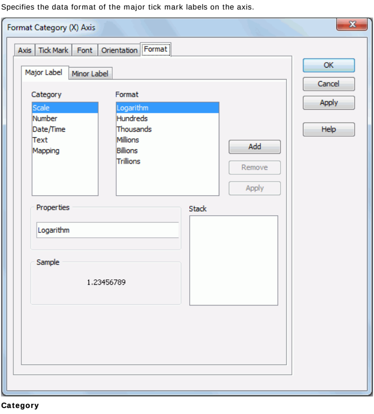
Specifies to customize the rotation angle of the minor tick mark label text on the axis. When you change the rotation angle in the text box, the red line in the spin box will change correspondingly, and vice versa.

Format tab

The tab consists of two sub tabs: major label tab and minor label tab.

Major Label tab

Specifies the data format of the major tick mark labels on the axis.



Category

Lists the category types. Select one to customize its format.

Format

Displays all the formats of the selected category. Select the required one and click Add to add it as the format of the specified category. You can add only one format for each category. For more details about the formats, see A detailed chart property reference.

Properties

Displays the properties of the format you select. If the formats listed in the Format box cannot meet your requirement, define the format in the text field and then click Add to add it as the format of the specified category.

Sample

Displays the selected format effects.

Stack

Lists all the formats you select from different categories.

Add

Adds a format to the Stack list box.

Remove

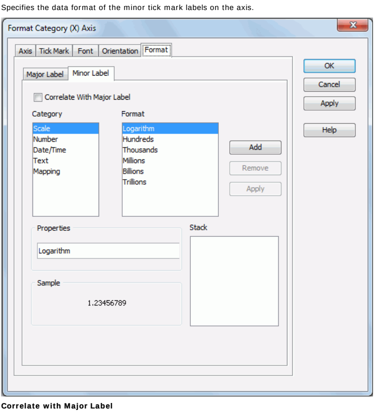
Removes a format from the Stack list box.

Apply

Applies the specified format in the Stack list box to the major tick mark labels.

Minor Label tab

Specifies the data format of the minor tick mark labels on the axis.



Correlate with Major Label

If checked, the data format of the minor tick mark labels will correlate with that of the major tick mark labels automatically. Only when it is unchecked can the format properties of the minor tick mark labels take effect.

Category

Lists the category types. Select one to customize its format.

Format

Displays all the formats of the selected category. Select the required one and click Add to add it as the format of the specified category. You can add only one format for each category. For more details about the formats, see A detailed chart property reference.

Properties

Displays the properties of the format you select. If the formats listed in the Format box cannot meet your requirement, define the format in the text field and then click Add to add it as the format of the specified category.

Sample

Displays the selected format effects.

Stack

Lists all the formats you select from different categories.

Add

Adds a format to the Stack list box.

Remove

Removes a format from the Stack list box.

Apply

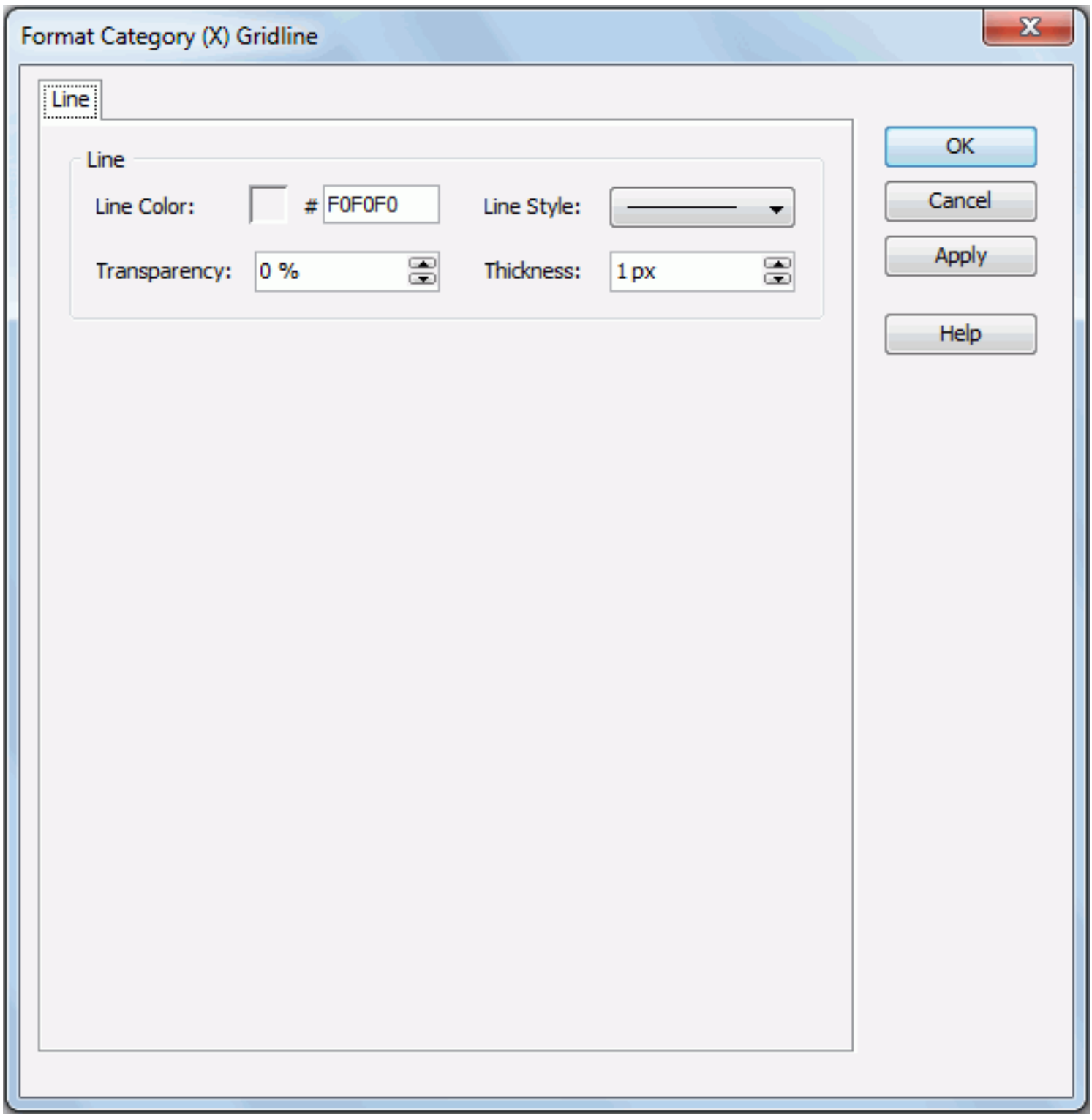
Applies the specified format in the Stack list box to the minor tick mark labels.

Related topics:

- Formatting the category axis

Format Category (X) Gridline dialog

This dialog appears when you double-click a category (X) gridline of a chart, or right-click a chart and then select Format Gridlines > Category (X) Gridline from the shortcut menu. It helps you to format the gridlines shown on the category (X) axis of a chart.



The following are details about options in the dialog:

Line Color

Specifies the color schema of the gridlines.

To change the color, click the color image button to display the color palette in which you can select a color, or click **More Colors** to customize the color within a wider range in the Pick a Color dialog. You can also input the hexadecimal value of a color directly in the text box after #.

Line Style

Specifies the line style of the gridlines.

Transparency

Specifies the transparency of the color that is applied to the gridlines.

Thickness

Specifies the thickness of the gridlines, in pixels.

OK

Applies the changes and closes the dialog.

Cancel

Does not retain any changes and closes the dialog.

Apply

Applies all changes and leaves the dialog open.

Help

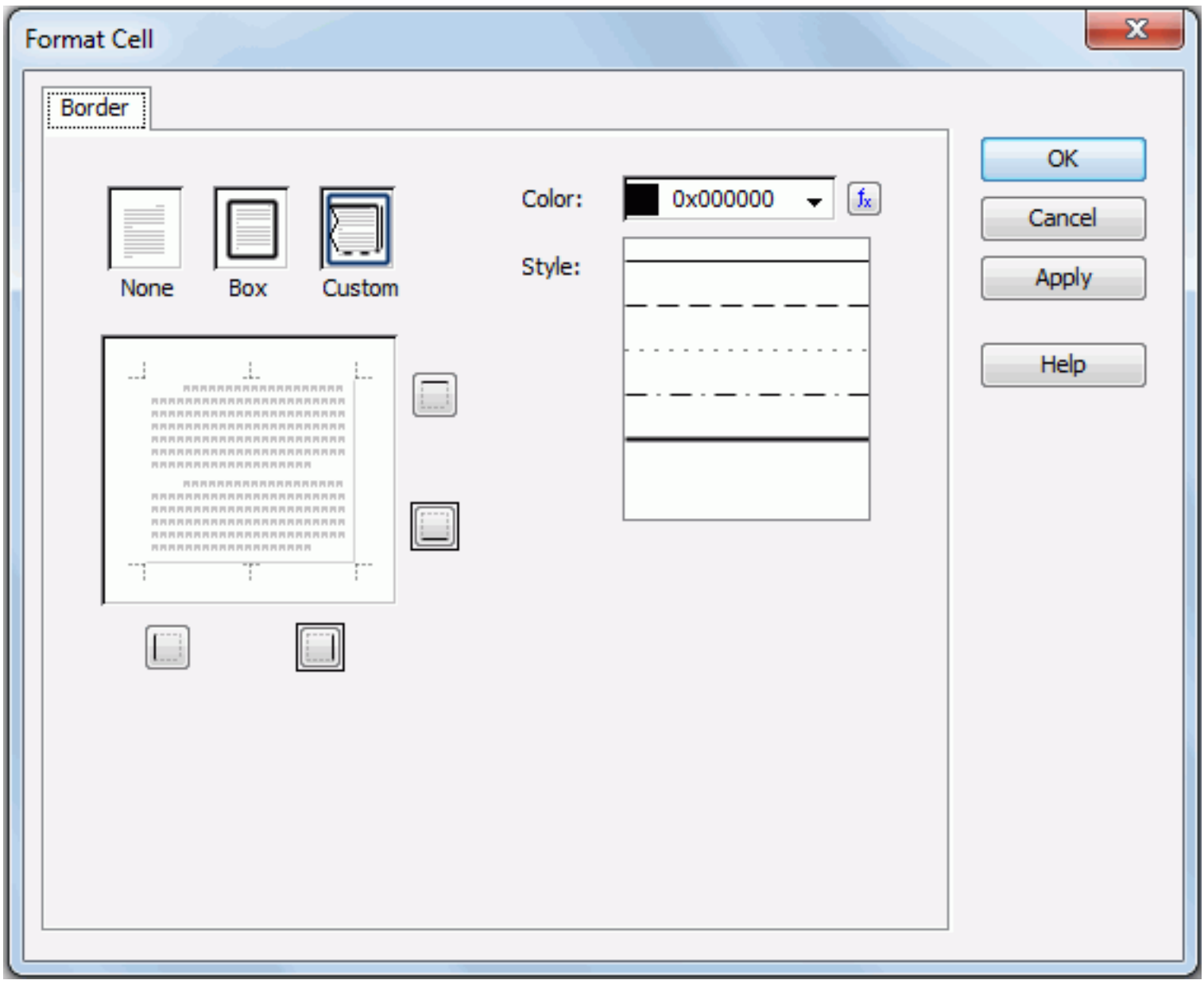
Displays the help document about this feature.

Related topics:

- Formatting the gridlines

Format Cell dialog

This dialog appears when you right-click a table cell and then select Format Border from the shortcut menu. It enables you to specify border for a selected table cell.



The following are details about options in the dialog:

Border

Specifies properties for border of the selected table cell.

- **None**
  If selected, no border of the selected table cell will be displayed.
- **Box**
  If selected, insert a box border around the selected table cell, formatted with the currently selected line style and color settings.
- **Custom**
  If selected, create a custom border and customize every border, by clicking the corresponding border buttons. If you click one of the border buttons, the Custom will be selected automatically.
- **Color**
  Specifies the line color schema for border of the selected table cell. If you want to use a formula to control the line color, click and select a formula from the drop-down list.
- **Style**
  Specifies the line style for border of the selected table cell.

OK

Applies all changes and closes the dialog.

Cancel

Does not retain any changes and closes the dialog.

Apply

Applies all changes, leaving the dialog open.

Help

Displays the help document about this feature.

Related topics:

- Formatting cells in a table

Format Dial Gauge dialog for library component

This dialog appears when you right-click a dial gauge of a chart in a library component and select Format Dial Gauge from the shortcut menu. It helps you to format dial gauges in a chart, and consists of the following tabs:

- General tab
- Fill tab
- Border tab
- Data Label tab
- Behaviors tab

OK

Applies the changes and closes the dialog.

Cancel

Does not retain any changes and closes the dialog.

Apply

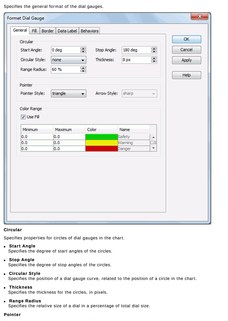
Applies all changes and leaves the dialog open.

Help

Displays the help document about this feature.

General tab

Specifies the general format of the dial gauges.



Circular

Specifies properties for circles of dial gauges in the chart.

- **Start Angle**
  Specifies the degree of start angles of the circles.
- **Stop Angle**
  Specifies the degree of stop angles of the circles.
- **Circular Style**
  Specifies the position of a dial gauge curve, related to the position of a circle in the chart.
- **Thickness**
  Specifies the thickness for the circles, in pixels.
- **Range Radius**
  Specifies the relative size of a dial in a percentage of total dial size.

Pointer

Specifies properties for pointers of dial gauges in the chart.

- **Pointer Style**
  Specifies the style applied to pointers of the dial gauges.
  - **triangle**
    If selected, pointers will be triangle shapes.
  - **diamond**
    If selected, pointers will be diamond shapes.
  - **arrow**
    If selected, pointers will be arrow shapes.
- **Arrow Style**
  Specifies the style applied to arrows of the pointers.
  - **sharp**
    If selected, arrows will be sharp ones.
  - **round**
    If selected, arrows will be round ones.
  - **plain**
    If selected, arrows will be plain ones.

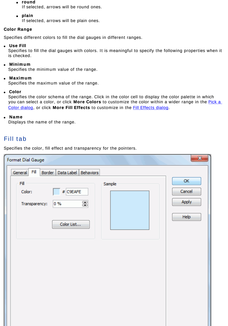
Color Range

Specifies different colors to fill the dial gauges in different ranges.

- **Use Fill**
  Allows to fill the dial gauges with colors. It is meaningful to specify the following properties when it is checked.
- **Minimum**
  Specifies the minimum value of the range.
- **Maximum**
  Specifies the maximum value of the range.
- **Color**
  Specifies the color scheme of the range. Click in the color cell to display the color palette in which you can select a color, or click **More Colors** to customize the color within a wider range in the Pick a Color dialog, or click **More Fill Effects** to customize in the Fill Effects dialog.
- **Name**
  Displays the name of the range.

Fill tab

Specifies the color, fill effect and transparency for the pointers.



Color

Specifies the color scheme for the selected pointers in the same data series.

To change the color, click the color image button to display the color palette in which you can select a color, or click **More Colors** to customize the color within a wider range in the Pick a Color dialog. You can also click **More Fill Effects** to customize in the Fill Effects dialog. You can also input the hexadecimal value of a color directly in the text box after # if you have changed the default color.

Transparency

Specifies the transparency of the color scheme.

Color List

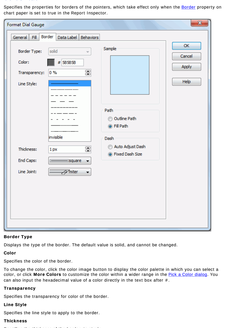
Opens the Color List dialog to modify color pattern for pointers in the same data series.

Sample

Displays a preview sample of your selection.

Border tab

Specifies the properties for borders of the pointers, which take effect only when the Border property on chart paper is set to true in the Report Inspector.



Border Type

Displays the type of the border. The default value is solid, and cannot be changed.

Color

Specifies the color of the border.

To change the color, click the color image button to display the color palette in which you can select a color, or click **More Colors** to customize the color within a wider range in the Pick a Color dialog. You can also input the hexadecimal value of a color directly in the text box after #.

Transparency

Specifies the transparency for color of the border.

Line Style

Specifies the line style to apply to the border.

Thickness

Specifies the line thickness of the border, in pixels.

End Caps

Specifies the ending style of the border line.

- **butt**
  Ends unclosed sub paths and dash segments with no added decoration.
- **round**
  Ends unclosed sub paths and dash segments with a round decoration that has a radius equal to half of the width of the pen.
- **square**
  Ends unclosed sub paths and dash segments with a square projection that extends beyond the end of the segment to a distance equal to half of the line width.

Line Joint

Specifies the line joint style for the border line.

- **miter**
  Joins path segments by extending their outside edges until they meet.
- **round**
  Joins path segments by rounding off the corner at a radius of half the line width.
- **bevel**
  Joins path segments by connecting the outer corners of their wide outlines with a straight segment.

Sample

Displays a preview sample of your selection.

Path

Specifies the fill pattern of the border line.

- **Outline Path**
  Specifies the fill pattern of the border line to be outline path.
- **Fill Path**
  Specifies the fill pattern of the border line to be whole path.

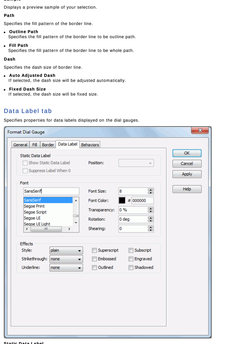
Dash

Specifies the dash size of border line.

- **Auto Adjusted Dash**
  If selected, the dash size will be adjusted automatically.
- **Fixed Dash Size**
  If selected, the dash size will be fixed size.

Data Label tab

Specifies properties for data labels displayed on the dial gauges.



Static Data Label

Specifies properties of the static data labels. Not supported on gauge chart.

Font

Specifies the font format of text in the data labels.

- **Font list**
  Lists all the available font faces that can be selected to apply to the text.
- **Font Size**
  Specifies the font size of the text.
- **Font Color**
  Specifies the font color of the text.

To change the color, click the color image button to display the color palette in which you can select a color, or click **More Colors** to customize the color within a wider range in the Pick a Color dialog, or click **More Fill Effects** to customize in the Fill Effects dialog. You can also input the hexadecimal value of a color directly in the text box after #.

- **Transparency**
  Specifies the transparency of the text.
- **Rotation**
  Specifies the rotation angle of the text around its center, in degrees. The default value is 0.
- **Shearing**
  Specifies the gradient of the text.

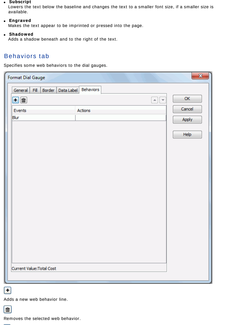
Effects

Specifies the special effects of text in the data labels.

- **Style**
  Specifies the font style of the text. It can be one of the following: plain, bold, italic, and bold italic.
- **Strikethrough**
  Specifies the style of the horizontal line which the text is struck through. It can be one of the following: none, single line, double line, and bold line.
- **Underline**
  Specifies the style of the horizontal line under the text. It can be one of the following: none, single line, single lower, bold line, bold lower, double lines, bold double, patterned line, and bold patterned.
- **Superscript**
  Makes the text above the baseline and changes the text to a smaller font size. If a smaller size is available.
- **Embossed**
  Makes the text appear to be raised off the page in relief.
- **Outlined**
  Displays the inner and outer borders of each character.
- **Subscript**
  Lowers the text below the baseline and changes the text to a smaller font size. If a smaller size is available.
- **Engraved**
  Makes the text appear to be imprinted or pressed into the page.
- **Shadowed**
  Adds a shadow beneath and to the right of the text.

Behaviors tab

Specifies some web behaviors to the dial gauges.



Adds a new web behavior line.

Removes the selected web behavior.

Moves the selected web behavior up a step.

Moves the selected web behavior down a step.

Events

Specifies the trigger event.

Actions

Specifies the action you want the event to trigger.

Opens the Web Action List dialog to bind a web action to the event.

Related topics:

- Formatting the gauge

Format Dial Gauge dialog for report

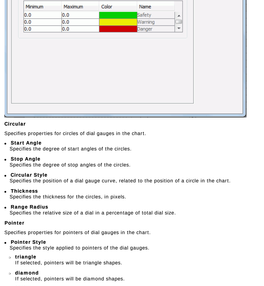
This dialog appears when you right-click a dial gauge of a chart in a page report or web report and select Format Dial Gauge from the shortcut menu. It helps you to format dial gauges in a chart, and consists of the following tabs:

- General tab
- Fill tab
- Border tab
- Data Label tab

OK
Applies the changes and closes the dialog.
Cancel
Does not retain any changes and closes the dialog.
Apply
Applies all changes and leaves the dialog open.
Help
Displays the help document about this feature.

General tab

Specifies the general format of the dial gauges.



Circular
Specifies properties for circles of dial gauges in the chart.

- **Start Angle**
  Specifies the degree of start angles of the circles.
- **Stop Angle**
  Specifies the degree of stop angles of the circles.
- **Circular Style**
  Specifies the position of a dial gauge curve, related to the position of a circle in the chart.
- **Thickness**
  Specifies the thickness for the circles, in pixels.
- **Range Radius**
  Specifies the relative size of a dial in a percentage of total dial size.

Pointer
Specifies properties for pointers of dial gauges in the chart.

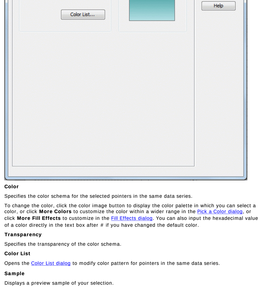
- **Pointer Style**
  Specifies the style applied to pointers of the dial gauges.
  - **triangle**
    If selected, pointers will be triangle shapes.
  - **diamond**
    If selected, pointers will be diamond shapes.
  - **arrow**
    If selected, pointers will be arrow shapes.
    - **Arrow Style**
      Specifies the style applied to arrows of the pointers.
      - **sharp**
        If selected, arrows will be sharp ones.
      - **round**
        If selected, arrows will be round ones.
      - **plain**
        If selected, arrows will be plain ones.

Color Range
Specifies different colors to fill the dial gauges in different ranges.

- **Use Fill**
  Specifies to fill the dial gauges with colors. It is meaningful to specify the following properties when it is checked.
- **Minimum**
  Specifies the minimum value of the range.
- **Maximum**
  Specifies the maximum value of the range.
- **Color**
  Specifies the color schema of the range. Click in the color cell to display the color palette in which you can select a color, or click **More Colors** to customize the color within a wider range in the Pick a Color dialog, or click **More Fill Effects** to customize in the Fill Effects dialog.
- **Name**
  Displays the name of the range.

Fill tab

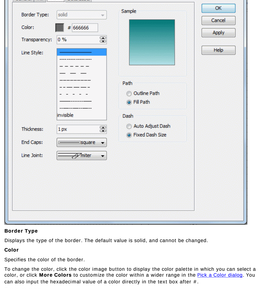
Specifies the color, fill effect and transparency for the pointers.



Color
Specifies the color schema for the selected pointers in the same data series.
To change the color, click the color image button to display the color palette in which you can select a color, or click **More Colors** to customize the color within a wider range in the Pick a Color dialog, or click **More Fill Effects** to customize in the Fill Effects dialog. You can also input the hexadecimal value of a color directly in the text box after **#**. If you have changed the default color.
Transparency
Specifies the transparency of the color schema.
Color List
Opens the Color List dialog to modify color pattern for pointers in the same data series.
Sample
Displays a preview sample of your selection.

Border tab

Specifies the properties for borders of the pointers, which take effect only when the Border property on chart paper is set to true in the Report Inspector.



Border Type
Displays the type of the border. The default value is solid, and cannot be changed.
Color
Specifies the color of the border.
To change the color, click the color image button to display the color palette in which you can select a color, or click **More Colors** to customize the color within a wider range in the Pick a Color dialog. You can also input the hexadecimal value of a color directly in the text box after **#**.
Transparency
Specifies the transparency for color of the border.
Line Style
Specifies the line style in the border.
Thickness
Specifies the thickness of the border, in pixels.
End Caps
Specifies the ending style of the border line.

- **butt**
  Ends unclosed sub paths and dash segments with no added decoration.
- **round**
  Ends unclosed sub paths and dash segments with a round decoration that has a radius equal to half of the width of the pen.
- **square**
  Ends unclosed sub paths and dash segments with a square projection that extends beyond the end of the segment to a distance equal to half of the line width.

Line Joint
Specifies the line joint style for the border line.

- **miter**
  Joins path segments by extending their outside edges until they meet.
- **round**
  Joins path segments by rounding off the corner at a radius of half the line width.
- **bevel**
  Joins path segments by connecting the outer corners of their wide outlines with a straight segment.

Sample
Displays a preview sample of your selection.
Path
Specifies the fill pattern of the border line.

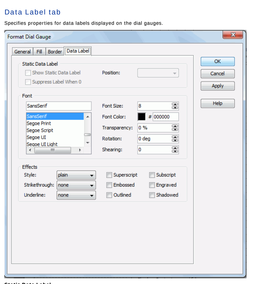
- **Outline Path**
  Specifies the fill pattern of the border line to be outline path.
- **Fill Path**
  Specifies the fill pattern of the border line to be whole path.

Dash
Specifies the dash size of border line.

- **Auto Adjusted Dash**
  If selected, the dash size will be adjusted automatically.
- **Fixed Dash Size**
  If selected, the dash size will be fixed size.

Data Label tab

Specifies the properties for data labels displayed on the dial gauges.



Static Data Label
Specifies properties of the static data labels. Not supported on gauge chart.
Font
Specifies the font format of text in the data labels.

- **Font list**
  Lists all the available font faces that can be selected to apply to the text.
- **Font Size**
  Specifies the font size of the text.
- **Font Color**
  Specifies the font color of the text.
  To change the color, click the color image button to display the color palette in which you can select a color, or click **More Colors** to customize the color within a wider range in the Pick a Color dialog, or click **More Fill Effects** to customize in the Fill Effects dialog. You can also input the hexadecimal value of a color directly in the text box after **#**.
- **Transparency**
  Specifies the transparency of the text.
- **Rotation**
  Specifies the rotation angle of the text around its center, in degrees. The default value is 0.
- **Shearing**
  Specifies the gradient of the text.

Effects
Specifies the special effects of text in the data labels.

- **Style**
  Specifies the font style of the text. It can be one of the following: plain, bold, italic, and bold italic.
- **Strikethrough**
  Specifies the style of the horizontal line with which the text is struck through. It can be one of the following: none, thin line, bold line, and double lines.
- **Underline**
  Specifies the style of the horizontal line under the text. It can be one of the following: none, single, single lower, bold line, bold lower, double lines, bold double, patterned line, and bold patterned.
- **Superscript**
  Raises the text above the baseline and changes the text to a smaller font size, if a smaller size is available.
- **Embossed**
  Makes the text appear to be raised off the page in relief.
- **Outlined**
  Displays the inner and outer borders of each character.
- **Subscript**
  Lowers the text below the baseline and changes the text to a smaller font size, if a smaller size is available.
- **Engraved**
  Makes the text appear to be imprinted or pressed into the page.
- **Shadowed**
  Adds a shadow beneath and to the right of the text.

Related topics:

- Formatting the gauge

Format Donut dialog for library component

This dialog appears when you double-click a donut of a chart in a library component, or right-click it and select Format Donut from the shortcut menu. It helps you to format donuts of a chart, and consists of the following tabs:

- General tab
- Fill tab
- Border tab
- Data Label tab
- Behaviors tab

OK

Accepts the formatting of the donuts and closes the dialog.

Cancel

Cancels the formatting of the donuts and closes the dialog.

Apply

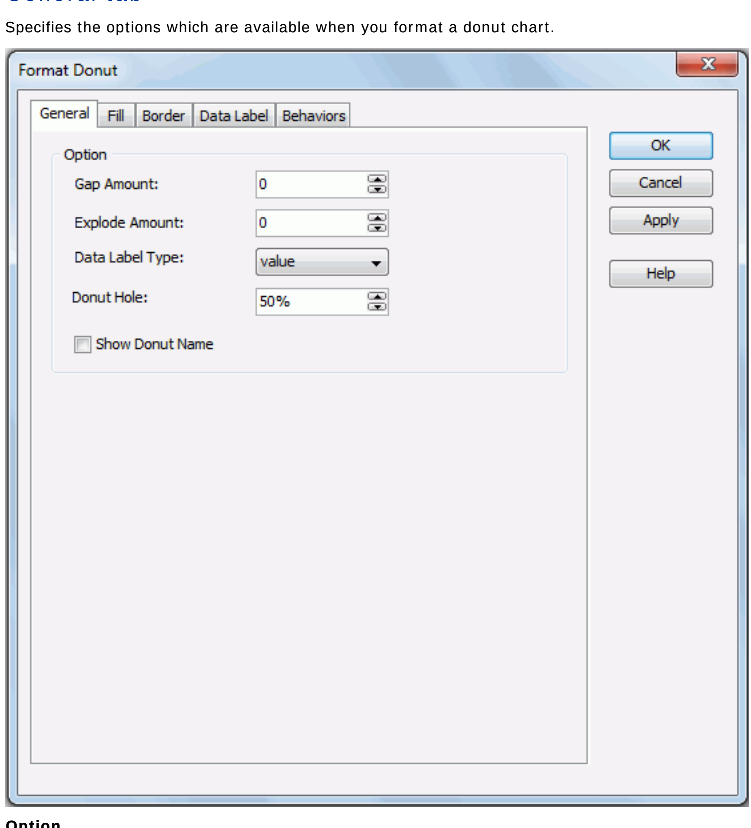
Accepts the formatting of the donuts and leaves the dialog open.

Help

Displays the help document about this feature.

General tab

Specifies the options which are available when you format a donut chart.

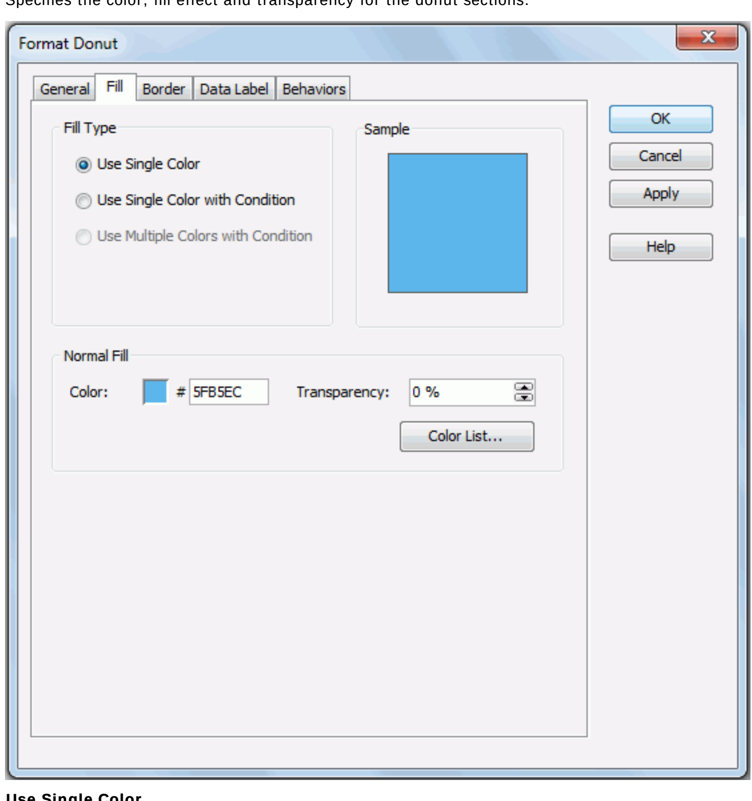


Option

- **Gap Amount**
  Specifies the gap between two donuts.
- **Explode Amount**
  Specifies the gap between every section and the central point of a donut.
- **Data Label Type**
  Specifies the display type for data value in the data label. It can be value, category name, percent, value and percent.
- **Donut Hole**
  Specifies the radius percentage of the donut hole to the total pie circle.
- **Show Donut Name**
  Specifies whether to show the name of the donut.

Fill tab

Specifies the color, fill effect and transparency for the donut sections.



Use Single Color

If checked, the donut sections will use the single color pattern.

- **Color**
  Specifies the color schema for the selected donut sections in the same data series.

  To change the color, click the color image button to display the color palette in which you can select a color, or click **More Colors** to customize the color within a wider range in the Pick a Color dialog, or click **More Fill Effects** to customize in the Fill Effects dialog. You can also input the hexadecimal value of a color directly in the text box after # if you have changed the default color.
- **Transparency**
  Specifies the transparency of the color schema.
- **Color List**
  Opens the Color List dialog to modify color pattern for donut sections in the same data series.

Use Single Color with Condition

If checked, each donut section can have its single color pattern based on defined conditions. You can click the Advanced or Normal button to switch between the two editing modes to edit the conditions.

- **Normal**
  - **Select Field**
    Lists all the available fields to which the conditional fill can be applied. Select the field on which you want to define the conditions from the drop-down list.
  - **Values**
    Specifies the value of the condition. Available when you select a category or series field or a field whose return value is not a number from the Select Field drop-down list.
  - **Start Value**
    Specifies the start value of the condition. Available only when you select a value field or a field that returns a number from the Select Field drop-down list.
  - **End Value**
    Specifies the end value of the condition. Available only when you select a value field or a field that returns a number from the Select Field drop-down list.
  - **Color**
    Specifies the color that will be applied to the values which meet the condition.

    To change the color, click the color image button to display the color palette in which you can select a color, or click **More Colors** to customize the color within a wider range in the Pick a Color dialog, or click **More Fill Effects** to customize in the Fill Effects dialog. You can also input the hexadecimal value of a color directly in the text box after #.
  - **Transparency**
    Specifies the transparency for the color of the condition.
  - Adds a new condition line.
  - Removes the selected condition line.
  - **Label**
    If checked, you can modify the condition expression of the selected condition line which will be shown as the legend entry label.
  - **Value**
    Specifies to display data in the condition expression as value. Available only when you select a value field or a field that returns a number from the Select Field drop-down list.
  - **Percent**
    Specifies to display data in the condition expression in percent. Available only when you select a value field or a field that returns a number from the Select Field drop-down list and takes effect only on 100% Stacked Bar/Bench 2-D chart and 100% Stacked Bar/Bench 3-D chart.
- **Advanced**
  - **Condition**
    Displays the condition you have defined in the Edit Condition dialog.
  - **Color**
    Specifies the color that will be applied to the values which meet the condition.

    To change the color, click the color image button to display the color palette in which you can select a color, or click **More Colors** to customize the color within a wider range in the Pick a Color dialog, or click **More Fill Effects** to customize in the Fill Effects dialog. You can also input the hexadecimal value of a color directly in the text box after #.
  - **Transparency**
    Specifies the transparency for the color of the condition.
  - Opens the Edit Conditions dialog to create or edit a condition.
  - **Label**
    If checked, you can modify the condition expression of the selected condition line which will be shown as the legend entry label.
  - **Value**
    Specifies to display data in the condition expression as value. Available only when you select a value field or a field that returns a number from the Select Field drop-down list.
  - **Percent**
    Specifies to display data in the condition expression in percent. Available only when you select a value field or a field that returns a number from the Select Field drop-down list.

Use Multiple Colors with Condition

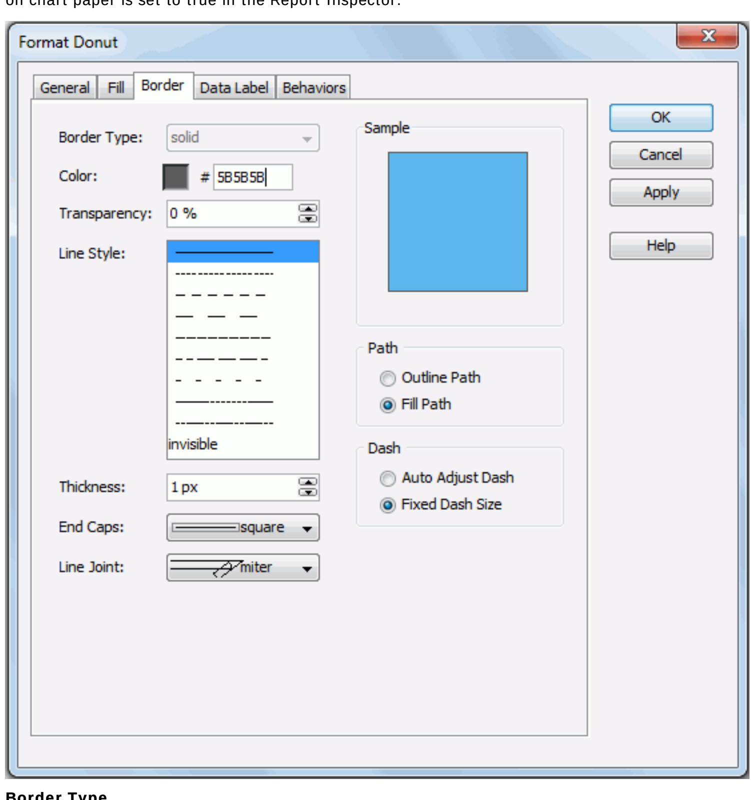
Not supported on donut chart.

Sample

Displays a preview sample of your selection.

Border tab

Specifies the properties for the border of the donuts, which take effect only when the Border property on chart paper is set to true in the Report Inspector.



Border Type

Displays the type for border of the donuts. The default value is solid, and cannot be changed.

Color

Specifies the color for the border of the donuts.

To change the color, click the color image button to display the color palette in which you can select a color, or click **More Colors** to customize the color within a wider range in the Pick a Color dialog. You can also input the hexadecimal value of a color directly in the text box after #.

Transparency

Specifies the transparency for color of the border.

Line Style

Specifies the line style to apply to the border of the donuts.

Thickness

Specifies the thickness of the border, in pixels.

End Caps

Specifies the ending style of the border line.

- **butt**
  Ends unclosed sub paths and dash segments with no added decoration.
- **round**
  Ends unclosed sub paths and dash segments with a round decoration that has a radius equal to half of the width of the pen.
- **square**
  Ends unclosed sub paths and dash segments with a square projection that extends beyond the end of the segment to a distance equal to half of the line width.

Line Joint

Specifies the line joint style for the border line.

- **miter**
  Joins path segments by extending their outside edges until they meet.
- **round**
  Joins path segments by rounding off the corner at a radius of half the line width.
- **bevel**
  Joins path segments by connecting the outer corners of their wide outlines with a straight segment.

Sample

Displays a preview sample of your selection.

Path

Specifies the fill pattern of the border line.

- **Outline Path**
  Specifies the fill pattern of the border line to be outline path.
- **Fill Path**
  Specifies the fill pattern of the border line to be whole path.

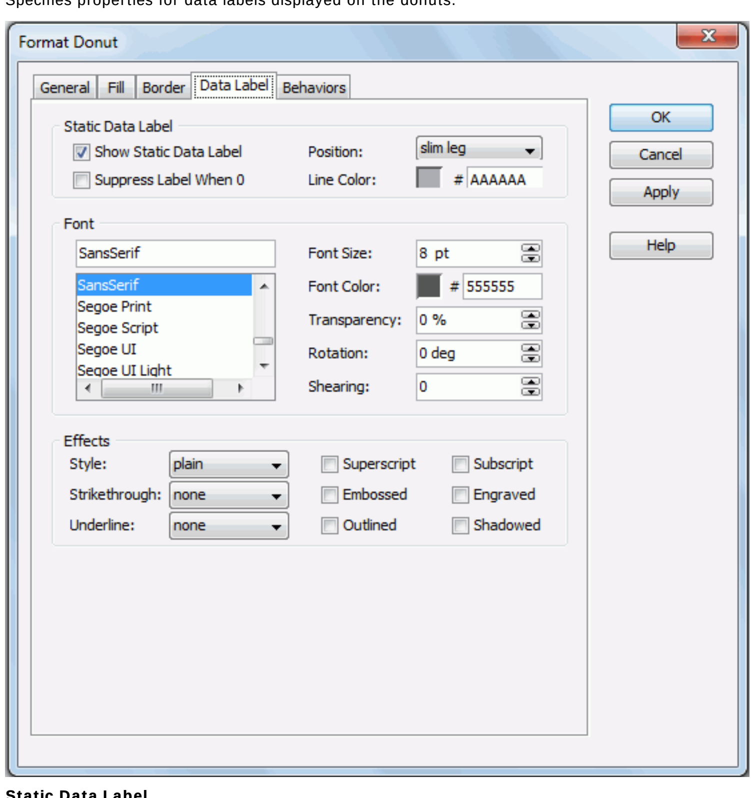
Dash

Specifies the dash size of border line.

- **Auto Adjusted Dash**
  If selected, the dash size will be adjusted automatically.
- **Fixed Dash Size**
  If selected, the dash size will be fixed size.

Data Label tab

Specifies properties for data labels displayed on the donuts.



Static Data Label

Specifies properties of the static data labels on the donuts.

- **Show Static Data Label**
  Specifies whether or not to show the static data labels on the donuts. Only when it is checked can the following static data label related properties take effect.
- **Position**
  Specifies the position of the static data labels on the donuts.
  - **auto fit**
    If selected, the static data labels will be displayed automatically.
  - **sticker**
    If selected, the static data labels will be displayed beside the donut sections.
  - **slim leg**
    If selected, the static data labels will be displayed beside the donut sections and pointed by thin lines.
  - **best fit**
    If selected, the static data labels will be displayed at the best fit position automatically.
  - **on slices**
    If selected, the static data labels will be displayed on the donut sections (slices).
- **Suppress Label When 0**
  If true, the static data label whose value is 0 will not be displayed on the chart.
- **Line Color**
  Specifies the color of the thin lines that are used to point to the static data labels.

  To change the color, click the color image button to display the color palette in which you can select a color, or click **More Colors** to customize the color within a wider range in the Pick a Color dialog. You can also input the hexadecimal value of a color directly in the text box after #.

Font

Specifies the font format of text in the data labels.

- **Font list**
  Lists all the available font faces that can be selected to apply to the text.
- **Font Size**
  Specifies the font size of the text.
- **Font Color**
  Specifies the font color of the text.

  To change the color, click the color image button to display the color palette in which you can select a color, or click **More Colors** to customize the color within a wider range in the Pick a Color dialog, or click **More Fill Effects** to customize in the Fill Effects dialog. You can also input the hexadecimal value of a color directly in the text box after #.
- **Transparency**
  Specifies the transparency of the text.
- **Rotation**
  Specifies the rotation angle of the text around its center, in degrees. The default value is 0.
- **Shearing**
  Specifies the gradient of the text.

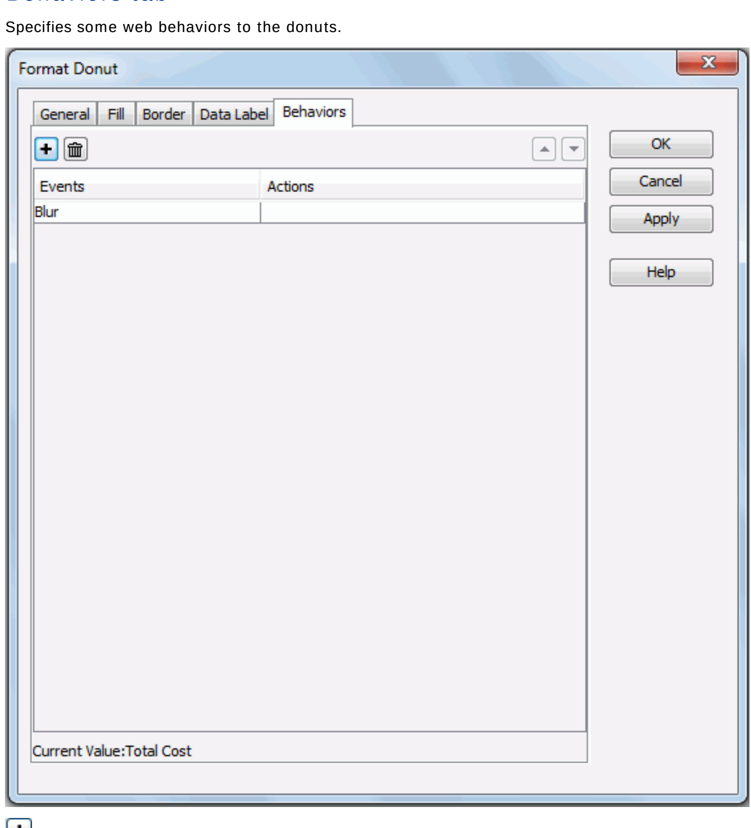
Effects

Specifies the special effects of text in the data labels.

- **Style**
  Specifies the font style of the text. It can be one of the following: plain, bold, italic, and bold italic.
- **Strikethrough**
  Specifies the style of the horizontal line with which the text is struck through. It can be one of the following: none, thin line, bold line, and double lines.
- **Underline**
  Specifies the style of the horizontal line under the text. It can be one of the following: none, single, single lower, bold line, bold lower, double lines, bold double, patterned line, and bold patterned.
- **Superscript**
  Raises the text above the baseline and changes the text to a smaller font size, if a smaller size is available.
- **Embossed**
  Makes the text appear to be raised off the page in relief.
- **Outlined**
  Displays the inner and outer borders of each character.
- **Subscript**
  Lowers the text below the baseline and changes the text to a smaller font size, if a smaller size is available.
- **Engraved**
  Makes the text appear to be imprinted or pressed into the page.
- **Shadowed**
  Adds a shadow beneath and to the right of the text.

Behaviors tab

Specifies some web behaviors to the donuts.



Adds a new web behavior line.

Removes the selected web behavior.

Moves the selected web behavior up a step.

Moves the selected web behavior down a step.

Events

Specifies the trigger event.

Actions

Specifies the action you want the event to trigger.

- Opens the Web Action List dialog to bind a web action to the event.

Related topics:

- Formatting the pies/donuts

Format Donut dialog for report

This dialog appears when you double-click a donut of a chart in a page report or web report, or right-click a donut and then select Format from the shortcut menu. It helps you to format donuts of a chart, and consists of the following tabs:

- General tab
- Fill tab
- Border tab
- Data Label tab

OK
Accepts the formatting of the donuts and closes the dialog.

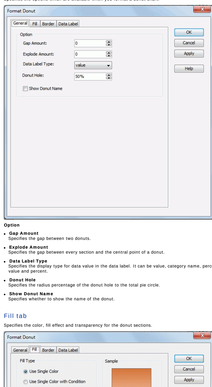
Cancel
Cancels the formatting of the donuts and closes the dialog.

Apply
Accepts the formatting of the donuts and leaves the dialog open.

Help
Displays the help document about this feature.

General tab

Specifies the options which are available when you format a donut chart.

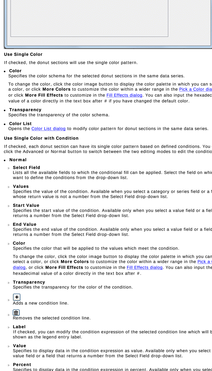


Option

- **Gap Amount**
  Specifies the gap between two donuts.
- **Explode Amount**
  Specifies the gap between every section and the central point of a donut.
- **Type Label Type**
  Specifies the display type for data value in the data label. It can be value, category name, percent, value and percent.
- **Donut Hole**
  Specifies the radius percentage of the donut hole to the total pie circle.
- **Show Donut Name**
  Specifies whether to show the name of the donut.

Fill tab

Specifies the color, fill effect and transparency for the donut sections.



Use Single Color
If checked, the donut sections will use the single color pattern.

- **Color**
  Specifies the color scheme for the selected donut sections in the same data series.

  To change the color, click the color image button to display the color palette in which you can select a color, or click **More Colors** to customize the color within a wider range in the Pick a Color dialog. Or click **More Fill Effects** to customize in the Fill Effects dialog. You can also input the hexadecimal value of a color directly in the text box after # if you have changed the default color.
- **Transparency**
  Specifies the transparency of the color scheme.
- **Color List**
  Opens the Color List dialog to modify color pattern for donut sections in the same data series.

Use Single Color with Condition
If checked, each donut section can have its single color pattern based on defined conditions. You can click the Advanced or Normal button to switch between the two editing modes to edit the conditions.

- **Normal**
  - **Select Field**
    Lists all the available fields to which the conditional fill can be applied. Select the field on which you want to define the conditions from the drop-down list.
  - **Color**
    Specifies the value of the condition. Available when you select a category or series field or a field whose return value is not a number from the Select Field drop-down list.
  - **Start Value**
    Specifies the start value of the condition. Available only when you select a value field or a field that returns a number from the Select Field drop-down list.
  - **End Value**
    Specifies the end value of the condition. Available only when you select a value field or a field that returns a number from the Select Field drop-down list.
  - **Color**
    Specifies the color that will be applied to the values which meet the condition.

    To change the color, click the color image button to display the color palette in which you can select a color, or click **More Colors** to customize the color within a wider range in the Pick a Color dialog, or click **More Fill Effects** to customize in the Fill Effects dialog. You can also input the hexadecimal value of a color directly in the text box after #.
  - **Transparency**
    Specifies the transparency for the color of the condition.
  - +
    Adds a new condition line.
  - -
    Removes the selected condition line.
  - **Label**
    If checked, you can modify the condition expression of the selected condition line which will be shown as the legend entry label.
  - **Value**
    Specifies to display data in the condition expression as value. Available only when you select a value field or a field that returns a number from the Select Field drop-down list.
  - **Percent**
    Specifies to display data in the condition expression in percent. Available only when you select a value field or a field that returns a number from the Select Field drop-down list.
- **Advanced**
  - **Condition**
    Specifies the condition you have defined in the Edit Condition dialog.
  - **Color**
    Specifies the color that will be applied to the values which meet the condition.

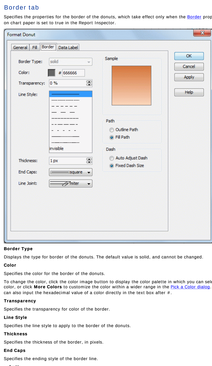
    To change the color, click the color image button to display the color palette in which you can select a color, or click **More Colors** to customize the color within a wider range in the Pick a Color dialog, or click **More Fill Effects** to customize in the Fill Effects dialog. You can also input the hexadecimal value of a color directly in the text box after #.
  - **Transparency**
    Specifies the transparency for the color of the condition.
  - Edit...
    Opens the Edit Conditions dialog to create or edit a condition.
  - **Label**
    If checked, you can modify the condition expression of the selected condition line which will be shown as the legend entry label.
  - **Value**
    Specifies to display data in the condition expression as value. Available only when you select a value field or a field that returns a number from the Select Field drop-down list.
  - **Percent**
    Specifies to display data in the condition expression in percent. Available only when you select a value field or a field that returns a number from the Select Field drop-down list.

Use Multiple Colors with Condition
Not supported on donut chart.

Sample
Displays a preview sample of your selection.

Border tab

Specifies the properties for the border of the donuts, which take effect only when the Border property on chart paper is set to true in the Report Inspector.



Border Type
Displays the type for border of the donuts. The default value is solid, and cannot be changed.

Color
Specifies the color for the border of the donuts.

To change the color, click the color image button to display the color palette in which you can select a color, or click **More Colors** to customize the color within a wider range in the Pick a Color dialog. You can also input the hexadecimal value of a color directly in the text box after #.

Transparency
Specifies the transparency for color of the border.

Line Style
Specifies the line style to apply to the border of the donuts.

Thickness
Specifies the thickness of the border, in pixels.

End Caps
Specifies the ending style of the border line.

- **butt**
  Ends unclosed sub-paths and dash segments with no added decoration.
- **round**
  Ends unclosed sub-paths and dash segments with a round decoration that has a radius equal to half of the width of the pen.
- **square**
  Ends unclosed sub-paths and dash segments with a square projection that extends beyond the end of the segment to a distance equal to half of the line width.

Line Joint
Specifies the line joint style for the border line.

- **miter**
  Joins path segments by extending their outside edges until they meet.
- **round**
  Joins path segments by rounding off the corner at a radius of half the line width.
- **bevel**
  Joins path segments by connecting the outer corners of their wide outlines with a straight segment.

Sample
Displays a preview sample of your selection.

Path
Specifies the fill pattern of the border line.

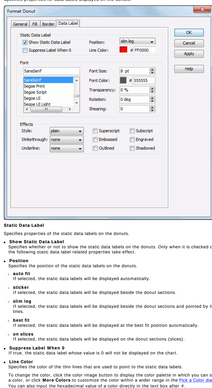
- **Outline Path**
  Specifies the fill pattern of the border line to be outline path.
- **Fill Path**
  Specifies the fill pattern of the border line to be whole path.

Dash
Specifies the dash size of border line.

- **Auto Adjusted Dash**
  If selected, the dash size will be adjusted automatically.
- **Fixed Dash Size**
  If selected, the dash size will be fixed size.

Data Label tab

Specifies properties for data labels displayed on the donuts.



Static Data Label
Specifies properties of the static data labels on the donuts.

- **Show Static Data Label**
  Specifies whether or not to show the static data labels on the donuts. Only when it is checked can the following static data label-related properties take effect.
- **Position**
  Specifies the position of the static data labels on the donuts.
  - **auto fit**
    If selected, the static data labels will be displayed automatically.
  - **sticker**
    If selected, the static data labels will be displayed beside the donut sections.
  - **slim tag**
    If selected, the static data labels will be displayed beside the donut sections and pointed by thin lines.
  - **best fit**
    If selected, the static data labels will be displayed at the best fit position automatically.
  - **on slices**
    If selected, the static data labels will be displayed on the donut sections (slices).
- **Suppress Label When 0**
  If true, the static data label whose value is 0 will not be displayed on the donut.
- **Line Color**
  Specifies the color of the thin lines that are used to point to the static data labels.

  To change the color, click the color image button to display the color palette in which you can select a color, or click **More Colors** to customize the color within a wider range in the Pick a Color dialog. You can also input the hexadecimal value of a color directly in the text box after #.

Font
Specifies the font format of text in the data labels.

- **Font Set**
  Lists all the available font faces that can be selected to apply to the text.
- **Font Size**
  Specifies the font size of the text.
- **Font Color**
  Specifies the font color of the text.

  To change the color, click the color image button to display the color palette in which you can select a color, or click **More Colors** to customize the color within a wider range in the Pick a Color dialog. Or click **More Fill Effects** to customize in the Fill Effects dialog. You can also input the hexadecimal value of a color directly in the text box after #.
- **Transparency**
  Specifies the transparency of the text.
- **Rotation**
  Specifies the rotation angle of the text around its center, in degrees. The default value is 0.
- **Shearing**
  Specifies the gradient of the text.

Effects
Specifies the special effects of text in the data labels.

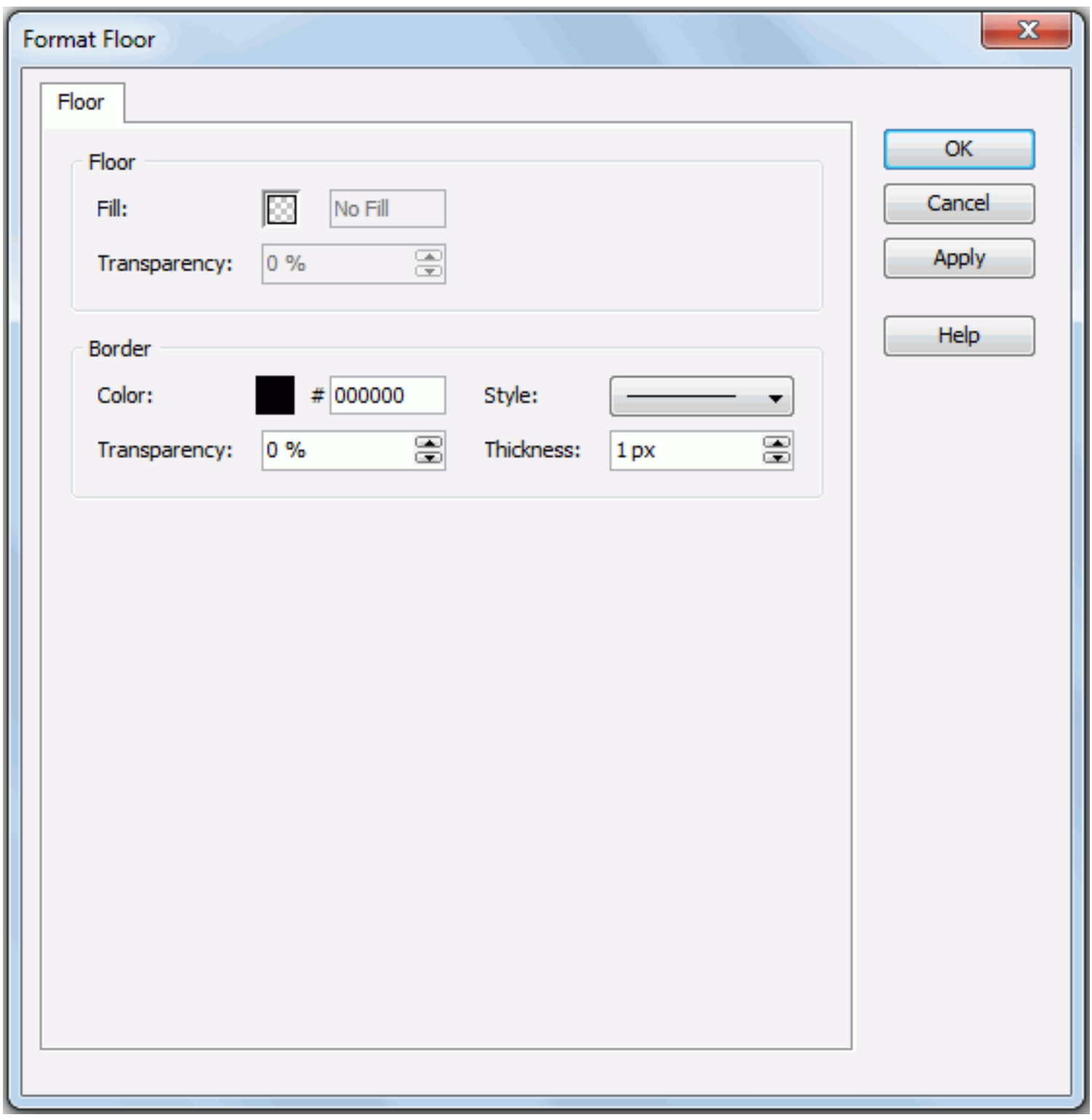
- **Style**
  Specifies the font style of the text. It can be one of the following: plain, bold, italic, and bold italic.
- **Strikethrough**
  Specifies the style of the horizontal line with which the text is struck through. It can be one of the following: none, single line, bold line, and double lines.
- **Underline**
  Specifies the style of the horizontal line under the text. It can be one of the following: none, single, single lower, bold line, bold lower, double lines, bold double, patterned line, and bold patterned.
- **Superscript**
  Raises the text above the baseline and changes the text to a smaller font size, if a smaller size is available.
- **Embossed**
  Makes the text appear to be raised off the page in relief.
- **Outlined**
  Displays the inner and outer borders of each character.
- **Subscript**
  Lowers the text below the baseline and changes the text to a smaller font size, if a smaller size is available.
- **Engraved**
  Makes the text appear to be imprinted or pressed into the page.
- **Shadowed**
  Adds a shadow beneath and to the right of the text.

Related topics:

- Formatting the pies/donuts

Format Floor dialog

This dialog appears when you double-click the floor of a 3-D chart, or right-click a 3-D chart and then select Format Walls > Format Floor from the shortcut menu.



The following are details about options in the dialog:

Floor

Specifies the color or fill effect and the transparency for the floor of the 3-D chart.

- **Fill**
  Specifies the color schema for the floor of a 3-D chart.

  To change the color, click the color image button to display the color palette in which you can select a color, or click **More Colors** to customize the color within a wider range in the Pick a Color dialog, or click **More Fill Effects** to customize in the Fill Effects dialog. You can also input the hexadecimal value of a color directly in the text box after #.

- **Transparency**
  Specifies the transparency of the color schema that is applied to the floor of a 3-D chart.

Border

Specifies properties for border of the chart floor.

- **Color**
  Specifies the color for border of the chart floor.

  To change the color, click the color image button to display the color palette in which you can select a color, or click **More Colors** to customize the color within a wider range in the Pick a Color dialog. You can also input the hexadecimal value of a color directly in the text box after #.

- **Transparency**
  Specifies the transparency for color of the border.

- **Style**
  Specifies the style for border of the chart floor.

- **Thickness**
  Specifies the thickness for border of the chart floor, in pixels.

OK

Applies the changes and closes the dialog.

Cancel

Does not retain any changes and closes the dialog.

Apply

Applies all changes and leaves the dialog open.

Help

Displays the help document about this feature.

Related topics:

- Formatting the wall/floor

Format Indicator dialog for library component

This dialog appears when you double-click an indicator of a chart in a library component, or right-click it and then select Format Indicator from the shortcut menu. It enables you to format the indicators in an indicator chart, and consists of the following tabs:

- General tab
- Border tab
- Data Label tab
- Behaviors tab

OK
Applies the changes and closes the dialog.

Cancel
Does not retain any changes and closes the dialog.

Apply
Applies all changes and leaves the dialog open.

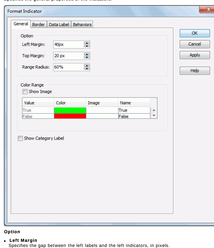
Help
Displays the help document about this feature.

General tab

The options in the tab differ with value types of the indicator chart: Boolean type, or String/Numeric type.

General tab for Boolean type values

Specifies the general properties of the indicators.



Option

- **Left Margin**
  Specifies the gap between the left labels and the left indicators, in pixels.
- **Top Margin**
  Specifies the gap between the top labels and the top indicators, in pixels.
- **Range Radius**
  Specifies the relative size of an indicator in a percentage of the total indicator size.

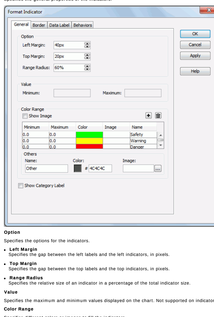
Color Range
Specifies different colors or images to fill the indicators.

- **Show Image**
  Specifies whether to use images instead of colors to represent the values. If checked, you can select an image to represent a specified value, and the color setting in the Color column will be ignored.
- **Value**
  Lists the value for the indicator chart.
- **Color**
  Specifies the color scheme for the value. Click in the color cell to display the color palette in which you can select a color, or click **More Colors** to customize the color within a wider range in the Pick a Color dialog, or click **More Fill Effects** to customize in the Fill Effects dialog.
- **Image**
  Specifies the image to represent the value.
- **Name**
  Specifies the name of the value.

Show Category Label
Specifies whether to show the category labels for the indicators. The label displays the category value each indicator represents.

General tab for String/Numeric type values

Specifies the general properties of the indicators.



Option
Specifies the options for the indicators.

- **Left Margin**
  Specifies the gap between the left labels and the left indicators, in pixels.
- **Top Margin**
  Specifies the gap between the top labels and the top indicators, in pixels.
- **Range Radius**
  Specifies the relative size of an indicator in a percentage of the total indicator size.

Value
Specifies the maximum and minimum values displayed on the chart. Not supported on indicator chart.

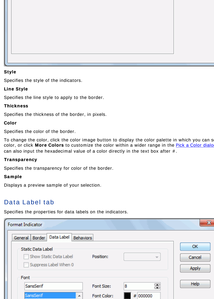
Color Range
Specifies different colors or images to fill the indicators.

- **Show Image**
  Specifies whether to use images instead of colors to represent the value ranges. If checked, the color setting in the Color column will be ignored.
- Adds a new value range.
- Removes the selected value range.
- **Minimum**
  Specifies the minimum value of the range.
- **Maximum**
  Specifies the maximum value of the range.
- **Color**
  Specifies the color scheme of the value range. Click in the color cell to display the color palette in which you can select a color, or click **More Colors** to customize the color within a wider range in the Pick a Color dialog, or click **More Fill Effects** to customize in the Fill Effects dialog.
- **Image**
  Specifies the image to represent the value range.
- **Name**
  Specifies the name of the value range.
- **Others**
  Specifies the properties for values that do not fall into any of the ranges you have defined.
  - **Name**
    Specifies the name for the values.
  - **Color**
    Specifies the color for the values.
    To change the color, click the color image button to display the color palette in which you can select a color, or click **More Colors** to customize the color within a wider range in the Pick a Color dialog. You can also input the hexadecimal value of a color directly in the text box after #.
  - **Image**
    Specifies the image to represent the values.

Show Category Label
Specifies whether to show the category labels for the indicators. The label displays the category value each indicator represents.

Border tab

Specifies the properties for border of the indicators, which take effect only when the Border property on chart paper is set to true in the Report Inspector and the indicators are not represented by images.



Style
Specifies the style of the indicators.

Line Style
Specifies the line style to apply to the border.

Thickness
Specifies the thickness of the border, in pixels.

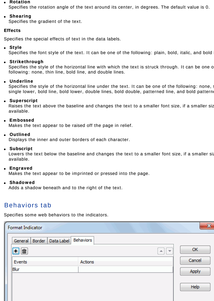
Color
Specifies the color of the border.
To change the color, click the color image button to display the color palette in which you can select a color, or click **More Colors** to customize the color within a wider range in the Pick a Color dialog. You can also input the hexadecimal value of a color directly in the text box after #.

Transparency
Specifies the transparency for color of the border.

Sample
Displays a preview sample of your selection.

Data Label tab

Specifies the properties for data labels on the indicators.



Static Data Label
Specifies properties of the static data labels. Not supported on indicator chart.

Font
Specifies the font format of text in the data labels.

- **Font list**
  Lists all the available font faces that can be selected to apply to the text.
- **Font Size**
  Specifies the font size of the text.
- **Font Color**
  Specifies the font color of the text.
  To change the color, click the color image button to display the color palette in which you can select a color, or click **More Colors** to customize the color within a wider range in the Pick a Color dialog, or click **More Fill Effects** to customize in the Fill Effects dialog. You can also input the hexadecimal value of a color directly in the text box after #.
- **Transparency**
  Specifies the transparency of the text.
- **Rotation**
  Specifies the rotation angle of the text around its center, in degrees. The default value is 0.
- **Shearing**
  Specifies the gradient of the text.

Effects
Specifies the special effects of text in the data labels.

- **Style**
  Specifies the font style of the text. It can be one of the following: plain, bold, italic, and bold italic.
- **Strikethrough**
  Specifies the style of the horizontal line with which the text is struck through. It can be one of the following: none, thin line, bold line, and double line.
- **Underline**
  Specifies the style of the horizontal line under the text. It can be one of the following: none, single, single lower, bold line, bold lower, double lines, bold double, patterned line, and bold patterned.
- **Superscript**
  Lifts the text above the baseline and changes the text to a smaller font size, if a smaller size is available.
- **Embossed**
  Makes the text appear to be raised off the page in relief.
- **Outlined**
  Displays the inner and outer borders of each character.
- **Subscript**
  Lowers the text below the baseline and changes the text to a smaller font size, if a smaller size is available.
- **Engraved**
  Makes the text appear to be imprinted or pressed into the page.
- **Shadowed**
  Adds a shadow beneath and to the right of the text.

Behaviors tab

Specifies some web behaviors to the indicators.

- Adds a new web behavior line.
- Removes the selected web behavior.
- Moves the selected web behavior up a step.
- Moves the selected web behavior down a step.

Events
Specifies the trigger event.

Actions
Specifies the action you want the event to trigger.

- Opens the Web Action List dialog to bind a web action to the event.

Related topics:

- Formatting the indicators

Format Indicator dialog for report

This dialog appears when you double-click an indicator of a chart in a page report or web report, or right-click it and then select Format Indicator from the shortcut menu. It enables you to format the indicators in an indicator chart, and consists of the following tabs:

- General tab
- Border tab
- Data Label tab

OK

Applies the changes and closes the dialog.

Cancel

Does not retain any changes and closes the dialog.

Apply

Applies all changes and leaves the dialog open.

Help

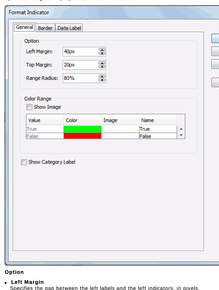
Displays the help document about this feature.

General tab

The options in the tab differ with value types of the indicator chart: Boolean type, or String/Numeric type.

General tab for Boolean type values

Specifies the general properties of the indicators.



Option

- **Left Margin**
  Specifies the gap between the left labels and the left indicators, in pixels.
- **Top Margin**
  Specifies the gap between the top labels and the top indicators, in pixels.
- **Range Radius**
  Specifies the relative size of an indicator in a percentage of the total indicator size.

Color Range

Specifies different colors or images to fill the indicators.

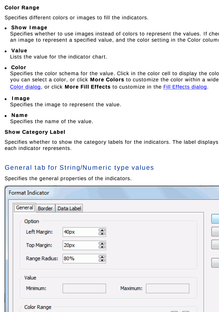
- **Show Image**
  Specifies whether to use images instead of colors to represent the values. If checked, you can select an image to represent a specified value, and the color setting in the Color column will be ignored.
- **Value**
  Lists the value for the indicator chart.
- **Color**
  Specifies the color schema for the value. Click in the color cell to display the color palette in which you can select a color, or click **More Colors** to customize the color within a wider range in the Pick a Color dialog, or click **More Fill Effects** to customize in the Fill Effects dialog.
- **Image**
  Specifies the image to represent the value.
- **Name**
  Specifies the name of the value.

Show Category Label

Specifies whether to show the category labels for the indicators. The label displays the category value each indicator represents.

General tab for String/Numeric type values

Specifies the general properties of the indicators.



Option

Specifies the options for the indicators.

- **Left Margin**
  Specifies the gap between the left labels and the left indicators, in pixels.
- **Top Margin**
  Specifies the gap between the top labels and the top indicators, in pixels.
- **Range Radius**
  Specifies the relative size of an indicator in a percentage of the total indicator size.

Value

Specifies the maximum and minimum values displayed on the chart. Not supported on indicator chart.

Color Range

Specifies different colors or images to fill the indicators.

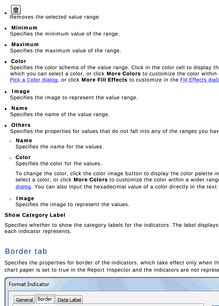
- **Show Image**
  Specifies whether to use images instead of colors to represent the value ranges. If checked, the color setting in the Color column will be ignored.
- Adds a new value range.
- Removes the selected value range.
- **Minimum**
  Specifies the minimum value of the range.
- **Maximum**
  Specifies the maximum value of the range.
- **Color**
  Specifies the color schema of the value range. Click in the color cell to display the color palette in which you can select a color, or click **More Colors** to customize the color within a wider range in the Pick a Color dialog, or click **More Fill Effects** to customize in the Fill Effects dialog.
- **Image**
  Specifies the image to represent the value range.
- **Name**
  Specifies the name of the value range.
- **Others**
  Specifies the properties for values that do not fall into any of the ranges you have defined.
  - **Name**
    Specifies the name for the values.
  - **Color**
    Specifies the color for the values.
    To change the color, click the color image button to display the color palette in which you can select a color, or click **More Colors** to customize the color within a wider range in the Pick a Color dialog. You can also input the hexadecimal value of a color directly in the text box after #.
  - **Image**
    Specifies the image to represent the values.

Show Category Label

Specifies whether to show the category labels for the indicators. The label displays the category value each indicator represents.

Border tab

Specifies the properties for border of the indicators, which take effect only when the Border property on chart paper is set to true in the Report Inspector and the indicators are not represented by images.



Style

Specifies the style of the indicators.

Line Style

Specifies the line style to apply to the border.

Thickness

Specifies the thickness of the border, in pixels.

Color

Specifies the color of the border.

To change the color, click the color image button to display the color palette in which you can select a color, or click **More Colors** to customize the color within a wider range in the Pick a Color dialog. You can also input the hexadecimal value of a color directly in the text box after #.

Transparency

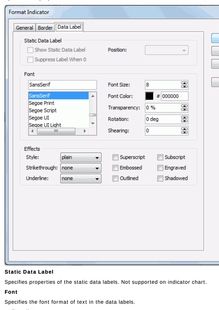
Specifies the transparency for color of the border.

Sample

Displays a preview sample of your selection.

Data Label tab

Specifies the properties for data labels on the indicators.



Static Data Label

Specifies properties of the static data labels. Not supported on indicator chart.

Font

Specifies the font format of text in the data labels.

- **Font list**
  Lists all the available font faces that can be selected to apply to the text.
- **Font Size**
  Specifies the font size of the text.
- **Font Color**
  Specifies the font color of the text.
  To change the color, click the color image button to display the color palette in which you can select a color, or click **More Colors** to customize the color within a wider range in the Pick a Color dialog, or click **More Fill Effects** to customize in the Fill Effects dialog. You can also input the hexadecimal value of a color directly in the text box after #.
- **Transparency**
  Specifies the transparency of the text.
- **Rotation**
  Specifies the rotation angle of the text around its center, in degrees. The default value is 0.
- **Shearing**
  Specifies the gradient of the text.

Effects

Specifies the special effects of text in the data labels.

- **Style**
  Specifies the font style of the text. It can be one of the following: plain, bold, italic, and bold italic.
- **Strikethrough**
  Specifies the style of the horizontal line with which the text is struck through. It can be one of the following: none, thin line, bold line, and double lines.
- **Underline**
  Specifies the style of the horizontal line under the text. It can be one of the following: none, single, single lower, bold line, bold lower, double lines, bold double, patterned line, and bold patterned.
- **Superscript**
  Raises the text above the baseline and changes the text to a smaller font size, if a smaller size is available.
- **Embossed**
  Makes the text appear to be raised off the page in relief.
- **Outlined**
  Displays the inner and outer borders of each character.
- **Subscript**
  Lowers the text below the baseline and changes the text to a smaller font size, if a smaller size is available.
- **Engraved**
  Makes the text appear to be imprinted or pressed into the page.
- **Shadowed**
  Adds a shadow beneath and to the right of the text.

Related topics:

- Formatting the indicators

Format Label dialog

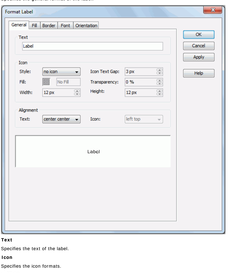
This dialog appears when you right-click a label of a chart and then select Format Label from the shortcut menu. It helps you to format the titles of a chart, and consists of the following tabs:

- General tab
- Fill tab
- Border tab
- Font tab
- Orientation tab

OK
Applies the changes and closes the dialog.
Cancel
Does not retain any changes and closes the dialog.
Apply
Applies all changes and leaves the dialog open.
Help
Displays the help document about this feature.

General tab

Specifies the general format of the label.



Text
Specifies the text of the label.
Icon
Specifies the icon formats.

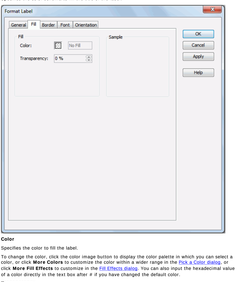
- **Icon**
  Specifies the style for the icon that will be used in the label. If you don't want to show any icon in the label, you can choose No Icon from the drop-down list.
- **Icon Text Gap**
  Specifies the distance between the icon and the label text.
- **Fill**
  Specifies the color to fill the selected icon.
  To change the color, click the color image button to display the color palette in which you can select a color, or click **More Colors** to customize the color within a wider range in the Pick a Color dialog, or click **More Fill Effects** to customize in the Fill Effects dialog. You can also input the hexadecimal value of a color directly in the text box after # if you have changed the default color.
- **Transparency**
  Specifies the transparency of the color that is used to fill the selected icon.
- **Width**
  Specifies the width of the icon, in pixels.
- **Height**
  Specifies the height of the icon, in pixels.

Alignment
Specifies the alignment format for text and icons in the label.

- **Text**
  Specifies the alignment format for text in the label.
- **Icon**
  Specifies the alignment format for icons in the label.

Fill tab

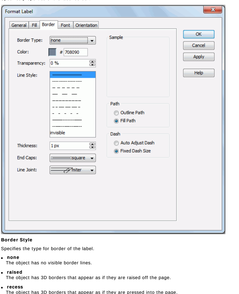
Specifies the color schema to fill the title of the label.



Color
Specifies the color to fill the label.
To change the color, click the color image button to display the color palette in which you can select a color, or click **More Colors** to customize the color within a wider range in the Pick a Color dialog, or click **More Fill Effects** to customize in the Fill Effects dialog. You can also input the hexadecimal value of a color directly in the text box after # if you have changed the default color.
Transparency
Specifies the transparency of the color.
Sample
Displays the specified color and transparency effects.

Border tab

Specifies properties of the label border.



Border Style
Specifies the type for border of the label.

- **none**
  The object has no visible border lines.
- **raised**
  The object has 3D borders that appear as if they are raised off the page.
- **recess**
  The object has 3D borders that appear as if they are pressed into the page.
- **shadow**
  The object has two shadowed borders, beneath and to the right of the object.
- **solid**
  The object has single-line borders.

Color
Specifies the color for border of the label.
To change the color, click the color image button to display the color palette in which you can select a color, or click **More Colors** to customize the color within a wider range in the Pick a Color dialog. You can also input the hexadecimal value of a color directly in the text box after #.
Line Style
Specifies the line style to apply to the label border.
Thickness
Specifies the thickness of the label border, in pixels.
Transparency
Specifies the transparency for the color of the label border.
End Caps
Specifies the ending style of the border line.

- **Butt**
  Ends unclosed sub paths and dash segments with no added decoration.
- **Round**
  Ends unclosed sub paths and dash segments with a round decoration that has a radius equal to half of the width of the pen.
- **Square**
  Ends unclosed sub paths and dash segments with a square projection that extends beyond the end of the segment to a distance equal to half of the line width.

Line Joint
Specifies the line joint style for the border line.

- **Miter**
  Joins path segments by extending their outside edges until they meet.
- **Round**
  Joins path segments by rounding off the corner at a radius of half the line width.
- **Bevel**
  Joins path segments by connecting the outer corners of their wide outlines with a straight segment.

Path
Specifies the fill pattern of the border line.

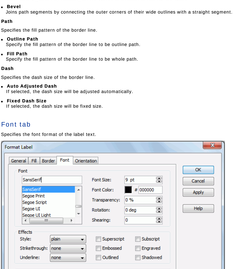
- **Outline Path**
  Specify the fill pattern of the border line to be outline path.
- **Fill Path**
  Specify the fill pattern of the border line to be whole path.

Dash
Specifies the dash size of the border line.

- **Auto Adjusted Dash**
  If selected, the dash size will be adjusted automatically.
- **Fixed Dash Size**
  If selected, the dash size will be fixed size.

Font tab

Specifies the font format of the label text.



Font
Specifies the font format of label text.

- **Font list**
  Lists all the available font faces that can be selected to apply to the text.
- **Font Size**
  Specifies the font size of the text.
- **Font Color**
  Specifies the font color of the text.
  To change the color, click the color image button to display the color palette in which you can select a color, or click **More Colors** to customize the color within a wider range in the Pick a Color dialog, or click **More Fill Effects** to customize in the Fill Effects dialog. You can also input the hexadecimal value of a color directly in the text box after #.
- **Transparency**
  Specifies the transparency of the text.
- **Rotation**
  Specifies the rotation angle of the text around its center, in degrees. The default value is 0.
- **Shearing**
  Specifies the gradient of the text.

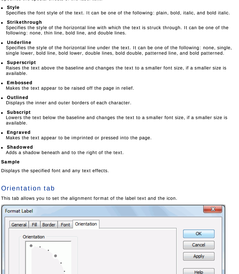
Effects
Specifies the special effects of the label text.

- **Style**
  Specifies the font style of the text. It can be one of the following: plain, bold, italic, and bold italic.
- **Strikethrough**
  Specifies the type of the horizontal line with which the text is struck through. It can be one of the following: none, thin line, bold line, and double lines.
- **Underline**
  Specifies the style of the horizontal line under the text. It can be one of the following: none, single, single lower, bold line, bold lower, double lines, bold double, patterned line, and bold patterned.
- **Superscript**
  Raises the text above the baseline and changes the text to a smaller font size. If a smaller size is available.
- **Embossed**
  Makes the text appear to be raised off the page in relief.
- **Outlined**
  Displays the inner and outer borders of each character.
- **Subscript**
  Lowers the text below the baseline and changes the text to a smaller font size. If a smaller size is available.
- **Engraved**
  Makes the text appear to be imprinted or pressed into the page.
- **Shadowed**
  Adds a shadow beneath and to the right of the text.

Sample
Displays the specified font and any text effects.

Orientation tab

This tab allows you to set the alignment format of the label text and the icon.



Orientation
Drag the pointer in the box to specify the alignment format of the label text and the icon.
Angle
Specify the alignment format of the label text and the icon.

Related topics:

- Formatting the labels

Format Legend dialog for library component

This dialog appears when you double-click the legend of a chart in a library component, or right-click it and choose **Properties**. Use this dialog to specify the legend's appearance. This dialog allows you to format the legend in the following tabs:

- Fill tab
- Border tab
- Placement tab
- Font tab
- Orientation tab
- Format tab
- Mark tab
- Behaviors tab

Apply

Applies the changes and closes the dialog.

Cancel

Closes the dialog without any changes and closes the dialog.

Apply

Applies all changes and leaves the dialog open.

Help

Displays the help document about this feature.

Fill tab

Specifies the color and its transparency for the legend.

Color:

Specifies the color to fill the legend.

To change the color, click the color swatch or enter the color name in the color palette in which you can select a color. Enter **RGB (Red,Green,Blue)** or **HSL (Hue,Saturation,Luminance)** values in the color palette to define a custom color. Enter **Auto** to use the default color. Enter **None** or **0** if you have changed the default color.

Transparency

Specifies the transparency of the color.

Border tab

Specifies the properties for border of the legend.

Border Style

Specifies the type for border of the legend.

- **None**
  The object has no visible border.
- **Inset**
  The object has 3D borders that appear as if they are raised off the page.
- **Outset**
  The object has 3D borders that appear as if they are pressed into the page.
- **Flat**
  The object has two shadowed borders, lightest and to the right of the object.
- **Shadow**
  The object has single line borders.

Specifies the color for border of the legend.

Specifies the line style for the chart's outline. Enter or choose the color palette in which you can select a color. Enter **RGB (Red,Green,Blue)** or **HSL (Hue,Saturation,Luminance)** values in the color palette to define a custom color. You can also input the hexadecimal value of a color directly in the text box after #.

Transparency

Specifies the line style for the chart's outline.

Thickness

Specifies the thickness of the border, in pixels.

Transparency

Specifies the transparency for the color of the border.

End Caps

Specifies the ending style of the line.

- **Round**
  Draws rounded ends with pointed and square extensions with no dotted decoration.
- **Round**
  Draws rounded ends with pointed and square extensions with a round decoration that has a radius equal to half the width of the pen.
- **Square**
  Draws square ends with pointed and square extensions with a square proportion that extends beyond the end of the segment a distance equal to half of the line width.

Line Joint

Specifies the line joint style for the border line.

- **Miter**
  Joins each segments by extending their outside edges until they meet.
- **Round**
  Joins each segments by connecting the outer corners of each line with a circular arc.
- **Bevel**
  Joins each segments by connecting the outer corners of their wide outlines with a straight segment.

Mark

Specifies the fill pattern of the border line.

- **Auto Mark**
  Specifies the fill pattern of the border line to be solid.
- **Fill Mark**
  Specifies the fill pattern of the border line to be stripe pattern.

Gap

Specifies the dash size of the border line.

- **Auto Adjusted Gap**
  Specifies the dash size will be adjusted automatically.
- **Pixel/Chart Size**
  Specifies the dash size will be fixed size.

Sample

Displays a preview sample of your selection.

Placement tab

Specifies the location of the legend object.

Alignment

Specifies the alignment for the legend.

Spacing

Specifies the distance between the legend and its labels.

- **Vertical Margin**
  Specifies the vertical distance between the legend and its labels.
- **Horizontal Margin**
  Specifies the vertical distance between legend labels, in pixels.
- **Vertical Inner Spacing**
  Specifies the vertical distance between legend labels, in pixels.
- **Horizontal Inner Spacing**
  Specifies the horizontal distance between legend entry labels, in pixels.

Reverse Legend

Specifies whether the legend entries will be re-arranged in a reverse order.

Show Scrollbar

Specifies whether to show a scrollbar on the legend to fully view the legend content when the content exceeds the legend's size.

Translate

Specifies whether to translate the legend entry label text when the text overflow the frame.

Font tab

Specifies the font format of text in the legend entry labels.

Font

Specifies the font format of text in the legend entry labels.

- **Font Size**
  Lists all the available font faces that can be selected to apply to the text.
- **Transparency**
  Specifies the text color of the text.
- **Font Color**
  Specifies the text color of the text.

Specifies the text rotation. Enter the rotation angle further in the color palette in which you can select a color. Or enter **More Colors** to open the color editor in order range in the Field Color dialog. Enter **Auto** to use the default color. Enter **None** or **0** in Field Color. You can also input the hexadecimal value of a color directly in the text box after #.

Drawing

Specifies the special effects of text in the legend.

- **Rotation**
  Specifies the text rotation of the text around its center, in degrees. The default value is 0.
- **Shearing**
  Specifies the gradient of the text.

Effects

Specifies the special effects of text in the legend.

- **Underline**
  Specifies the text style of the text. It can be one of the following: plain, bold, italic, and bold italic.
- **Strikeout**
  Specifies the text of the horizontal line with which the text is struck through. It can be one of the following: plain, bold, italic, and bold italic.
- **Shadow**
  Specifies the text of the horizontal line under the text. It can be one of the following: none, single, double, thick, dotted, dashed, dash dot, dash dot dot, wavy, and wavy heavy.
- **Superscript**
  Specifies the text of the horizontal line above the text. It can be one of the following: none, single, double, thick, dotted, dashed, dash dot, dash dot dot, wavy, and wavy heavy.
- **Subscript**
  Specifies the text of a character to be raised off the page in relief.
- **Embossed**
  Specifies the text of a character to be lowered off the page.
- **Outline**
  Specifies the text of a character to be lowered off the page.
- **Shadow**
  Specifies the text of a character to be raised off the page.
- **Engraved**
  Specifies the text of a character to be lowered off the page.
- **Shadowed**
  Specifies the text of a character to be raised off the page.

Sample

Displays the specified font and any text effects.

Orientation tab

Specifies the alignment format of the legend entry labels.

Orientation

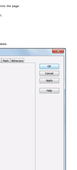
Specifies the alignment of the legend entry labels.

Angle

Specifies the angle of the legend entry labels.

Format tab

Specifies properties for the legend entry labels.



Category

Specifies different formats to display the legend entry labels.

- **Stack**
  Lists all the formats of a value of a stack type that can be applied to the legend entry labels.
- **Date**
  Lists all the formats of a value of a number type that can be applied to the legend entry labels.
- **Time**
  Lists all the formats of a value of a date/time type that can be applied to the legend entry labels.
- **Text**
  Lists all the formats of a value of a text type that can be applied to the legend entry labels.

Properties

Specifies the properties for the legend entry labels.

Stack

Lists all the formats that you selected from different categories.

Add

Adds the effects of the selected format that has been added into the Stack list box.

Delete

Removes a format from the Stack list box.

Sample

Displays the specified format and any text effects.

Note: To make settings in this tab take effect, you need to first make sure that the property Legend Entry Source of the chart's legend has been set to Legend Label Format in the Report Inspector.

Mark tab

Specifies the format of the legend entry marks.

Customize

Specifies the method to customize the format of the legend entry marks.

Spin Field Type

Specifies whether to apply the format of text and line styles to the legend entry marks. If checked, the legend entry marks are automatically inherit the style and color of line and text. Otherwise, the legend entry marks are automatically inherit the style and color of lines and text.

Mark Items

Specifies the style of the legend entry marks. Select the mark item from the box one by one and specify the style of each mark item in the dialogs below. Available entry mark items are Customize a checkbox, Text, Line, Text and Line.

- Customize a mark item to the box.

- Text
  Specifies a selected mark item as a text.

- Line
  Specifies a selected mark item as a line.

- Text and Line
  Specifies a selected mark item as a text and a line.

Width:

Specifies the width of the icons, in pixels.

Height:

Specifies the height of the icons, in pixels.

Gap:

Specifies the horizontal alignment format of the icons. It could be left, center and bottom.

Text

Specifies the text gap between each entry mark and entry label, in pixels.

Border

Specifies the border properties of the icons. Available only when Customize is checked.

Thickness:

Specifies the line thickness of the icons.

To change the color, click the color swatch or enter the color name in the color palette in which you can select a color. Enter **RGB (Red,Green,Blue)** or **HSL (Hue,Saturation,Luminance)** values in the color palette to define a custom color. You can also input the hexadecimal value of a color directly in the text box after # or click the color swatch to select the color.

Transparency:

Specifies the transparency of the icon border. Input the transparency directly in the text box or click the up and down arrows to control the value.

Sample

Displays the specified icon and mark effects.

Behaviors tab

Specifies some new behaviors to the legend of the chart.

- Add a new web behavior link

- Remove the selected web behavior.

- Edit the selected web behavior as a step.

Actions

Specifies the trigger event.

Actions

Specifies the action to perform when the trigger event occurs.

Click on the Web Action List dialog to bind a web action to the event.

Related topics
- Formatting the legend

Format Legend dialog for page report

This dialog appears when you double-click the legend of a chart in a page report, or right-click it and choose **Properties** from the shortcut menu. This dialog allows you to format the legend in a chart, and contains all the following tabs:

- Fill tab
- Border tab
- Placement tab
- Font tab
- Orientation tab
- Format tab
- Mark tab

OK

Applies the changes and closes the dialog.

Cancel

Does not retain any changes and closes the dialog.

Apply

Applies all changes and leaves the dialog open.

Help

Displays the Help dialog.

Fill tab

Specifies the color and its transparency for the legend of the chart.

Color

Specifies the color to fill the legend.

To change the color, click the color image button to display the color palette in which you can select a color, or click **More Colors** to customize the color within a wider range in the Pick a Color dialog. Alternatively, you can **Pick Color** to customize the color within a wider range in the Pick a Color dialog. You can also input the hexadecimal value of a color directly in the text box after it. Once you have changed the default color.

Transparency

Specifies the transparency of the color.

Border tab

Specifies the properties for border of the legend.

Border Style

Specifies the type for border of the legend.

- **none**
  The object has no visible border lines.
- **rounded**
  The object has 3D borders that appear as if they are raised off the page.
- **raised**
  The object has 3D borders that appear as if they are pressed into the page.
- **shadow**
  The object has two shadowed borders, beneath and to the right of the object.
- **solid**
  The object has single-line borders.

Color

Specifies the line color to apply to the border of the legend.

Thickness

Specifies the thickness of the border, in pixels.

Transparency

Specifies the transparency for the color of the border.

End Caps

Specifies the ending style of the border line.

- **flat**
  Draws unclosed sub paths and dash segments with no added decoration.
- **round**
  Draws unclosed sub paths and dash segments with a round decoration that has a radius equal to half of the width of the pen.
- **square**
  Draws unclosed sub paths and dash segments with a square projection that extends beyond the end of the segment that is the distance equal to half of the pen width.

Line Joint

Specifies the line joint style for the border line.

- **miter**
  Miter joint segments by extending their outside edges until they meet.
- **Round**
  Joins path segments by rounding off the corner at a radius of half the line width.
- **Bevel**
  Joins path segments by connecting the outer corners of their wide outlines with a straight segment.

Path

Specifies the fill pattern of the border line.

- **Outline Path**
  Specify the fill pattern of the border line to be outline path.
- **Fill Path**
  Specify the fill pattern of the border line to be white path.

Dash

Specifies the dash style of the border line.

- **Pattern Adjusted Dash**
  The dash lengths and dash size will be adjusted automatically.
- **Fixed Dash Size**
  Specifies the dash size will be fixed size.

Preview

Displays a preview sample of your selections.

Placement tab

Specifies the location of the legend object.

Position

Specifies the position of the legend to be left, right, top, bottom or docked it by Plot area automatically. Select one of the options.

Alignment

Specifies the alignment format for the legend.

Legend Label Gap

Specifies the horizontal distance between the legend entry labels.

- **Vertical Margin**
  Specifies the vertical distance between the legend entry labels, in pixels.
- **Vertical Label Spacing**
  Specifies the vertical distance between legend labels, in pixels.

Horizontal Margin

Specifies the horizontal distance between legend entry labels, in pixels.

- **Horizontal Label Spacing**
  Specifies the horizontal distance between the legend entry labels, in pixels.

Reverse Legend

Specifies whether the legend entries will be re-arranged in a reverse order.

Show Scrollbar

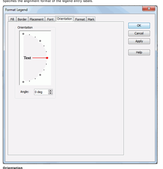
Specifies whether to show a scrollbar on the legend to fully view the legend content when the content does not fit into the legend.

Truncate

Specifies whether to truncate the legend entry label text when the text overflow the labels.

Font tab

Specifies the font format of text in the legend entry.



Font

Specifies the font format of text in the legend entry labels.

- **Font Name**
  Lists all the available font faces that can be selected to apply to the text.
- **Font Size**
  Specifies the font size of the text.
- **Font Color**
  Specifies the font color of the text.
  To change the color, click the color image button to display the color palette in which you can select a color, or click **More Colors** to customize the color within a wider range in the Pick a Color dialog. Alternatively, you can **Pick Color** to customize the color within a wider range in the Pick a Color dialog. You can also input the hexadecimal value of a color directly in the text box after it.
- **Transparency**
  Specifies the transparency of the text.
- **Rotation**
  Specifies the rotation angle of the text around its center, in degrees. The default value is 0.
- **Shading**
  Specifies the gradient of the text.

Effects

Specifies the special effects of text in the legend entry labels.

- **Strikethrough**
  Specifies the style of the horizontal line in the text. It can be one of the following: none, dash, solid, and dash-dot.
- **Underline**
  Specifies the style of the horizontal line which which the text is struck through. It can be one of the following: none, dash, dot, dash-dot, dash-dot-dot, double lines, and double lines.
- **Subscript**
  Specifies the style of the horizontal line under the text. It can be set of the following: none, single, double, dash, dot, dash-dot, dash-dot-dot, double lines, and double underlined lines and bold underlined.
- **Superscript**
  Raises the text above the baseline and changes the text to a smaller font size, if a smaller size is available.
- **Embossed**
  Makes the text appear to be raised above the page.
- **Outline**
  Displays the outline and scale borders of each character.
- **Subscript**
  Lowers the text below the baseline and changes the text to a smaller font size, if a smaller size is available.
- **Engraved**
  Makes the text appear to be impressed or pressed into the page.
- **Shadowed**
  Places a shadow beneath and to the right of the text.

Shadow

Displays the specified font and key text effects.

Orientation tab

Specifies the alignment format of the legend entry labels.

Orientation

Specifies the orientation of the legend entry labels.

Angle

Specifies the angle of the legend entry labels.

Format tab

Specifies properties for the legend entry labels.

Category

Specifies different formats to display the legend entry labels.

- **Static**
  Lists all the formats of a value of a scale type that can be applied to the legend entry labels.
- **Number**
  Lists all the formats of a value of a number type that can be applied to the legend entry labels.
- **Date**
  Lists all the formats of a value of a date/time type that can be applied to the legend entry labels.
- **Text**
  Lists all the formats of a value of a text type that can be applied to the legend entry labels.
- **Mapping**
  Lists all the formats of a value of a mapping type that can be applied to the legend entry labels.

Format

Displays all the formats of the selected category. For more details about every format, see a decimal place property reference.

Properties

Displays the properties for the format that you selected.

Stack

Lists all the formats that you selected from different categories.

Remove

Removes a format from the Stack that has been added into the Stack list box.

Add

Adds a format to the Stack list box.

Clear

Removes all formats from the Stack list box.

Apply

Applies the specified format in the Stack list box to the legend entry labels.

Note: To make settings in this tab take effect, you need to first make sure that the property Label Format Source of the chart legend has been set to Legend Entries in the Report Inspector.

Mark tab

Specifies the format of the legend entry marks.

Customize

Specifies whether to customize the format of the legend entry marks.

Use Node as Mark

Specifies whether to apply the format of line and line nodes to the legend entry marks. If selected, the legend entry marks will automatically inherit the style and color of line nodes. Available for line charts only.

Mark Items

Specifies the style of the legend entry marks. Select the mark item from the box one by one and customize the style for each item from the drop-down list. Available only when Customize is checked.

- **Add**
  Adds a mark item to the list.
- **Remove**
  Removes a specified mark item from the box.
- **Up**
  Moves a selected mark item up a step.
- **Down**
  Moves a selected mark item down a step.

Uses

Specifies the properties of the icons.

- **Width**
  Specifies the width of the icons, in pixels.
- **Height**
  Specifies the height of the icons, in pixels.
- **H-Alignment**
  Specifies the horizontal alignment format of the icons. It could be left, center and right.
- **V-Alignment**
  Specifies the vertical alignment format of the icons. It could be top, center and bottom.

Gap

Specifies the gap between each entry marks and entry label, in pixels.

Border

Specifies the border properties of the icons. Available only when Customize is checked.

- **Color**
  Specifies the color of the icon border.
  To change the color, click the color image button to display the color palette in which you can select a color, or click **More Colors** to customize the color within a wider range in the Pick a Color dialog.
- **Thickness**
  Specifies the thickness of the icon border, in pixels.
- **Transparency**
  Specifies the transparency of the icon border. Input the transparency directly in the text box or click the **Pick** button to control the color.

Fill

Displays the specified font and mark effects.

Formatting the legend

Format Legend dialog for web report

This dialog appears when you double-click the legend of a chart in a web report, or right-click it and select Format Legend from the shortcut menu. This dialog allows you to format the legend in a chart, and consists of the following tabs:

- Fill tab
- Border tab
- Placement tab
- Font tab
- Orientation tab
- Format tab
- Mark tab

OK

Applies the changes and closes the dialog.

Cancel

Does not retain any changes and closes the dialog.

Apply

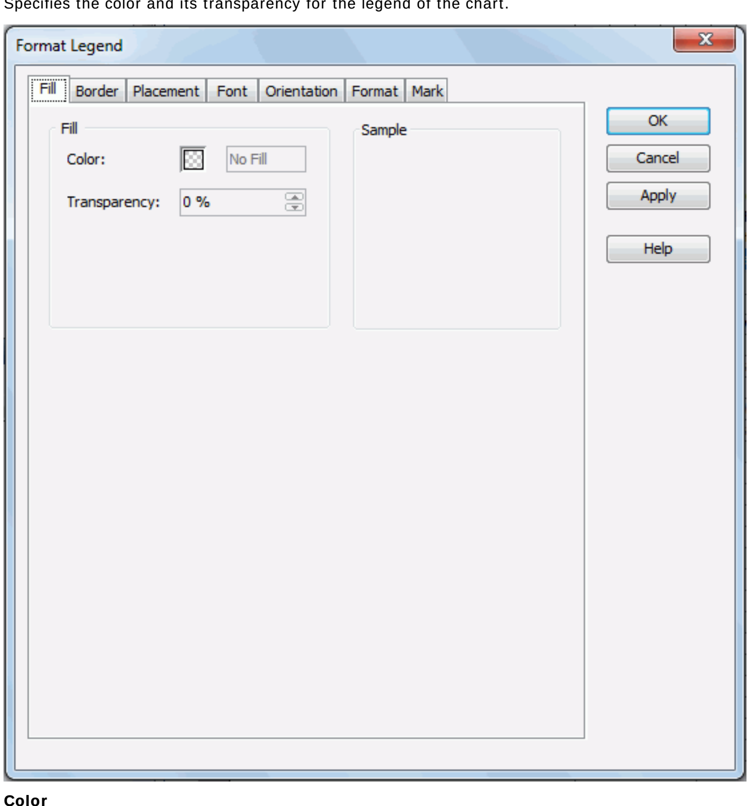
Applies all changes and leaves the dialog open.

Help

Displays the help document about this feature.

Fill tab

Specifies the color and its transparency for the legend of the chart.



Color

Specifies the color to fill the legend.

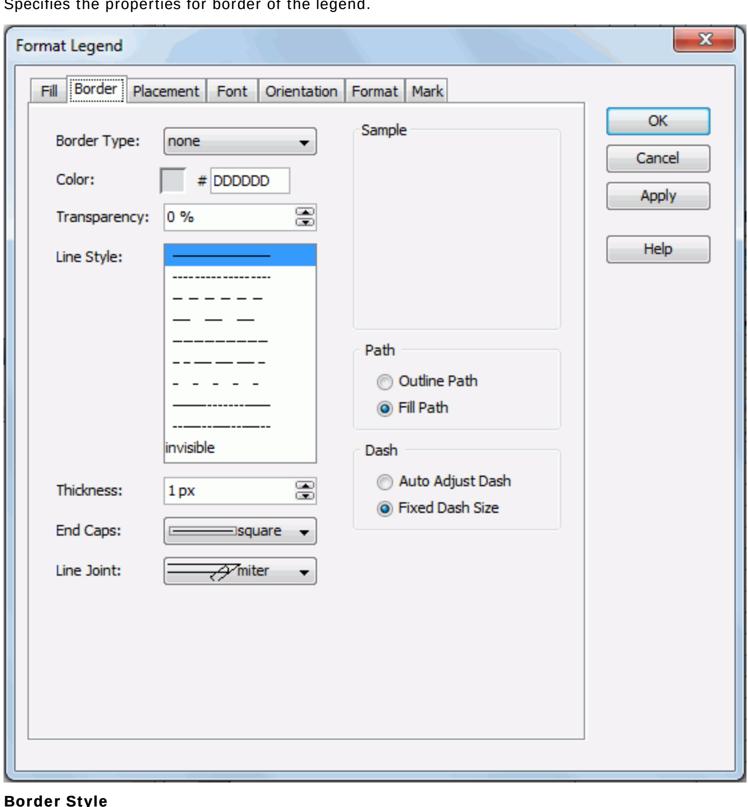
To change the color, click the color image button to display the color palette in which you can select a color, or click **More Colors** to customize the color within a wider range in the Pick a Color dialog, or click **More Fill Effects** to customize in the Fill Effects dialog. You can also input the hexadecimal value of a color directly in the text box after # if you have changed the default color.

Transparency

Specifies the transparency of the color.

Border tab

Specifies the properties for border of the legend.



Border Style

Specifies the type for border of the legend.

- **none**
  The object has no visible border lines.
- **raised**
  The object has 3D borders that appear as if they are raised off the page.
- **recess**
  The object has 3D borders that appear as if they are pressed into the page.
- **shadow**
  The object has two shadowed borders, beneath and to the right of the object.
- **solid**
  The object has single-line borders.

Color

Specifies the color for border of the legend.

To change the color, click the color image button to display the color palette in which you can select a color, or click **More Colors** to customize the color within a wider range in the Pick a Color dialog. You can also input the hexadecimal value of a color directly in the text box after #.

Line Style

Specifies the line style to apply to the border of the legend.

Thickness

Specifies the thickness of the border, in pixels

Transparency

Specifies the transparency for the color of the border.

End Caps

Specifies the ending style of the border line.

- **Butt**
  Ends unclosed sub paths and dash segments with no added decoration.
- **Round**
  Ends unclosed sub paths and dash segments with a round decoration that has a radius equal to half of the width of the pen.
- **Square**
  Ends unclosed sub paths and dash segments with a square projection that extends beyond the end of the segment to a distance equal to half of the line width.

Line Joint

Specifies the line joint style for the border line.

- **Miter**
  Joins path segments by extending their outside edges until they meet.
- **Round**
  Joins path segments by rounding off the corner at a radius of half the line width.
- **Bevel**
  Joins path segments by connecting the outer corners of their wide outlines with a straight segment.

Path

Specifies the fill pattern of the border line.

- **Outline Path**
  Specify the fill pattern of the border line to be outline path.
- **Fill Path**
  Specify the fill pattern of the border line to be whole path.

Dash

Specifies the dash size of the border line.

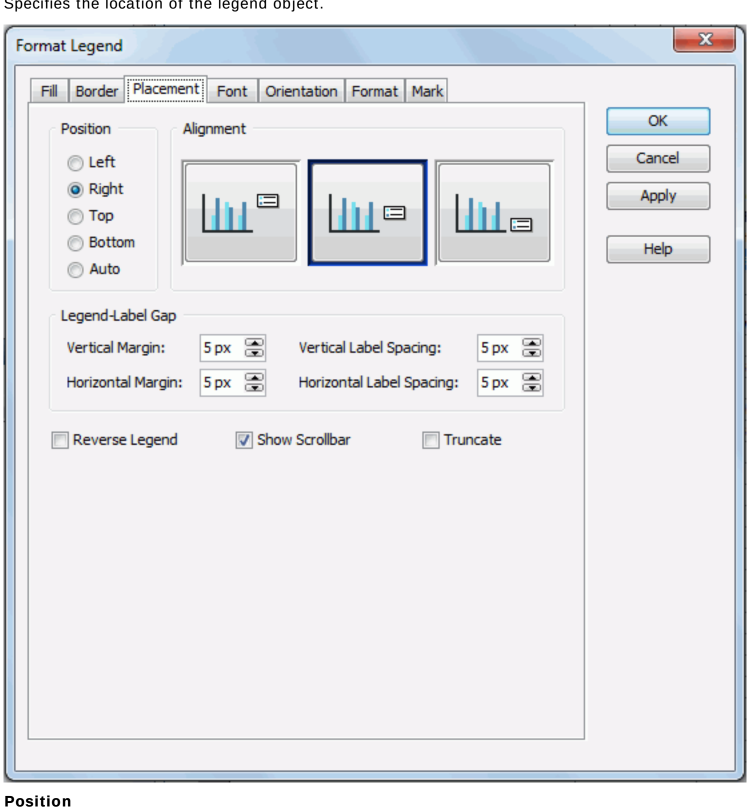
- **Auto Adjusted Dash**
  If selected, the dash size will be adjusted automatically.
- **Fixed Dash Size**
  If selected, the dash size will be fixed size.

Sample

Displays a preview sample of your selection.

Placement tab

Specifies the location of the legend object.



Position

Specifies the position of the legend to be left, right, top, bottom or decided it by JReport automatically. You can preview the samples.

Alignment

Specifies the alignment format for the legend.

Legend-Label Gap

Specifies the minimum distance between the legend entry labels.

- **Vertical Margin**
  Specifies the minimum vertical distance between the legend entry labels, in pixels.
- **Vertical Label Spacing**
  Specifies the vertical distance between legend labels, in pixels.
- **Horizontal Margin**
  Specifies the minimum horizontal distance between legend entry labels, in pixels.
- **Horizontal Label Spacing**
  Specifies the horizontal distance between the legend entry labels, in pixels.

Reverse Legend

Specifies whether the legend entries will be re-arranged in a reverse order.

Show Scrollbar

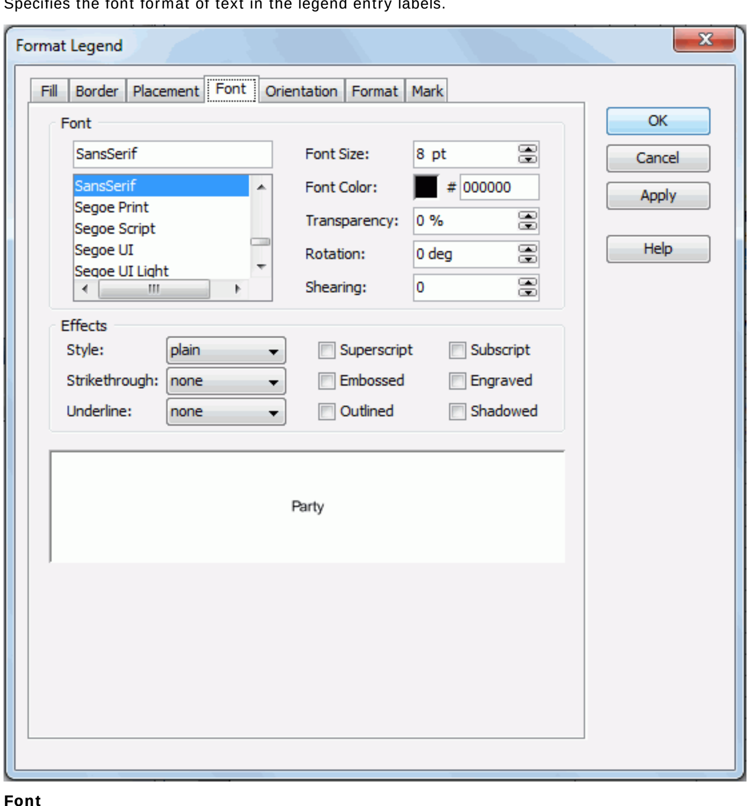
Specifies whether to show a scrollbar on the legend to fully view the legend content when the content does not fit into the legend.

Truncate

Specifies whether to truncate the legend entry label text when the text overflow the labels.

Font tab

Specifies the font format of text in the legend entry labels.



Font

Specifies the font format of text in the legend entry labels.

- **Font list**
  Lists all the available font faces that can be selected to apply to the text.
- **Font Size**
  Specifies the font size of the text.
- **Font Color**
  Specifies the font color of the text.

  To change the color, click the color image button to display the color palette in which you can select a color, or click **More Colors** to customize the color within a wider range in the Pick a Color dialog, or click **More Fill Effects** to customize in the Fill Effects dialog. You can also input the hexadecimal value of a color directly in the text box after #.
- **Transparency**
  Specifies the transparency of the text.
- **Rotation**
  Specifies the rotation angle of the text around its center, in degrees. The default value is 0.
- **Shearing**
  Specifies the gradient of the text.

Effects

Specifies the special effects of text in the legend entry labels.

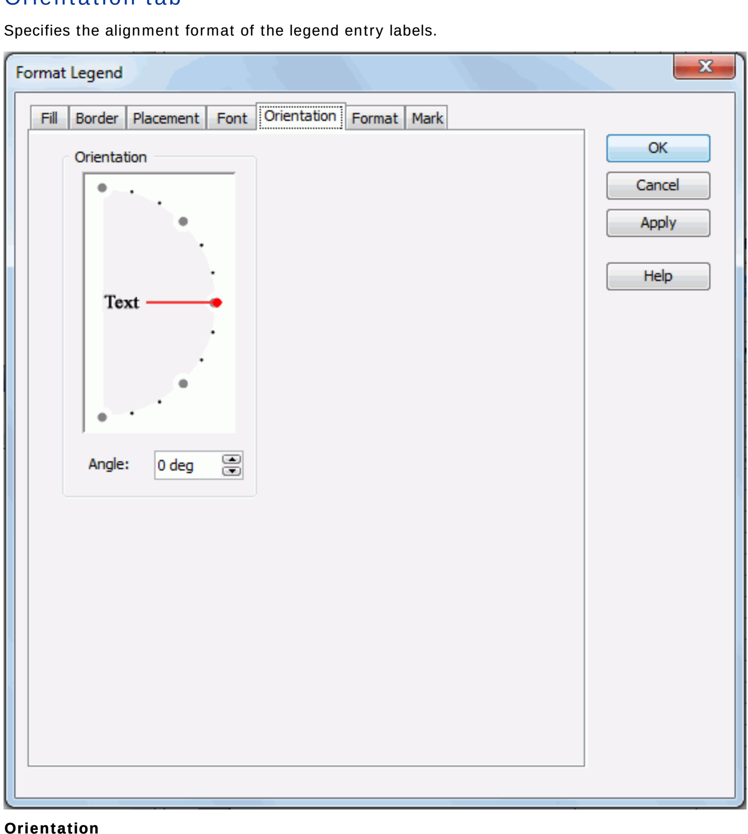
- **Style**
  Specifies the font style of the text. It can be one of the following: plain, bold, italic, and bold italic.
- **Strikethrough**
  Specifies the style of the horizontal line with which the text is struck through. It can be one of the following: none, thin line, bold line, and double lines.
- **Underline**
  Specifies the style of the horizontal line under the text. It can be one of the following: none, single, single lower, bold line, bold lower, double lines, bold double, patterned line, and bold patterned.
- **Superscript**
  Raises the text above the baseline and changes the text to a smaller font size, if a smaller size is available.
- **Embossed**
  Makes the text appear to be raised off the page in relief.
- **Outlined**
  Displays the inner and outer borders of each character.
- **Subscript**
  Lowers the text below the baseline and changes the text to a smaller font size, if a smaller size is available.
- **Engraved**
  Makes the text appear to be imprinted or pressed into the page.
- **Shadowed**
  Adds a shadow beneath and to the right of the text.

Sample

Displays the specified font and any text effects.

Orientation tab

Specifies the alignment format of the legend entry labels.



Orientation

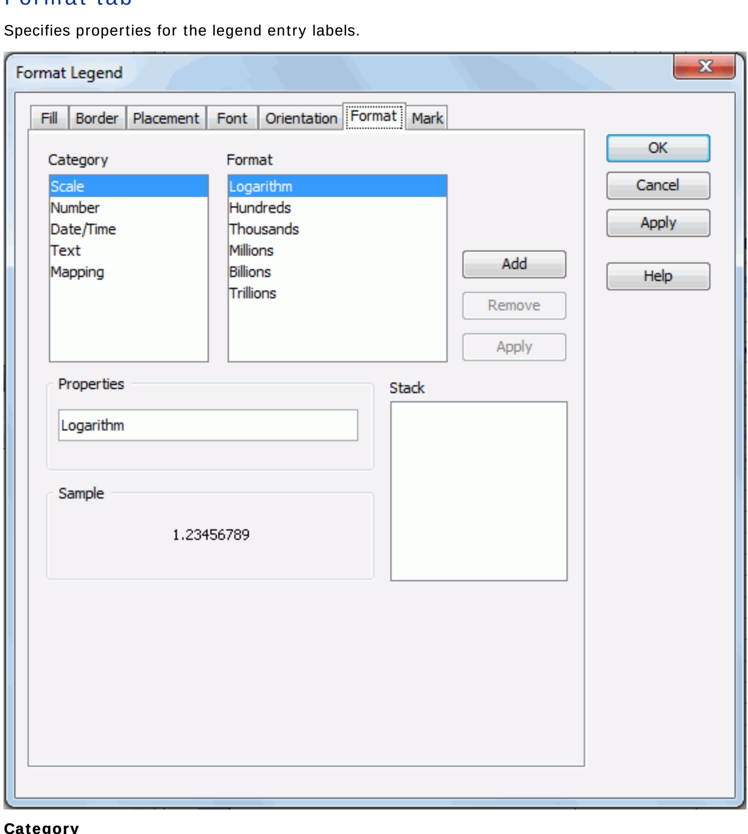
Specifies the orientation of the legend entry labels.

Angle

Specifies the angel of the legend entry labels.

Format tab

Specifies properties for the legend entry labels.



Category

Specifies different formats to display the legend entry labels.

- **Scale**
  Lists all the formats of a value of a scale type that can be applied to the legend entry labels.
- **Number**
  Lists all the formats of a value of a number type that can be applied to the legend entry labels.
- **Date/Time**
  Lists all the formats of a value of a date/time type that can be applied to the legend entry labels.
- **Text**
  Lists all the formats of a value of a text type that can be applied to the legend entry labels.
- **Mapping**
  Lists all the formats of a value of a mapping type that can be applied to the legend entry labels.

Format

Displays all the formats of the selected category. For more details about every format, see A detailed chart property reference.

Properties

Specifies the properties for the format that you selected.

Stack

Lists all the formats that you selected from different categories.

Sample

Displays the effects of the selected format that has been added into the Stack list box.

Add

Adds a format to the Stack list box.

Remove

Removes a format from the Stack list box.

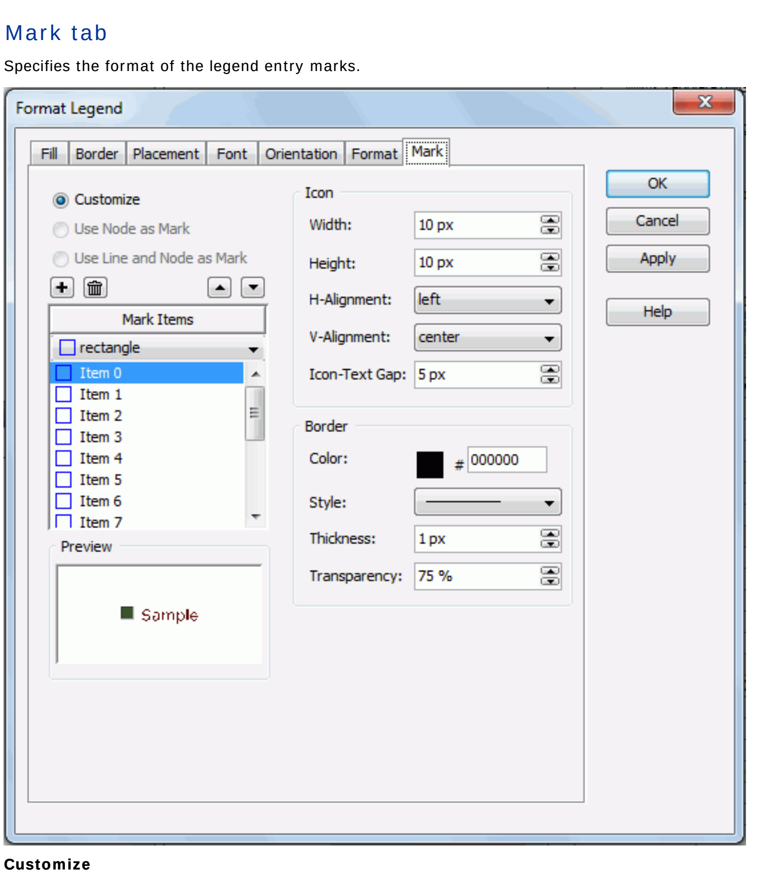
Apply

Applies the specified format in the Stack list box to the legend entry marks.

Note: To make settings in this tab take effect, you need to first make sure that the property Label Format Source of the chart legend has been set to Legend Label Format in the Report Inspector.

Mark tab

Specifies the format of the legend entry marks.



Customize

Specifies whether to customize the format of the legend entry marks.

Use Node as Mark

Specifies whether to apply the format of line nodes to the legend entry marks. If selected, the legend entry marks will automatically inherit the style and color of line nodes. Available for line charts only.

Use Line and Node as Mark

Specifies whether to apply the format of lines and line nodes to the legend entry marks. If checked, the legend entry marks will automatically inherit the style and color of lines and line nodes. Available for line charts only.

Mark Items

Specifies the style of the legend entry marks. Select the mark item from the box one by one and choose the style for each item from the drop-down list. Available only when Customize is checked.

- Adds a mark item to the box.
- Removes a specified mark item from the box.
- Moves a selected mark item up a step.
- Moves a selected mark item down a step.

Icon

Specifies the properties of the icons.

- **Width**
  Specifies the width of the icons, in pixels.
- **Height**
  Specifies the height of the icons, in pixels.
- **H-Alignment**
  Specifies the horizontal alignment format of the icons. It could be left, center and right.
- **V-Alignment**
  Specifies the vertical alignment format of the icons. It could be top, center and bottom.
- **Icon-Text Gap**
  Specifies the gap between each entry mark and entry label, in pixels.

Border

Specifies the border properties of the icons. Available only when Customize is checked.

- **Color**
  Specifies the color of the icon border.

  To change the color, click the color image button to display the color palette in which you can select a color, or click **More Colors** to customize the color within a wider range in the Pick a Color dialog.

  You can also input the hexadecimal value of a color directly in the text box after # or click to use a formula to control the color.
- **Style**
  Specifies the line style of the icon border.
- **Thickness**
  Specifies the thickness of the icon border, in pixels.
- **Transparency**
  Specifies the transparency of the icon border. Input the transparency directly in the text box or click to use a formula to control the color.

Preview

Displays the specified item and mark effects.

Related topics:

- Formatting the legend

Format Line dialog for library component

This dialog appears when you double-click a line in a chart in a library component, or right-click it and select Format Line from the shortcut menu. It helps you to format the selected chart line in the chart, and consists of the following tabs:

- General tab
- Fill tab
- Node tab
- Data Label tab
- Behaviors tab

OK

Applies the changes and closes the dialog.

Cancel

Does not retain any changes and closes the dialog.

Apply

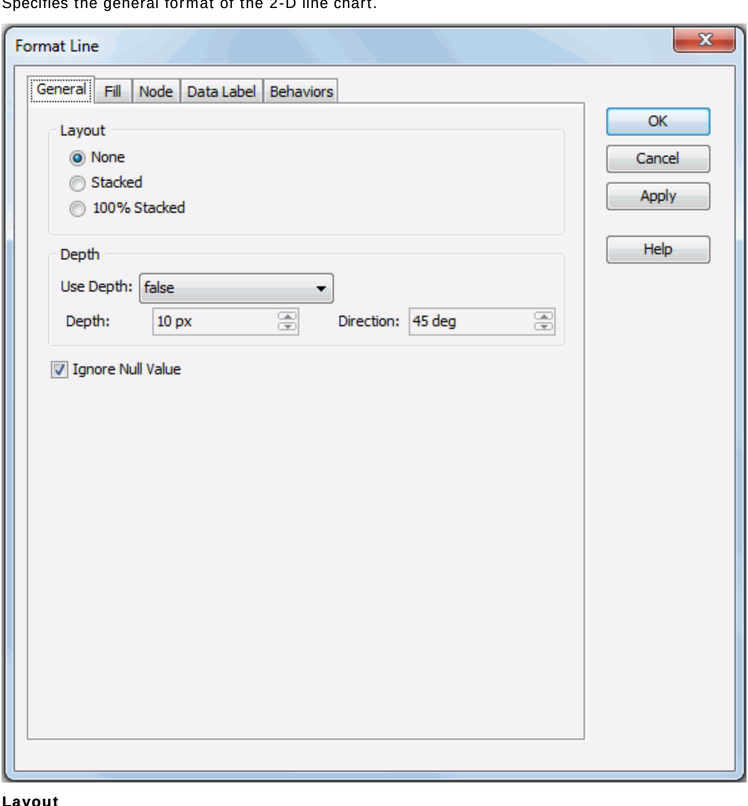
Applies all changes and leaves the dialog open.

Help

Displays the help document about this feature.

General tab

Specifies the general format of the 2-D line chart.



Layout

Specifies the layout of the 2-D lines in the chart.

- **None**
  If selected, the lines will display trend over categories.
- **Stacked**
  If selected, the stacked lines will display the trend of the contribution of each data value over categories.
- **100% Stacked**
  If selected, the 100% stacked lines will display the trend of the percentage each data value contributes over categories.

Depth

Specifies the depth properties for lines of the chart.

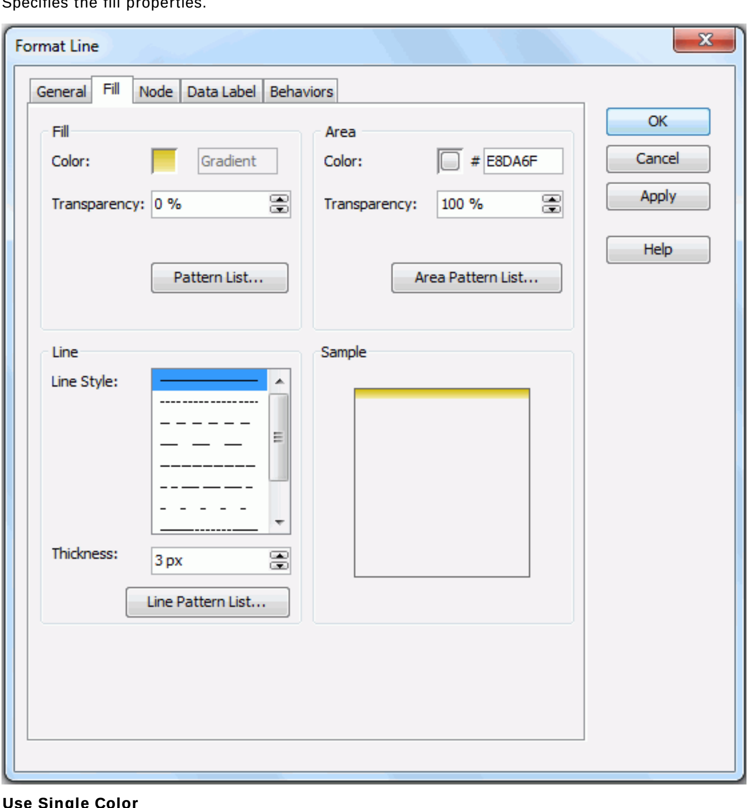
- **Use Depth**
  Specifies whether to make the lines in the chart three-dimensional. You can also use a formula to control the property.
  - **Depth**
    Specifies the depth of the lines, in pixels.
  - **Direction**
    Specifies the direction for depth of the lines, in degrees.

Ignore Null Value

Specifies whether to ignore null data values when drawing lines in the chart. If checked, when a null data value appears on a line, it is ignored and the line is drawn from the previous data value to the next data value directly, otherwise, the line is broken at the point of the null data value.

Fill tab

Specifies the fill properties.



Use Single Color

If checked, the line will use the single color pattern.

- **Fill**
  Specifies the fill effect of the chart lines.
  - **Color**
    Specifies the color schema for the selected chart line.

    To change the color, click the color image button to display the color palette in which you can select a color, or click **More Colors** to customize the color within a wider range in the Pick a Color dialog, or click **More Fill Effects** to customize in the Fill Effects dialog. You can also input the hexadecimal value of a color directly in the text box after # if you have changed the default color.
  - **Transparency**
    Specifies the transparency of the color schema.
  - **Pattern List**
    Opens the Color List dialog to modify color pattern for each line.
- **Area**
  Specifies properties for the areas which are formed by the chart axes and the lines. Applies to 2-D lines only.
  - **Color**
    Specifies the color schema for the selected chart line area.

    To change the color, click the color image button to display the color palette in which you can select a color, or click **More Colors** to customize the color within a wider range in the Pick a Color dialog, or click **More Fill Effects** to customize in the Fill Effects dialog. You can also input the hexadecimal value of a color directly in the text box after #.
  - **Transparency**
    Specifies the transparency of the color schema for the selected chart line area.
  - **Area Pattern List**
    Opens the Color List dialog to modify color pattern of the areas.
- **Line**
  Specifies the pattern of the chart lines.
  - **Line Style**
    Specifies the style of the selected chart line.
  - **Thickness**
    Specifies the thickness of the selected chart line, in pixels.
  - **Line Pattern List**
    Opens the Line Pattern List dialog to modify pattern of the lines in the same data series.

Use Single Color with Condition

Not supported on line chart.

Use Multiple Colors with Condition

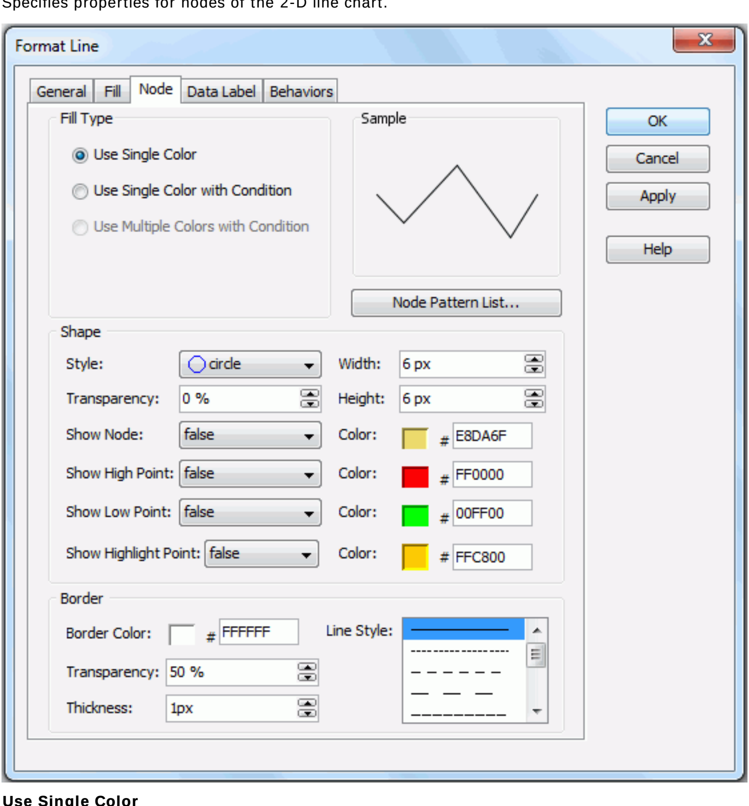
Not supported on line chart.

Sample

Displays a preview sample of your settings.

Node tab

Specifies properties for nodes of the 2-D line chart.



Use Single Color

If checked, the line nodes will use the single color pattern.

- **Shape**
  Specifies the shape properties of the nodes in the selected chart line.
  - **Style**
    Specifies the style of the nodes in the selected chart line.
  - **Transparency**
    Specifies the transparency of the nodes in the selected chart line.
  - **Width**
    Specifies the width of the nodes in the selected chart line, in pixels.
  - **Height**
    Specifies the height of the nodes in the selected chart line, in pixels.
  - **Show Node**
    Specifies whether to show nodes in the selected chart line. Only when it is set to true, the node related properties for the line chart can take effect. However, it does not control the properties Show High Point and Show Low Point and the Color property for each.
    - **Color**
      Specifies the color of the nodes in the selected chart line.

      To change the color, click the color image button to display the color palette in which you can select a color, or click **More Colors** to customize the color within a wider range in the Pick a Color dialog. You can also input the hexadecimal value of a color directly in the text box after #.
  - **Show High Point**
    Specifies whether to show the line node in the highest point in the selected chart line.
    - **Color**
      Specifies the color of the node in the highest point in the selected chart line.

      To change the color, click the color image button to display the color palette in which you can select a color, or click **More Colors** to customize the color within a wider range in the Pick a Color dialog. You can also input the hexadecimal value of a color directly in the text box after #.
  - **Show Low Point**
    Specifies whether to show the line node in the lowest point in the selected chart line.
    - **Color**
      Specifies the color of the node in the lowest point in the selected chart line.

      To change the color, click the color image button to display the color palette in which you can select a color, or click **More Colors** to customize the color within a wider range in the Pick a Color dialog. You can also input the hexadecimal value of a color directly in the text box after #.
  - **Show Highlight Point**
    Specifies whether to show the highlight point when you hover the mouse over a node in the selected chart line.
    - **Color**
      Specifies the color of the highlight point in the selected chart line.

      To change the color, click the color image button to display the color palette in which you can select a color, or click **More Colors** to customize the color within a wider range in the Pick a Color dialog. You can also input the hexadecimal value of a color directly in the text box after #.
- **Border**
  Specifies the border properties of the nodes in the selected chart line. Available when the node is not of the plus, multiplication, star1 or star2 style.
  - **Border Color**
    Specifies the color for the border of the nodes in the selected chart line.

    To change the color, click the color image button to display the color palette in which you can select a color, or click **More Colors** to customize the color within a wider range in the Pick a Color dialog. You can also input the hexadecimal value of a color directly in the text box after #.
  - **Transparency**
    Specifies the transparency for the border of the nodes in the selected chart line.
  - **Line Style**
    Specifies the line style for the border of the nodes in the selected chart line.
  - **Thickness**
    Specifies the thickness for the border of the nodes in the selected chart line, in pixels.
- **Line**
  Specifies the line properties of the nodes in the selected chart line. Available when the node is of the plus, multiplication, star1 or star2 style.
  - **Line Style**
    Specifies the line style of the nodes in the selected chart line.
  - **Thickness**
    Specifies the line thickness of the nodes in the selected chart line.
- **Node Pattern List**
  Opens the Node Pattern List dialog to modify the node pattern for each chart line.

Use Single Color with Condition

If checked, each node can have its single color pattern based on defined conditions. You can click the Advanced or Normal button to switch between the two editing modes to edit the conditions.

- **Normal**
  - **Select Field**
    Lists all the available fields to which the conditional fill can be applied. Select the field on which you want to define the conditions from the drop-down list.
  - **Values**
    Specifies the value of the condition. Available when you select a category or series field or a field whose return value is not a number from the Select Field drop-down list.
  - **Start Value**
    Specifies the start value of the condition. Available only when you select a value field or a field that returns a number from the Select Field drop-down list.
  - **End Value**
    Specifies the end value of the condition. Available only when you select a value field or a field that returns a number from the Select Field drop-down list.
  - **Color**
    Specifies the color that will be applied to the values which meet the condition.

    To change the color, click the color image button to display the color palette in which you can select a color, or click **More Colors** to customize the color within a wider range in the Pick a Color dialog, or click **More Fill Effects** to customize in the Fill Effects dialog. You can also input the hexadecimal value of a color directly in the text box after #.
  - **Transparency**
    Specifies the transparency for the color of the condition.
  - Adds a new condition line.
  - Removes the selected condition line.
  - **Label**
    If checked, you can modify the condition expression of the selected condition line which will be shown as the legend entry label.
  - **Value**
    Specifies to display data in the condition expression as value. Available only when you select a value field or a field that returns a number from the Select Field drop-down list.
  - **Percent**
    Specifies to display data in the condition expression in percent. Available only when you select a value field or a field that returns a number from the Select Field drop-down list and takes effect only on 100% Stacked Line 2-D chart.
- **Advanced**
  - **Condition**
    Displays the condition you have defined in the Edit Condition dialog.
  - **Color**
    Specifies the color that will be applied to the values which meet the condition.

    To change the color, click the color image button to display the color palette in which you can select a color, or click **More Colors** to customize the color within a wider range in the Pick a Color dialog, or click **More Fill Effects** to customize in the Fill Effects dialog. You can also input the hexadecimal value of a color directly in the text box after #.
  - **Transparency**
    Specifies the transparency for the color of the condition.
  - Opens the Edit Conditions dialog to create or edit a condition.
  - **Label**
    If checked, you can modify the condition expression of the selected condition line which will be shown as the legend entry label.
  - **Value**
    Specifies to display data in the condition expression as value. Available only when you select a value field or a field that returns a number from the Select Field drop-down list.
  - **Percent**
    Specifies to display data in the condition expression in percent. Available only when you select a value field or a field that returns a number from the Select Field drop-down list and takes effect only on 100% Stacked Line 2-D chart.

Use Multiple Colors with Condition

Not supported on line chart.

Apply Color to Line

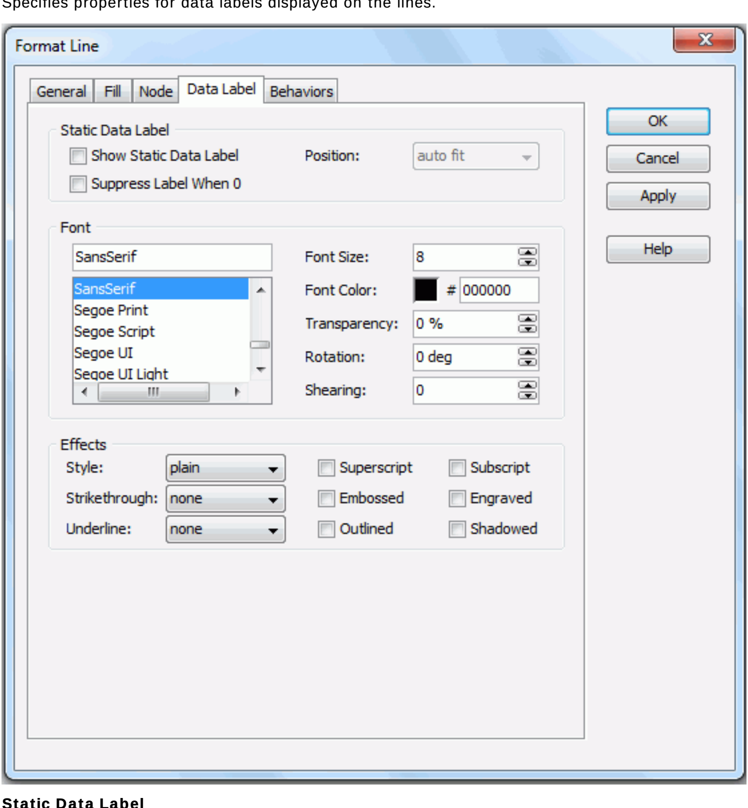
Not supported on line chart.

Sample

Displays a preview sample of your selection.

Data Label tab

Specifies properties for data labels displayed on the lines.



Static Data Label

Specifies properties of the static data labels on the lines.

- **Show Static Data Label**
  Specifies whether or not to show the static data labels on the lines. Only when it is checked can the following static data label related properties take effect.
- **Position**
  Specifies the position of the static data labels on the lines.
  - **auto fit**
    If selected, the static data labels will be displayed automatically.
  - **top center**
    If selected, the static data labels will be displayed on the top center of the line nodes.
  - **top left**
    If selected, the static data labels will be displayed on the top left of the line nodes.
  - **top right**
    If selected, the static data labels will be displayed on the top right of the line nodes.
  - **bottom left**
    If selected, the static data labels will be displayed at the bottom left of the line nodes.
  - **bottom center**
    If selected, the static data labels will be displayed at the bottom center of the line nodes.
  - **bottom right**
    If selected, the static data labels will be displayed at the bottom right of the line nodes.
- **Suppress Label When 0**
  If true, the static data label whose value is 0 will not be displayed on the chart.

Font

Specifies the font format of text in the data labels.

- **Font list**
  Lists all the available font faces that can be selected to apply to the text.
- **Font Size**
  Specifies the font size of the text.
- **Font Color**
  Specifies the font color of the text.

  To change the color, click the color image button to display the color palette in which you can select a color, or click **More Colors** to customize the color within a wider range in the Pick a Color dialog, or click **More Fill Effects** to customize in the Fill Effects dialog. You can also input the hexadecimal value of a color directly in the text box after #.
- **Transparency**
  Specifies the transparency of the text.
- **Rotation**
  Specifies the rotation angle of the text around its center, in degrees. The default value is 0.
- **Shearing**
  Specifies the gradient of the text.

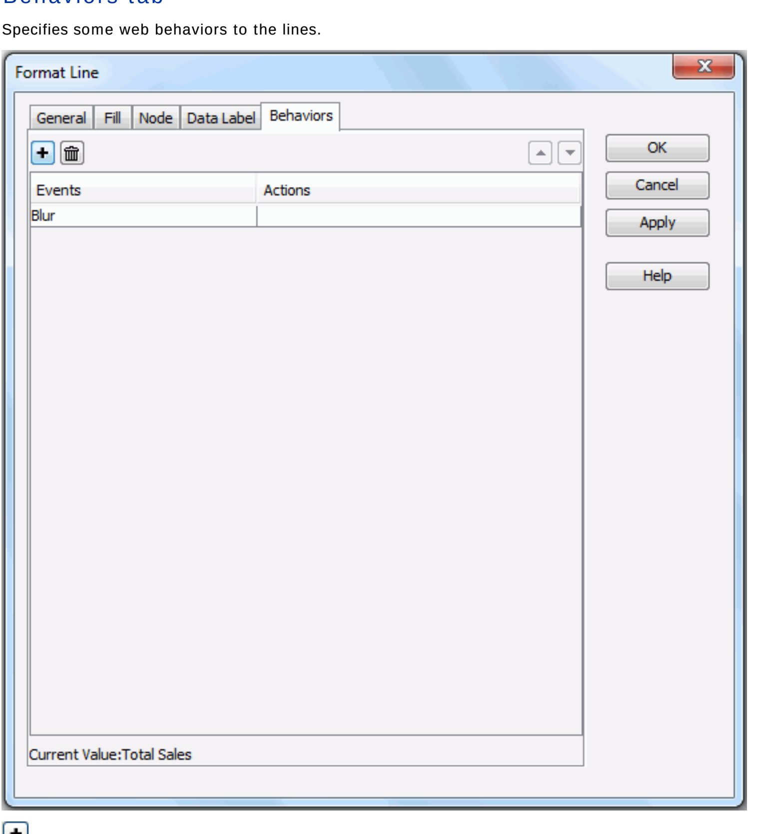
Effects

Specifies the special effects of text in the data labels.

- **Style**
  Specifies the font style of the text. It can be one of the following: plain, bold, italic, and bold italic.
- **Strikethrough**
  Specifies the style of the horizontal line with which the text is struck through. It can be one of the following: none, thin line, bold line, and double lines.
- **Underline**
  Specifies the style of the horizontal line under the text. It can be one of the following: none, single, single lower, bold line, bold lower, double lines, bold double, patterned line, and bold patterned.
- **Superscript**
  Raises the text above the baseline and changes the text to a smaller font size, if a smaller size is available.
- **Embossed**
  Makes the text appear to be raised off the page in relief.
- **Outlined**
  Displays the inner and outer borders of each character.
- **Subscript**
  Lowers the text below the baseline and changes the text to a smaller font size, if a smaller size is available.
- **Engraved**
  Makes the text appear to be imprinted or pressed into the page.
- **Shadowed**
  Adds a shadow beneath and to the right of the text.

Behaviors tab

Specifies some web behaviors to the lines.



Adds a new web behavior line.

Removes the selected web behavior.

Moves the selected web behavior up a step.

Moves the selected web behavior down a step.

Events

Specifies the trigger event.

Actions

Specifies the action you want the event to trigger.

- Opens the Web Action List dialog to bind a web action to the event.

Related topics:

- Formatting the lines

Format Line dialog for page report

This dialog appears when you double-click a line in a chart in a page report, or right-click it and select Format Line from the shortcut menu. It helps you to format the selected chart line in the chart, and consists of the following tabs:

- General tab
- Fill tab
- Node tab
- Border tab
- Data Label tab

OK

Applies the changes and closes the dialog.

Cancel

Does not retain any changes and closes the dialog.

Apply

Applies all changes and leaves the dialog open.

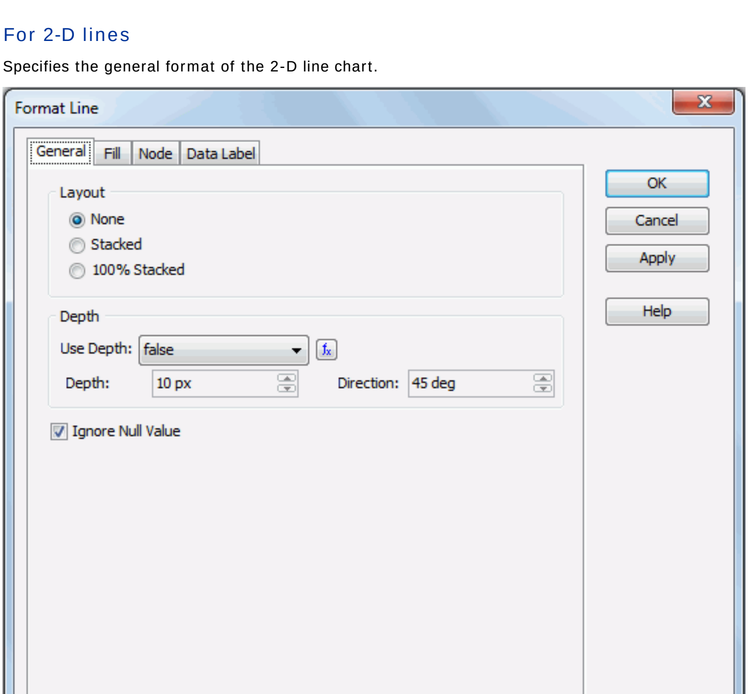
Help

Displays the help document about this feature.

General tab

For 2-D lines

Specifies the general format of the 2-D line chart.



Layout

Specifies the layout of the 2-D lines in the chart.

- **None**
  If selected, the lines will display trend over categories.
- **Stacked**
  If selected, the stacked lines will display the trend of the contribution of each data value over categories.
- **100% Stacked**
  If selected, the 100% stacked lines will display the trend of the percentage each data value contributes over categories.

Depth

Specifies the depth properties for lines of the chart.

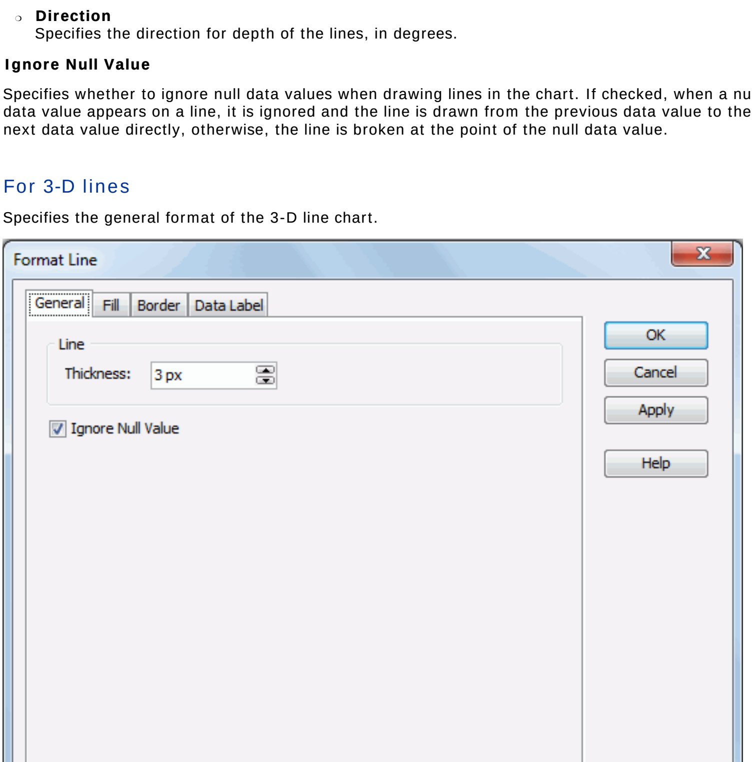
- **Use Depth**
  Specifies whether to make the lines in the chart three-dimensional. You can also use a formula to control the property.
  - **Depth**
    Specifies the depth of the lines, in pixels.
  - **Direction**
    Specifies the direction for depth of the lines, in degrees.

Ignore Null Value

Specifies whether to ignore null data values when drawing lines in the chart. If checked, when a null data value appears on a line, it is ignored and the line is drawn from the previous data value to the next data value directly, otherwise, the line is broken at the point of the null data value.

For 3-D lines

Specifies the general format of the 3-D line chart.



Thickness

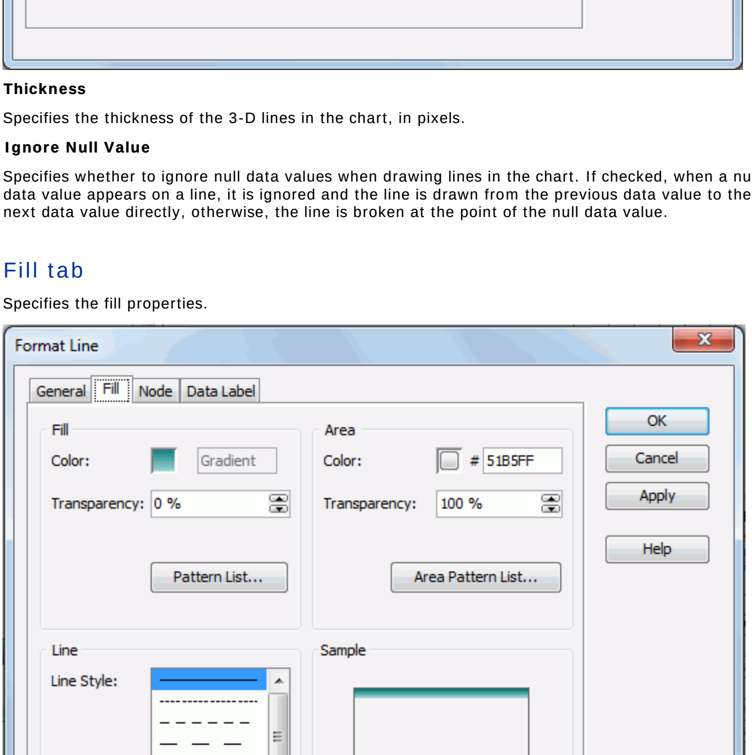
Specifies the thickness of the 3-D lines in the chart, in pixels.

Ignore Null Value

Specifies whether to ignore null data values when drawing lines in the chart. If checked, when a null data value appears on a line, it is ignored and the line is drawn from the previous data value to the next data value directly, otherwise, the line is broken at the point of the null data value.

Fill tab

Specifies the fill properties.



Use Single Color

If checked, the line will use the single color pattern.

- **Fill**
  Specifies the fill effect of the chart lines.
  - **Color**
    Specifies the color schema for the selected chart line.

    To change the color, click the color image button to display the color palette in which you can select a color, or click **More Colors** to customize the color within a wider range in the Pick a Color dialog, or click **More Fill Effects** to customize in the Fill Effects dialog. You can also input the hexadecimal value of a color directly in the text box after # if you have changed the default color.
  - **Transparency**
    Specifies the transparency of the color schema.
  - **Pattern List**
    Opens the Color List dialog to modify color pattern for each line.
- **Area**
  Specifies properties for the areas which are formed by the chart axes and the lines. Applies to 2-D lines only.
  - **Color**
    Specifies the color schema for the selected chart line area.

    To change the color, click the color image button to display the color palette in which you can select a color, or click **More Colors** to customize the color within a wider range in the Pick a Color dialog, or click **More Fill Effects** to customize in the Fill Effects dialog. You can also input the hexadecimal value of a color directly in the text box after #.
  - **Transparency**
    Specifies the transparency of the color schema for the selected chart line area.
  - **Area Pattern List**
    Opens the Color List dialog to modify color pattern of the areas.
- **Line**
  Specifies the pattern of the chart lines.
  - **Line Style**
    Specifies the style of the selected chart line.
  - **Thickness**
    Specifies the thickness of the selected chart line, in pixels.
  - **Line Pattern List**
    Opens the Line Pattern List dialog to modify pattern of the lines in the same data series.

Use Single Color with Condition

Not supported on line chart.

Use Multiple Colors with Condition

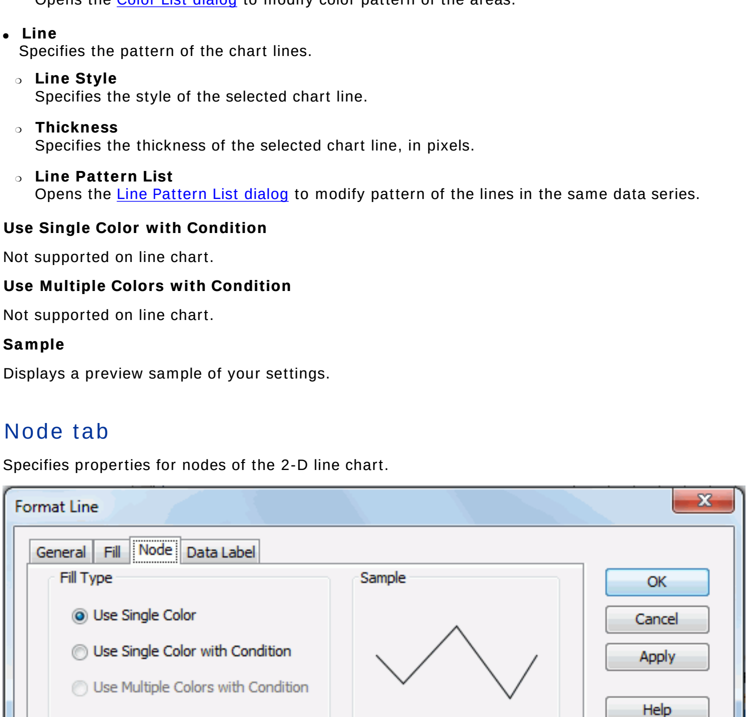
Not supported on line chart.

Sample

Displays a preview sample of your settings.

Node tab

Specifies properties for nodes of the 2-D line chart.



Use Single Color

If checked, the line nodes will use the single color pattern.

- **Shape**
  Specifies the shape properties of the nodes in the selected chart line.
  - **Style**
    Specifies the style of the nodes in the selected chart line.
  - **Transparency**
    Specifies the transparency of the nodes in the selected chart line.
  - **Width**
    Specifies the width of the nodes in the selected chart line, in pixels.
  - **Height**
    Specifies the height of the nodes in the selected chart line, in pixels.
  - **Show Node**
    Specifies whether to show nodes in the selected chart line. Only when it is set to true, the node related properties for the line chart can take effect. However, it does not control the properties Show High Point and Show Low Point and the Color property for each.
    - **Color**
      Specifies the color of the nodes in the selected chart line.

      To change the color, click the color image button to display the color palette in which you can select a color, or click **More Colors** to customize the color within a wider range in the Pick a Color dialog. You can also input the hexadecimal value of a color directly in the text box after # or click [fx] to use a formula to control the color.
  - **Show High Point**
    Specifies whether to show the line node in the highest point in the selected chart line.
    - **Color**
      Specifies the color of the node in the highest point in the selected chart line.

      To change the color, click the color image button to display the color palette in which you can select a color, or click **More Colors** to customize the color within a wider range in the Pick a Color dialog. You can also input the hexadecimal value of a color directly in the text box after # or click [fx] to use a formula to control the color.
  - **Show Low Point**
    Specifies whether to show the line node in the lowest point in the selected chart line.
    - **Color**
      Specifies the color of the node in the lowest point in the selected chart line.

      To change the color, click the color image button to display the color palette in which you can select a color, or click **More Colors** to customize the color within a wider range in the Pick a Color dialog. You can also input the hexadecimal value of a color directly in the text box after # or click [fx] to use a formula to control the color.
  - **Show Highlight Point**
    Specifies whether to show the highlight point when you hover the mouse over a node in the selected chart line.
    - **Color**
      Specifies the color of the highlight point in the selected chart line.

      To change the color, click the color image button to display the color palette in which you can select a color, or click **More Colors** to customize the color within a wider range in the Pick a Color dialog. You can also input the hexadecimal value of a color directly in the text box after # or.
- **Border**
  Specifies the border properties of the nodes in the selected chart line. Available when the node is not of the plus, multiplication, star1 or star2 style.
  - **Border Color**
    Specifies the color for the border of the nodes in the selected chart line.

    To change the color, click the color image button to display the color palette in which you can select a color, or click **More Colors** to customize the color within a wider range in the Pick a Color dialog. You can also input the hexadecimal value of a color directly in the text box after # or click [fx] to use a formula to control the color.
  - **Transparency**
    Specifies the transparency for the border of the nodes in the selected chart line. Input the transparency directly in the text box or click [fx] to use a formula to control the color.
  - **Line Style**
    Specifies the line style for the border of the nodes in the selected chart line.
  - **Thickness**
    Specifies the thickness for the border of the nodes in the selected chart line, in pixels.
- **Line**
  Specifies the line properties of the nodes in the selected chart line. Available when the node is of the plus, multiplication, star1 or star2 style.
  - **Line Style**
    Specifies the line style of the nodes in the selected chart line.
  - **Thickness**
    Specifies the line thickness of the nodes in the selected chart line.
- **Node Pattern List**
  Opens the Node Pattern List dialog to modify the node pattern for each chart line.

Use Single Color with Condition

If checked, each node can have its single color pattern based on defined conditions. You can click the Advanced or Normal button to switch between the two editing modes to edit the conditions.

- **Normal**
  - **Select Field**
    Lists all the available fields to which the conditional fill can be applied. Select the field on which you want to define the conditions from the drop-down list.
  - **Values**
    Specifies the value of the condition. Available when you select a category or series field or a field whose return value is not a number from the Select Field drop-down list.
  - **Start Value**
    Specifies the start value of the condition. Available only when you select a value field or a field that returns a number from the Select Field drop-down list.
  - **End Value**
    Specifies the end value of the condition. Available only when you select a value field or a field that returns a number from the Select Field drop-down list.
  - **Color**
    Specifies the color that will be applied to the values which meet the condition.

    To change the color, click the color image button to display the color palette in which you can select a color, or click **More Colors** to customize the color within a wider range in the Pick a Color dialog, or click **More Fill Effects** to customize in the Fill Effects dialog. You can also input the hexadecimal value of a color directly in the text box after #.
  - **Transparency**
    Specifies the transparency for the color of the condition.
  - [+]
    Adds a new condition line.
  - [Delete]
    Removes the selected condition line.
  - **Label**
    If checked, you can modify the condition expression of the selected condition line which will be shown as the legend entry label.
  - **Value**
    Specifies to display data in the condition expression as value. Available only when you select a value field or a field that returns a number from the Select Field drop-down list.
  - **Percent**
    Specifies to display data in the condition expression in percent. Available only when you select a value field or a field that returns a number from the Select Field drop-down list and takes effect only on 100% Stacked Line 2-D chart.
- **Advanced**
  - **Condition**
    Displays the condition you have defined in the Edit Condition dialog.
  - **Color**
    Specifies the color that will be applied to the values which meet the condition.

    To change the color, click the color image button to display the color palette in which you can select a color, or click **More Colors** to customize the color within a wider range in the Pick a Color dialog, or click **More Fill Effects** to customize in the Fill Effects dialog. You can also input the hexadecimal value of a color directly in the text box after #.
  - **Transparency**
    Specifies the transparency for the color of the condition.
  - [Edit]
    Opens the Edit Conditions dialog to create or edit a condition.
  - **Label**
    If checked, you can modify the condition expression of the selected condition line which will be shown as the legend entry label.
  - **Value**
    Specifies to display data in the condition expression as value. Available only when you select a value field or a field that returns a number from the Select Field drop-down list.
  - **Percent**
    Specifies to display data in the condition expression in percent. Available only when you select a value field or a field that returns a number from the Select Field drop-down list and takes effect only on 100% Stacked Line 2-D chart.

Use Multiple Colors with Condition

Not supported on line chart.

Apply Color to Line

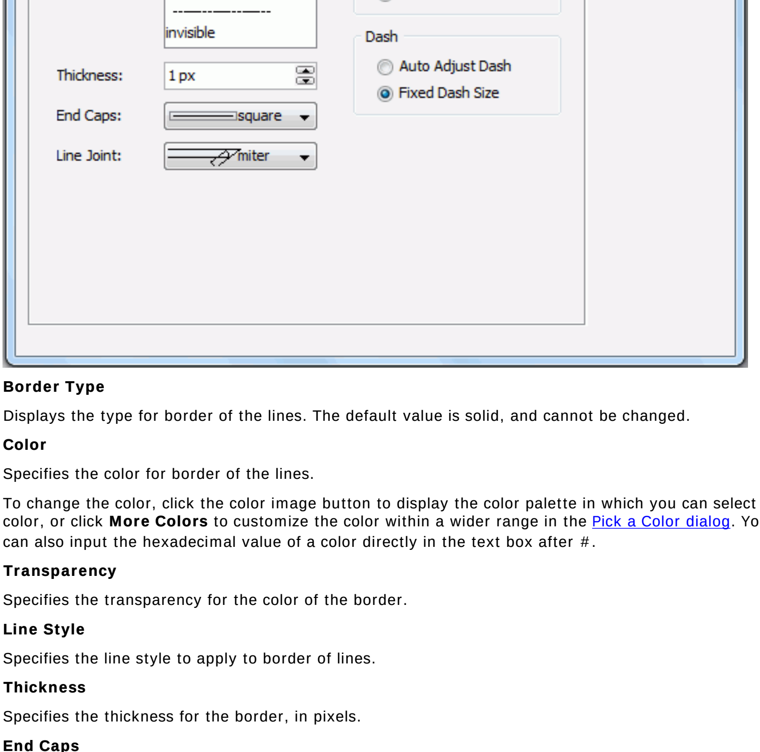
Not supported on line chart.

Sample

Displays a preview sample of your selection.

Border tab

Specifies properties for borders of the 3-D lines, which take effect only when the Border property on chart paper is set to true in the Report Inspector.



Border Type

Displays the type for border of the lines. The default value is solid, and cannot be changed.

Color

Specifies the color for border of the lines.

To change the color, click the color image button to display the color palette in which you can select a color, or click **More Colors** to customize the color within a wider range in the Pick a Color dialog. You can also input the hexadecimal value of a color directly in the text box after #.

Transparency

Specifies the transparency for the color of the border.

Line Style

Specifies the line style to apply to border of lines.

Thickness

Specifies the thickness for the border, in pixels.

End Caps

Specifies the ending style of the border line.

- **butt**
  Ends unclosed sub paths and dash segments with no added decoration.
- **round**
  Ends unclosed sub paths and dash segments with a round decoration that has a radius equal to half of the width of the pen.
- **square**
  Ends unclosed sub paths and dash segments with a square projection that extends beyond the end of the segment to a distance equal to half of the line width.

Line Joint

Specifies the line joint style for the border line.

- **miter**
  Joins path segments by extending their outside edges until they meet.
- **round**
  Joins path segments by rounding off the corner at a radius of half the line width.
- **bevel**
  Joins path segments by connecting the outer corners of their wide outlines with a straight segment.

Sample

Displays a preview sample of your selection.

Path

Specifies the fill pattern of the border line.

- **Outline Path**
  Specifies the fill pattern of the border line to be outline path.
- **Fill Path**
  Specifies the fill pattern of the border line to be whole path.

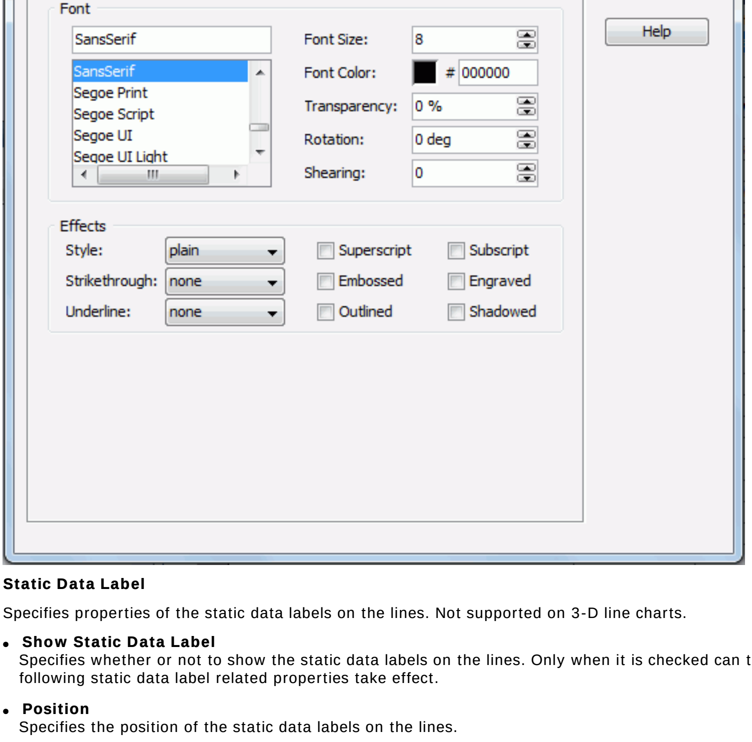
Dash

Specifies the dash size of border line.

- **Auto Adjusted Dash**
  If selected, the dash size will be adjusted automatically.
- **Fixed Dash Size**
  If selected, the dash size will be fixed size.

Data Label tab

Specifies properties for data labels displayed on the lines.



Static Data Label

Specifies properties of the static data labels on the lines. Not supported on 3-D line charts.

- **Show Static Data Label**
  Specifies whether or not to show the static data labels on the lines. Only when it is checked can the following static data label related properties take effect.
- **Position**
  Specifies the position of the static data labels on the lines.
  - **auto fit**
    If selected, the static data labels will be displayed automatically.
  - **top center**
    If selected, the static data labels will be displayed on the top center of the line nodes.
  - **top left**
    If selected, the static data labels will be displayed on the top left of the line nodes.
  - **top right**
    If selected, the static data labels will be displayed on the top right of the line nodes.
  - **bottom left**
    If selected, the static data labels will be displayed at the bottom left of the line nodes.
  - **bottom center**
    If selected, the static data labels will be displayed at the bottom center of the line nodes.
  - **bottom right**
    If selected, the static data labels will be displayed at the bottom right of the line nodes.
- **Suppress Label When 0**
  If true, the static data label whose value is 0 will not be displayed on the chart.

Font

Specifies the font format of text in the data labels.

- **Font list**
  Lists all the available font faces that can be selected to apply to the text.
- **Font Size**
  Specifies the font size of the text.
- **Font Color**
  Specifies the font color of the text.

  To change the color, click the color image button to display the color palette in which you can select a color, or click **More Colors** to customize the color within a wider range in the Pick a Color dialog, or click **More Fill Effects** to customize in the Fill Effects dialog. You can also input the hexadecimal value of a color directly in the text box after #.
- **Transparency**
  Specifies the transparency of the text.
- **Rotation**
  Specifies the rotation angle of the text around its center, in degrees. The default value is 0.
- **Shearing**
  Specifies the gradient of the text.

Effects

Specifies the special effects of text in the data labels.

- **Style**
  Specifies the font style of the text. It can be one of the following: plain, bold, italic, and bold italic.
- **Strikethrough**
  Specifies the style of the horizontal line with which the text is struck through. It can be one of the following: none, thin line, bold line, and double lines.
- **Underline**
  Specifies the style of the horizontal line under the text. It can be one of the following: none, single, single lower, bold line, bold lower, double lines, bold double, patterned line, and bold patterned.
- **Superscript**
  Raises the text above the baseline and changes the text to a smaller font size, if a smaller size is available.
- **Embossed**
  Makes the text appear to be raised off the page in relief.
- **Outlined**
  Displays the inner and outer borders of each character.
- **Subscript**
  Lowers the text below the baseline and changes the text to a smaller font size, if a smaller size is available.
- **Engraved**
  Makes the text appear to be imprinted or pressed into the page.
- **Shadowed**
  Adds a shadow beneath and to the right of the text.

Related topics:

- Formatting the lines

Format Line dialog for web report

This dialog appears when you double-click a line in a chart in a web report, or right-click it and select Format Line from the shortcut menu. It helps you to format the selected chart line in the chart, and consists of the following tabs:

- General tab
- Fill tab
- Node tab
- Data Label tab

OK

Applies the changes and closes the dialog.

Cancel

Does not retain any changes and closes the dialog.

Apply

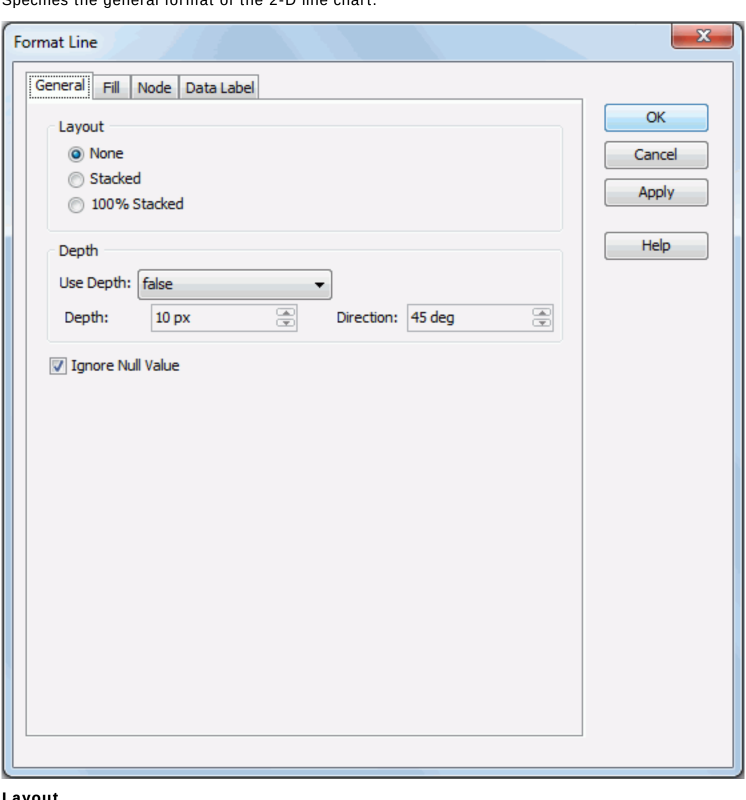
Applies all changes and leaves the dialog open.

Help

Displays the help document about this feature.

General tab

Specifies the general format of the 2-D line chart.



Layout

Specifies the layout of the 2-D lines in the chart.

- **None**
  If selected, the lines will display trend over categories.
- **Stacked**
  If selected, the stacked lines will display the trend of the contribution of each data value over categories.
- **100% Stacked**
  If selected, the 100% stacked lines will display the trend of the percentage each data value contributes over categories.

Depth

Specifies the depth properties for lines of the chart.

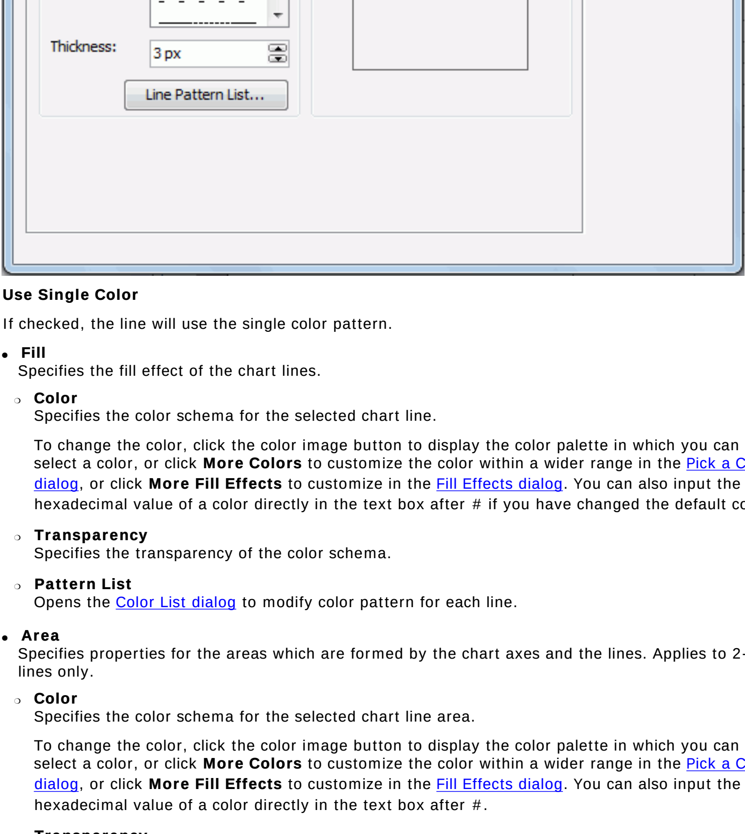
- **Use Depth**
  Specifies whether to make the lines in the chart three-dimensional. You can also use a formula to control the property.
  - **Depth**
    Specifies the depth of the lines, in pixels.
  - **Direction**
    Specifies the direction for depth of the lines, in degrees.

Ignore Null Value

Specifies whether to skip null data values in order to draw continuous lines in the chart. If checked, when a null data value appears on a line, it is ignored and the line is drawn from the previous data value to the next data value directly, otherwise, the line is broken at the point of the null data value.

Fill tab

Specifies the fill properties.



Use Single Color

If checked, the line will use the single color pattern.

- **Fill**
  Specifies the fill effect of the chart lines.
  - **Color**
    Specifies the color schema for the selected chart line.

    To change the color, click the color image button to display the color palette in which you can select a color, or click **More Colors** to customize the color within a wider range in the Pick a Color dialog, or click **More Fill Effects** to customize in the Fill Effects dialog. You can also input the hexadecimal value of a color directly in the text box after # if you have changed the default color.
  - **Transparency**
    Specifies the transparency of the color schema.
  - **Pattern List**
    Opens the Color List dialog to modify color pattern for each line.
- **Area**
  Specifies properties for the areas which are formed by the chart axes and the lines. Applies to 2-D lines only.
  - **Color**
    Specifies the color schema for the selected chart line area.

    To change the color, click the color image button to display the color palette in which you can select a color, or click **More Colors** to customize the color within a wider range in the Pick a Color dialog, or click **More Fill Effects** to customize in the Fill Effects dialog. You can also input the hexadecimal value of a color directly in the text box after #.
  - **Transparency**
    Specifies the transparency of the color schema for the selected chart line area.
  - **Area Pattern List**
    Opens the Color List dialog to modify color pattern of the areas.
- **Line**
  Specifies the pattern of the chart lines.
  - **Line Style**
    Specifies the style of the selected chart line.
  - **Thickness**
    Specifies the thickness of the selected chart line, in pixels.
  - **Line Pattern List**
    Opens the Line Pattern List dialog to modify pattern of the lines in the same data series.

Use Single Color with Condition

Not supported on line chart.

Use Multiple Colors with Condition

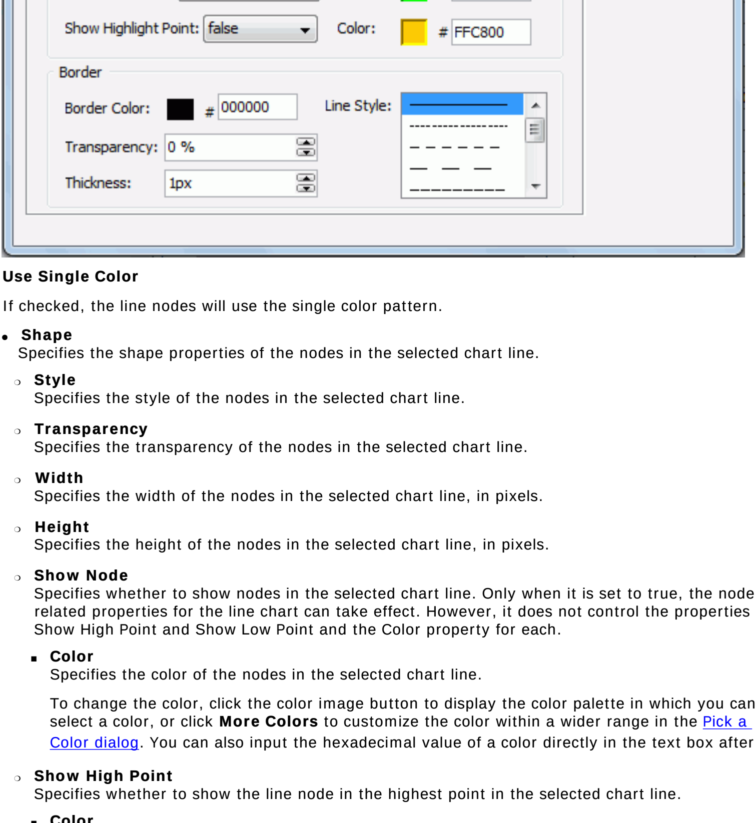
Not supported on line chart.

Sample

Displays a preview sample of your settings.

Node tab

Specifies properties for nodes of the 2-D line chart.



Use Single Color

If checked, the line nodes will use the single color pattern.

- **Shape**
  Specifies the shape properties of the nodes in the selected chart line.
  - **Style**
    Specifies the style of the nodes in the selected chart line.
  - **Transparency**
    Specifies the transparency of the nodes in the selected chart line.
  - **Width**
    Specifies the width of the nodes in the selected chart line, in pixels.
  - **Height**
    Specifies the height of the nodes in the selected chart line, in pixels.
  - **Show Node**
    Specifies whether to show nodes in the selected chart line. Only when it is set to true, the node related properties for the line chart can take effect. However, it does not control the properties Show High Point and Show Low Point and the Color property for each.
    - **Color**
      Specifies the color of the nodes in the selected chart line.

      To change the color, click the color image button to display the color palette in which you can select a color, or click **More Colors** to customize the color within a wider range in the Pick a Color dialog. You can also input the hexadecimal value of a color directly in the text box after #.
  - **Show High Point**
    Specifies whether to show the line node in the highest point in the selected chart line.
    - **Color**
      Specifies the color of the node in the highest point in the selected chart line.

      To change the color, click the color image button to display the color palette in which you can select a color, or click **More Colors** to customize the color within a wider range in the Pick a Color dialog. You can also input the hexadecimal value of a color directly in the text box after #.
  - **Show Low Point**
    Specifies whether to show the line node in the lowest point in the selected chart line.
    - **Color**
      Specifies the color of the node in the lowest point in the selected chart line.

      To change the color, click the color image button to display the color palette in which you can select a color, or click **More Colors** to customize the color within a wider range in the Pick a Color dialog. You can also input the hexadecimal value of a color directly in the text box after #.
  - **Show Highlight Point**
    Specifies whether to show the highlight point when you hover the mouse over a node in the selected chart line.
    - **Color**
      Specifies the color of the highlight point in the selected chart line.

      To change the color, click the color image button to display the color palette in which you can select a color, or click **More Colors** to customize the color within a wider range in the Pick a Color dialog. You can also input the hexadecimal value of a color directly in the text box after #.
- **Border**
  Specifies the border properties of the nodes in the selected chart line. Available when the node is not of the plus, multiplication, star1 or star2 style.
  - **Border Color**
    Specifies the color for the border of the nodes in the selected chart line.

    To change the color, click the color image button to display the color palette in which you can select a color, or click **More Colors** to customize the color within a wider range in the Pick a Color dialog. You can also input the hexadecimal value of a color directly in the text box after #.
  - **Transparency**
    Specifies the transparency for the border of the nodes in the selected chart line.
  - **Line Style**
    Specifies the line style for the border of the nodes in the selected chart line.
  - **Thickness**
    Specifies the thickness for the border of the nodes in the selected chart line, in pixels.
- **Line**
  Specifies the line properties of the nodes in the selected chart line. Available when the node is of the plus, multiplication, star1 or star2 style.
  - **Line Style**
    Specifies the line style of the nodes in the selected chart line.
  - **Thickness**
    Specifies the line thickness of the nodes in the selected chart line.
- **Node Pattern List**
  Opens the Node Pattern List dialog to modify the node pattern for each chart line.

Use Single Color with Condition

If checked, each node can have its single color pattern based on defined conditions. You can click the Advanced or Normal button to switch between the two editing modes to edit the conditions.

- **Normal**
  - **Select Field**
    Lists all the available fields to which the conditional fill can be applied. Select the field on which you want to define the conditions from the drop-down list.
  - **Values**
    Specifies the value of the condition. Available when you select a category or series field or a field whose return value is not a number from the Select Field drop-down list.
  - **Start Value**
    Specifies the start value of the condition. Available only when you select a value field or a field that returns a number from the Select Field drop-down list.
  - **End Value**
    Specifies the end value of the condition. Available only when you select a value field or a field that returns a number from the Select Field drop-down list.
  - **Color**
    Specifies the color that will be applied to the values which meet the condition.

    To change the color, click the color image button to display the color palette in which you can select a color, or click **More Colors** to customize the color within a wider range in the Pick a Color dialog, or click **More Fill Effects** to customize in the Fill Effects dialog. You can also input the hexadecimal value of a color directly in the text box after #.
  - **Transparency**
    Specifies the transparency for the color of the condition.
  - Adds a new condition line.
  - Removes the selected condition line.
  - **Label**
    If checked, you can modify the condition expression of the selected condition line which will be shown as the legend entry label.
  - **Value**
    Specifies to display data in the condition expression as value. Available only when you select a value field or a field that returns a number from the Select Field drop-down list.
  - **Percent**
    Specifies to display data in the condition expression in percent. Available only when you select a value field or a field that returns a number from the Select Field drop-down list and takes effect only on 100% Stacked Line 2-D chart.
- **Advanced**
  - **Condition**
    Displays the condition you have defined in the Edit Condition dialog.
  - **Color**
    Specifies the color that will be applied to the values which meet the condition.

    To change the color, click the color image button to display the color palette in which you can select a color, or click **More Colors** to customize the color within a wider range in the Pick a Color dialog, or click **More Fill Effects** to customize in the Fill Effects dialog. You can also input the hexadecimal value of a color directly in the text box after #.
  - **Transparency**
    Specifies the transparency for the color of the condition.
  - Opens the Edit Conditions dialog to create or edit a condition.
  - **Label**
    If checked, you can modify the condition expression of the selected condition line which will be shown as the legend entry label.
  - **Value**
    Specifies to display data in the condition expression as value. Available only when you select a value field or a field that returns a number from the Select Field drop-down list.
  - **Percent**
    Specifies to display data in the condition expression in percent. Available only when you select a value field or a field that returns a number from the Select Field drop-down list and takes effect only on 100% Stacked Line 2-D chart.

Use Multiple Colors with Condition

Not supported on line chart.

Apply Color to Line

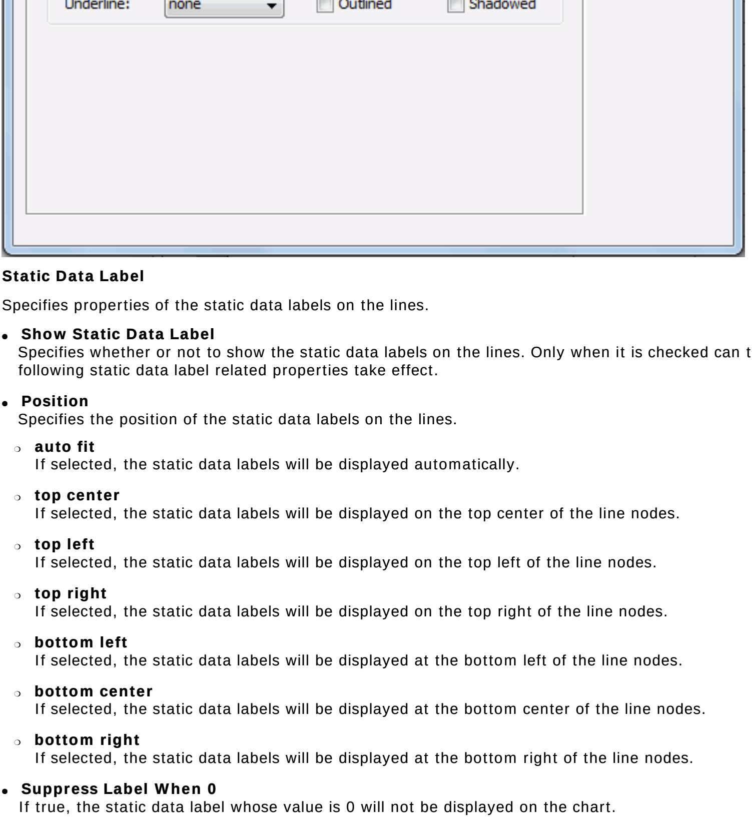
Not supported on line chart.

Sample

Displays a preview sample of your selection.

Data Label tab

Specifies properties for data labels displayed on the lines.



Static Data Label

Specifies properties of the static data labels on the lines.

- **Show Static Data Label**
  Specifies whether or not to show the static data labels on the lines. Only when it is checked can the following static data label related properties take effect.
- **Position**
  Specifies the position of the static data labels on the lines.
  - **auto fit**
    If selected, the static data labels will be displayed automatically.
  - **top center**
    If selected, the static data labels will be displayed on the top center of the line nodes.
  - **top left**
    If selected, the static data labels will be displayed on the top left of the line nodes.
  - **top right**
    If selected, the static data labels will be displayed on the top right of the line nodes.
  - **bottom left**
    If selected, the static data labels will be displayed at the bottom left of the line nodes.
  - **bottom center**
    If selected, the static data labels will be displayed at the bottom center of the line nodes.
  - **bottom right**
    If selected, the static data labels will be displayed at the bottom right of the line nodes.
- **Suppress Label When 0**
  If true, the static data label whose value is 0 will not be displayed on the chart.

Font

Specifies the font format of text in the data labels.

- **Font list**
  Lists all the available font faces that can be selected to apply to the text.
- **Font Size**
  Specifies the font size of the text.
- **Font Color**
  Specifies the font color of the text.

  To change the color, click the color image button to display the color palette in which you can select a color, or click **More Colors** to customize the color within a wider range in the Pick a Color dialog, or click **More Fill Effects** to customize in the Fill Effects dialog. You can also input the hexadecimal value of a color directly in the text box after #.
- **Transparency**
  Specifies the transparency of the text.
- **Rotation**
  Specifies the rotation angle of the text around its center, in degrees. The default value is 0.
- **Shearing**
  Specifies the gradient of the text.

Effects

Specifies the special effects of text in the data labels.

- **Style**
  Specifies the font style of the text. It can be one of the following: plain, bold, italic, and bold italic.
- **Strikethrough**
  Specifies the style of the horizontal line with which the text is struck through. It can be one of the following: none, thin line, bold line, and double lines.
- **Underline**
  Specifies the style of the horizontal line under the text. It can be one of the following: none, single, single lower, bold line, bold lower, double lines, bold double, patterned line, and bold patterned.
- **Superscript**
  Raises the text above the baseline and changes the text to a smaller font size, if a smaller size is available.
- **Embossed**
  Makes the text appear to be raised off the page in relief.
- **Outlined**
  Displays the inner and outer borders of each character.
- **Subscript**
  Lowers the text below the baseline and changes the text to a smaller font size, if a smaller size is available.
- **Engraved**
  Makes the text appear to be imprinted or pressed into the page.
- **Shadowed**
  Adds a shadow beneath and to the right of the text.

Related topics:

- Formatting the lines

Format Paper dialog

This dialog appears when you double-click a paper of a chart, or right-click a chart and then select Format Paper from the shortcut menu. The paper of a chart is the canvas on which the main chart is drawn.

This dialog allows you to format the paper in the chart, and consists of the following tabs:

- General tab (for 2-D charts)
- Rotation tab (for 3-D charts)
- Fill tab
- Border tab

OK

Applies the changes and closes the dialog.

Cancel

Does not retain any changes and closes the dialog.

Apply

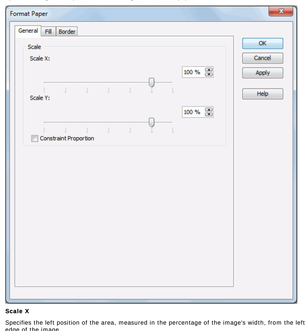
Applies all changes and leaves the dialog open.

Help

Displays the help document about this feature.

General tab (for 2-D charts)

Specifies the general options for formatting the 2-D chart paper.



Scale X

Specifies the left position of the area, measured in the percentage of the image's width, from the left edge of the image.

Scale Y

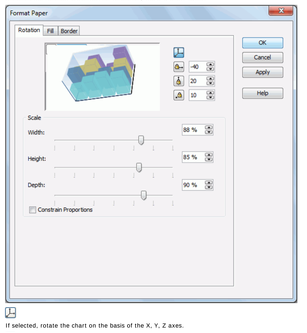
Specifies the bottom position of the image, measured in the percentage of the image's height, from the bottom edge of the image.

Constraint Proportion

Specifies whether to set the same scaling ratios for both X and Y axes.

Rotation tab (for 3-D charts)

Specifies the rotation properties for formatting the 3-D chart paper.



If selected, rotate the chart on the basis of the X, Y, Z axes.

If selected, rotate the chart just on the basis of the X axis.

If selected, rotate the chart just on the basis of the Y axis.

If selected, rotate the chart just on the basis of the Z axis.

Width

Specifies the scaling ratios of the width for the paper.

Height

Specifies the scaling ratios of the height for the paper.

Depth

Specifies the scaling ratios of the depth for the paper.

Constraint Proportion

Specifies whether to set the same scaling ratios for width, height and depth of the paper.

Fill tab

Specifies the color to fill the chart paper.

Color

Specifies the color to fill the paper.

To change the color, click the color image button to display the color palette in which you can select a color, or click **More Colors** to customize the color within a wider range in the Pick a Color dialog, or click **More Fill Effects** to customize in the Fill Effects dialog. You can also input the hexadecimal value of a color directly in the text box after # if you have changed the default color.

Transparency

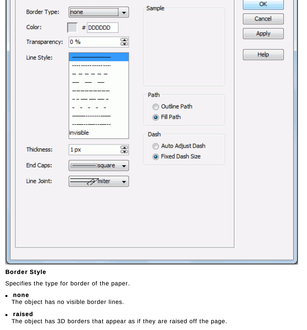
Specifies the transparency of the color.

Sample

Displays the specified color and transparency effects.

Border tab

Specifies properties for border of the chart paper.



Border Style

Specifies the type for border of the paper.

- **none**
  The object has no visible border lines.
- **raised**
  The object has 3D borders that appear as if they are raised off the page.
- **recess**
  The object has 3D borders that appear as if they are pressed into the page.
- **shadow**
  The object has two shadowed borders, beneath and to the right of the object.
- **solid**
  The object has single-line borders.

Color

Specifies the color for border of the chart paper.

To change the color, click the color image button to display the color palette in which you can select a color, or click **More Colors** to customize the color within a wider range in the Pick a Color dialog. You can also input the hexadecimal value of a color directly in the text box after #.

Line Style

Specifies the line style to apply to border of the chart paper.

Thickness

Specifies the thickness of the border, in pixels.

Transparency

Specifies the transparency for the color of the border.

End Caps

Specifies the ending style of the border line.

- **Butt**
  Ends unclosed sub paths and dash segments with no added decoration.
- **Round**
  Ends unclosed sub paths and dash segments with a round decoration that has a radius equal to half of the width of the pen.
- **Square**
  Ends unclosed sub paths and dash segments with a square projection that extends beyond the end of the segment to a distance equal to half of the line width.

Line Joint

Specifies the line joint style for the border line.

- **Miter**
  Joins path segments by extending their outside edges until they meet.
- **Round**
  Joins path segments by rounding off the corner at a radius of half the line width.
- **Bevel**
  Joins path segments by connecting the outer corners of their wide outlines with a straight segment.

Path

Specifies the fill pattern of the border line.

- **Outline Path**
  Specifies the fill pattern of the border line to be outline path.
- **Fill Path**
  Specifies the fill pattern of the border line to be whole path.

Dash

Specifies the dash size of the border line.

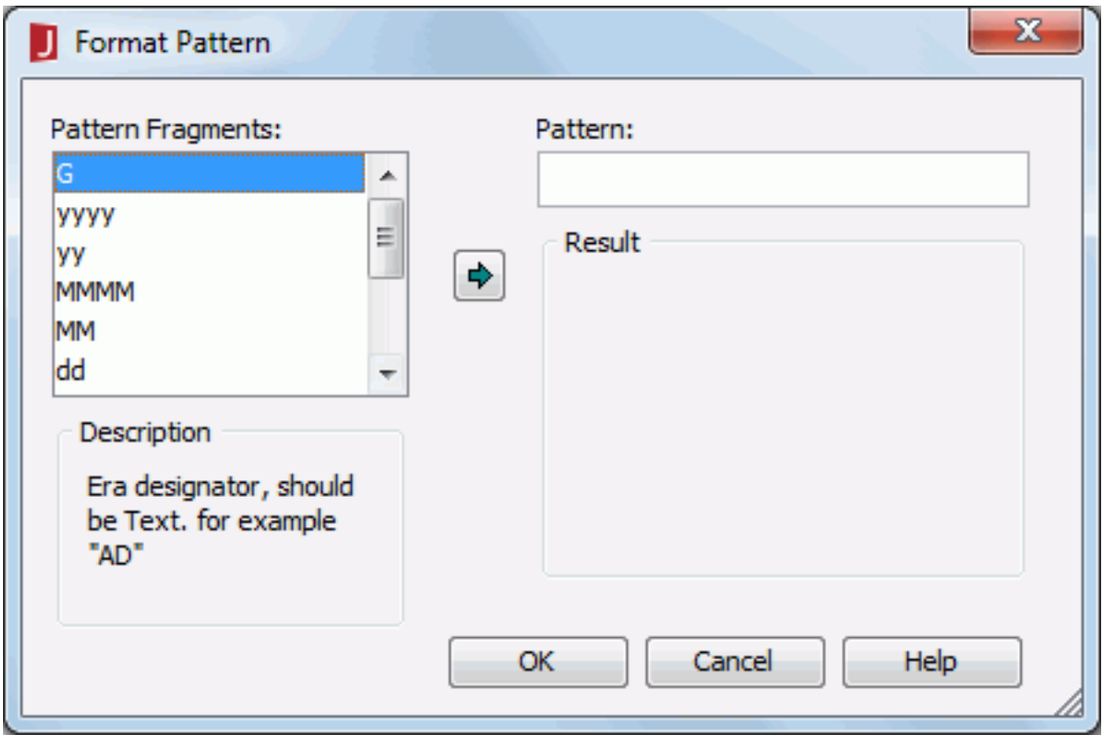
- **Auto Adjusted Dash**
  If selected, the dash size will be adjusted automatically.
- **Fixed Dash Size**
  If selected, the dash size will be fixed size.

Related topics:

- Formatting the paper

Format Pattern dialog

This dialog appears when you set the Format Pattern Type property to be Customize or click ... to modify the Format Pattern property in the Modify Schema Properties screen of the XML Connection Wizard. It helps you to specify patterns to format the Date, Time or DateTime data in the XML schema.



The following are details about options in the dialog:

Pattern Fragments

Lists all the pattern fragments that are supported by JReport Designer.

Description

Displays the detail descriptions of the selected pattern fragment.

Adds a selected pattern fragment to the Pattern box.

Pattern

Specifies the pattern to format the Date, Time or DateTime data in the XML schema.

To specify a pattern, select a pattern fragment and click to add it to the Pattern box, or double-click a pattern fragment to add it to the Pattern box. You can add several fragments as you want and any character can be input manually between two pattern fragments as a delimiter. For more details about the patterns, see Appendix 1: Details of format patterns.

Result

Displays a sample of the current date and time using the patterns you defined in the Pattern box.

OK

Accepts the changes and closes the dialog.

Cancel

Does not retain the changes and closes the dialog.

Help

Displays the help document about this feature.

Format Pie dialog for library component

This dialog appears when you double-click a pie of a chart in a library component, or right-click it and select Format Pie from the shortcut menu. It helps you to format pies of a chart, and consists of the following tabs:

- General tab
- Fill tab
- Border tab
- Data Label tab
- Behaviors tab

OK

Applies the changes and closes the dialog.

Cancel

Does not retain any changes and closes the dialog.

Apply

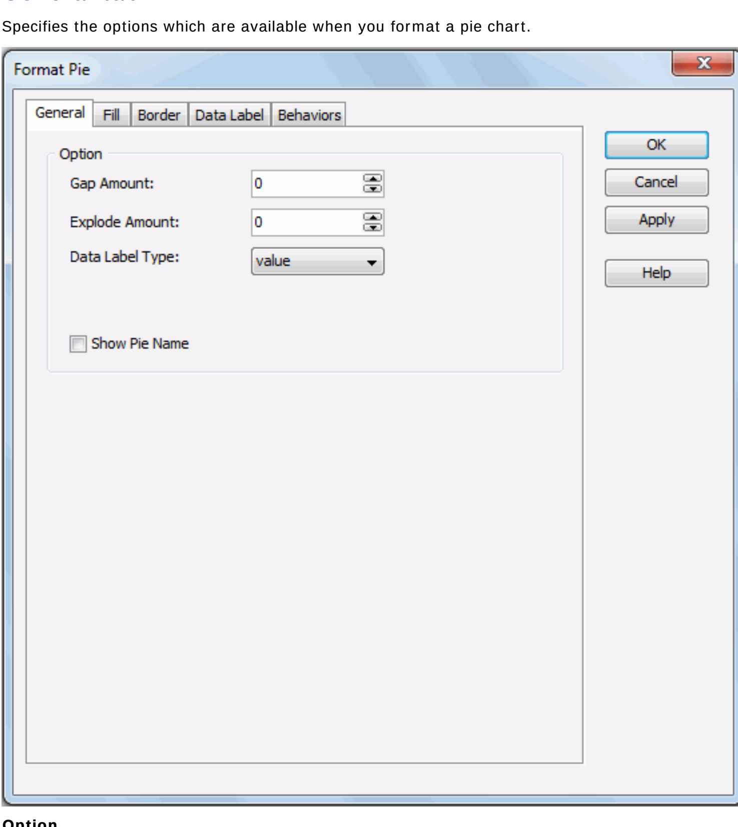
Applies all changes and leaves the dialog open.

Help

Displays the help document about this feature.

General tab

Specifies the options which are available when you format a pie chart.

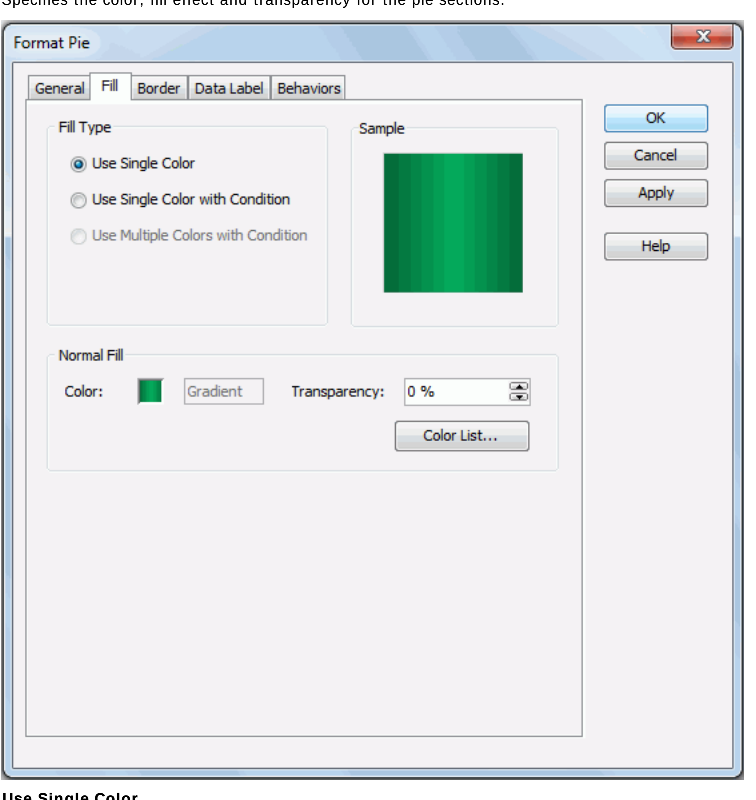


Option

- **Gap Amount**
  Specifies the gap between two pies.
- **Explode Amount**
  Specifies the gap between every section and the central point of a pie.
- **Data Label Type**
  Specifies the display type for data value in the data label. It can be value, category name, percent, value and percent.
- **Show Pie Name**
  Specifies whether to show the name of the pie.

Fill tab

Specifies the color, fill effect and transparency for the pie sections.



Use Single Color

If checked, the pie sections will use the single color pattern.

- **Color**
  Specifies the color schema for the selected pie sections in the same data series.

  To change the color, click the color image button to display the color palette in which you can select a color, or click **More Colors** to customize the color within a wider range in the Pick a Color dialog, or click **More Fill Effects** to customize in the Fill Effects dialog. You can also input the hexadecimal value of a color directly in the text box after # if you have changed the default color.
- **Transparency**
  Specifies the transparency of the color schema.
- **Color List**
  Opens the Color List dialog to modify color pattern for pie sections in the same data series.

Use Single Color with Condition

If checked, each pie section can have its single color pattern based on defined conditions. You can click the Advanced or Normal button to switch between the two editing modes to edit the conditions.

- **Normal**
  - **Select Field**
    Lists all the available fields to which the conditional fill can be applied. Select the field on which you want to define the conditions from the drop-down list.
  - **Values**
    Specifies the value of the condition. Available when you select a category or series field or a field whose return value is not a number from the Select Field drop-down list.
  - **Start Value**
    Specifies the start value of the condition. Available only when you select a value field or a field that returns a number from the Select Field drop-down list.
  - **End Value**
    Specifies the end value of the condition. Available only when you select a value field or a field that returns a number from the Select Field drop-down list.
  - **Color**
    Specifies the color that will be applied to the values which meet the condition.

    To change the color, click the color image button to display the color palette in which you can select a color, or click **More Colors** to customize the color within a wider range in the Pick a Color dialog, or click **More Fill Effects** to customize in the Fill Effects dialog. You can also input the hexadecimal value of a color directly in the text box after #.
  - **Transparency**
    Specifies the transparency for the color of the condition.
  - Adds a new condition line.
  - Removes the selected condition line.
  - **Label**
    If checked, you can modify the condition expression of the selected condition line which will be shown as the legend entry label.
  - **Value**
    Specifies to display data in the condition expression as value. Available only when you select a value field or a field that returns a number from the Select Field drop-down list.
  - **Percent**
    Specifies to display data in the condition expression in percent. Available only when you select a value field or a field that returns a number from the Select Field drop-down list.
- **Advanced**
  - **Condition**
    Displays the condition you have defined in the Edit Condition dialog.
  - **Color**
    Specifies the color that will be applied to the values which meet the condition.

    To change the color, click the color image button to display the color palette in which you can select a color, or click **More Colors** to customize the color within a wider range in the Pick a Color dialog, or click **More Fill Effects** to customize in the Fill Effects dialog. You can also input the hexadecimal value of a color directly in the text box after #.
  - **Transparency**
    Specifies the transparency for the color of the condition.
  - Opens the Edit Conditions dialog to create or edit a condition.
  - **Label**
    If checked, you can modify the condition expression of the selected condition line which will be shown as the legend entry label.
  - **Value**
    Specifies to display data in the condition expression as value. Available only when you select a value field or a field that returns a number from the Select Field drop-down list.
  - **Percent**
    Specifies to display data in the condition expression in percent. Available only when you select a value field or a field that returns a number from the Select Field drop-down list.

Use Multiple Colors with Condition

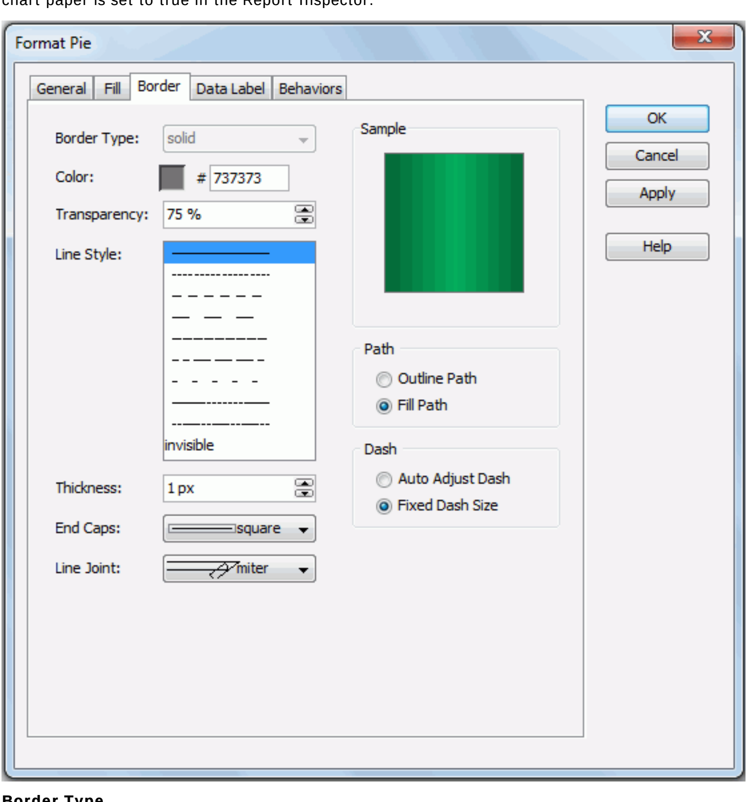
Not supported on pie chart.

Sample

Displays a preview sample of your selection.

Border tab

Specifies the properties for the border of the pies, which take effect only when the Border property on chart paper is set to true in the Report Inspector.



Border Type

Displays the type for border of the pies. The default value is solid, and cannot be changed.

Color

Specifies the color for the border of the pies.

To change the color, click the color image button to display the color palette in which you can select a color, or click **More Colors** to customize the color within a wider range in the Pick a Color dialog. You can also input the hexadecimal value of a color directly in the text box after #.

Transparency

Specifies the transparency for color of the border.

Line Style

Specifies the line style to apply to the border of the pies.

Thickness

Specifies the weight of the border.

End Caps

Specifies the ending style of the border line.

- **butt**
  Ends unclosed sub paths and dash segments with no added decoration.
- **round**
  Ends unclosed sub paths and dash segments with a round decoration that has a radius equal to half of the width of the pen.
- **square**
  Ends unclosed sub paths and dash segments with a square projection that extends beyond the end of the segment to a distance equal to half of the line width.

Line Joint

Specifies the line joint style for the border line.

- **miter**
  Joins path segments by extending their outside edges until they meet.
- **round**
  Joins path segments by rounding off the corner at a radius of half the line width.
- **bevel**
  Joins path segments by connecting the outer corners of their wide outlines with a straight segment.

Sample

Displays a preview sample of your selection.

Path

Specifies the fill pattern of the border line.

- **Outline Path**
  Specifies the fill pattern of the border line to be outline path.
- **Fill Path**
  Specifies the fill pattern of the border line to be whole path.

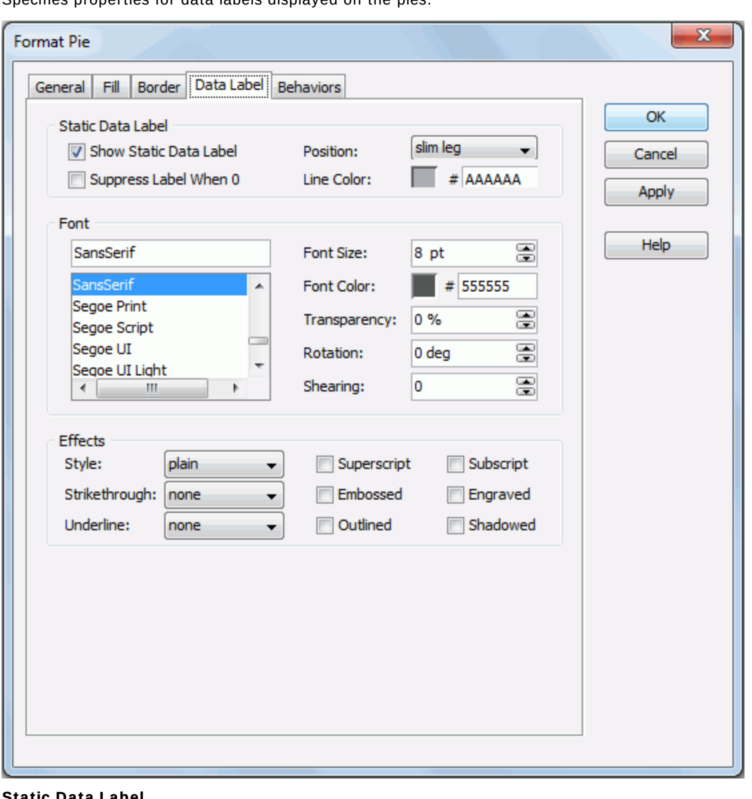
Dash

Specifies the dash size of border line.

- **Auto Adjusted Dash**
  If selected, the dash size will be adjusted automatically.
- **Fixed Dash Size**
  If selected, the dash size will be fixed size.

Data Label tab

Specifies properties for data labels displayed on the pies.



Static Data Label

Specifies properties of the static data labels on the pies.

- **Show Static Data Label**
  Specifies whether or not to show the static data labels on the pies.
- **Position**
  Specifies the position of the static data labels on the pies. Only when it is checked can the following static data label related properties take effect.
  - **auto fit**
    If selected, the static data labels will be displayed automatically.
  - **sticker**
    If selected, the static data labels will be displayed beside the pie sections.
  - **slim leg**
    If selected, the static data labels will be displayed beside the pie sections and pointed by thin lines.
  - **best fit**
    If selected, the static data labels will be displayed at the best fit position automatically.
  - **on slices**
    If selected, the static data labels will be displayed on the pie sections (slices).
- **Suppress Label When 0**
  If true, the static data label whose value is 0 will not be displayed on the chart.
- **Line Color**
  Specifies the color of the thin lines that are used to point to the static data labels.

  To change the color, click the color image button to display the color palette in which you can select a color, or click **More Colors** to customize the color within a wider range in the Pick a Color dialog. You can also input the hexadecimal value of a color directly in the text box after #.

Font

Specifies the font format of text in the data labels.

- **Font list**
  Lists all the available font faces that can be selected to apply to the text.
- **Font Size**
  Specifies the font size of the text.
- **Font Color**
  Specifies the font color of the text.

  To change the color, click the color image button to display the color palette in which you can select a color, or click **More Colors** to customize the color within a wider range in the Pick a Color dialog, or click **More Fill Effects** to customize in the Fill Effects dialog. You can also input the hexadecimal value of a color directly in the text box after #.
- **Transparency**
  Specifies the transparency of the text.
- **Rotation**
  Specifies the rotation angle of the text around its center, in degrees. The default value is 0.
- **Shearing**
  Specifies the gradient of the text.

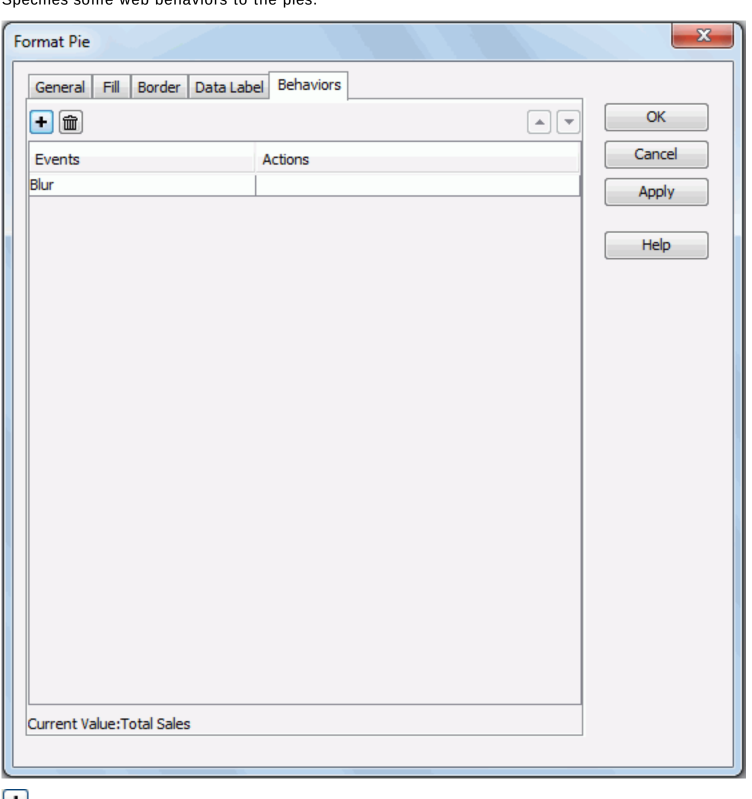
Effects

Specifies the special effects of text in the data labels.

- **Style**
  Specifies the font style of the text. It can be one of the following: plain, bold, italic, and bold italic.
- **Strikethrough**
  Specifies the style of the horizontal line with which the text is struck through. It can be one of the following: none, thin line, bold line, and double lines.
- **Underline**
  Specifies the style of the horizontal line under the text. It can be one of the following: none, single, single lower, bold line, bold lower, double lines, bold double, patterned line, and bold patterned.
- **Superscript**
  Raises the text above the baseline and changes the text to a smaller font size, if a smaller size is available.
- **Embossed**
  Makes the text appear to be raised off the page in relief.
- **Outlined**
  Displays the inner and outer borders of each character.
- **Subscript**
  Lowers the text below the baseline and changes the text to a smaller font size, if a smaller size is available.
- **Engraved**
  Makes the text appear to be imprinted or pressed into the page.
- **Shadowed**
  Adds a shadow beneath and to the right of the text.

Behaviors tab

Specifies some web behaviors to the pies.



Adds a new web behavior line.

Removes the selected web behavior.

Moves the selected web behavior up a step.

Moves the selected web behavior down a step.

Events

Specifies the trigger event.

Actions

Specifies the action you want the event to trigger.

- Opens the Web Action List dialog to bind a web action to the event.

Related topics:

- Formatting the pies/donuts

Format Pie dialog for report

This dialog appears when you double-click a pie of a chart in a page report or web report, or right-click it and select Format Pie from the shortcut menu. It helps you to format pies of a chart, and consists of the following tabs:

- General tab
- Fill tab
- Border tab
- Data Label tab

OK

Applies the changes and closes the dialog.

Cancel

Does not retain any changes and closes the dialog.

Apply

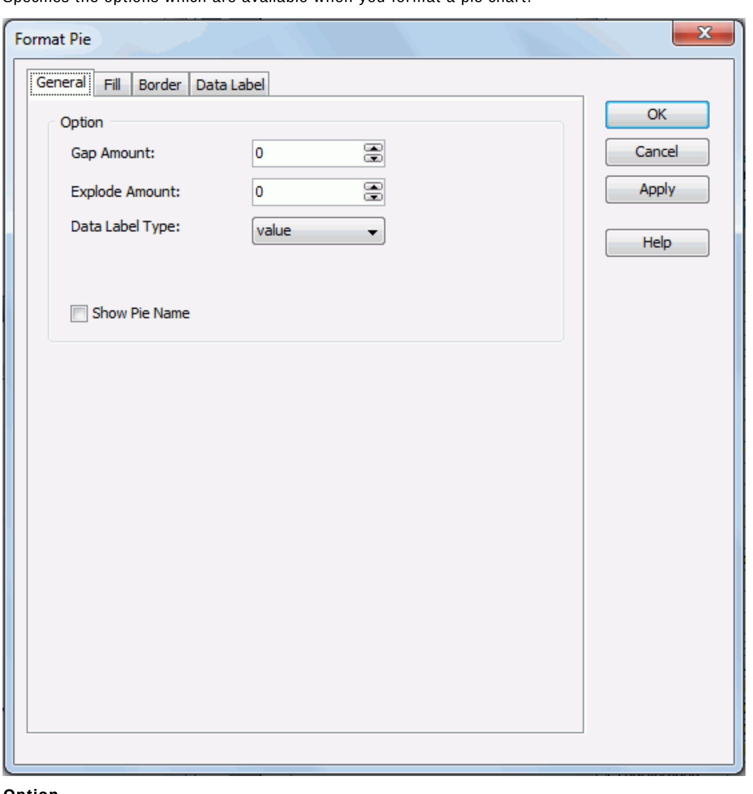
Applies all changes and leaves the dialog open.

Help

Displays the help document about this feature.

General tab

Specifies the options which are available when you format a pie chart.

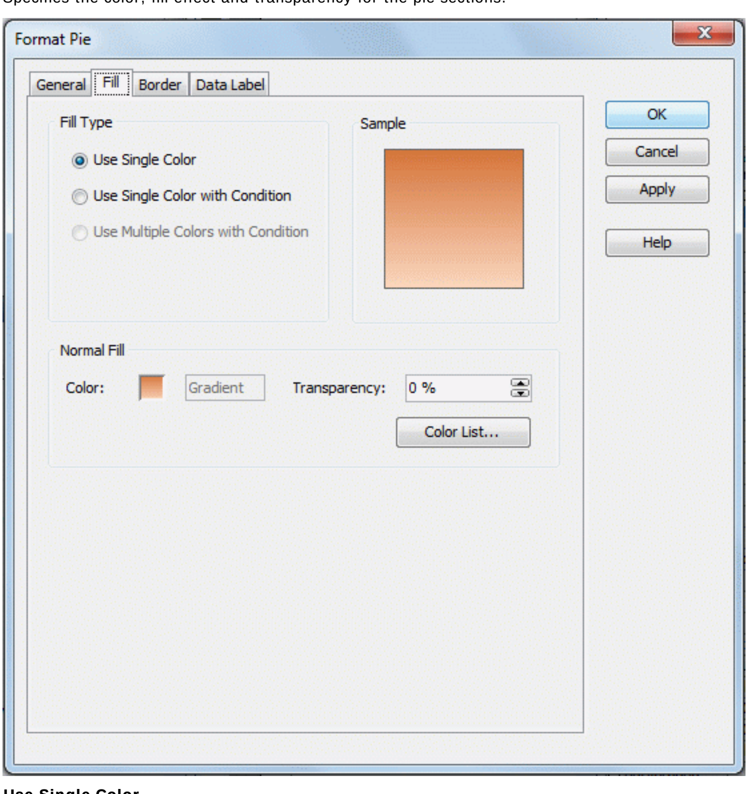


Option

- **Gap Amount**
  Specifies the gap between two pies.
- **Explode Amount**
  Specifies the gap between every section and the central point of a pie.
- **Data Label Type**
  Specifies the display type for data value in the data label. It can be value, category name, percent, value and percent.
- **Show Pie Name**
  Specifies whether to show the name of the pie.

Fill tab

Specifies the color, fill effect and transparency for the pie sections.



Use Single Color

If checked, the pie sections will use the single color pattern.

- **Color**
  Specifies the color schema for the selected pie sections in the same data series.

  To change the color, click the color image button to display the color palette in which you can select a color, or click **More Colors** to customize the color within a wider range in the Pick a Color dialog, or click **More Fill Effects** to customize in the Fill Effects dialog. You can also input the hexadecimal value of a color directly in the text box after # if you have changed the default color.
- **Transparency**
  Specifies the transparency of the color schema.
- **Color List**
  Opens the Color List dialog to modify color pattern for pie sections in the same data series.

Use Single Color with Condition

If checked, each pie section can have its single color pattern based on defined conditions. You can click the Advanced or Normal button to switch between the two editing modes to edit the conditions.

- **Normal**
  - **Select Field**
    Lists all the available fields to which the conditional fill can be applied. Select the field on which you want to define the conditions from the drop-down list.
  - **Values**
    Specifies the value of the condition. Available when you select a category or series field or a field whose return value is not a number from the Select Field drop-down list.
  - **Start Value**
    Specifies the start value of the condition. Available only when you select a value field or a field that returns a number from the Select Field drop-down list.
  - **End Value**
    Specifies the end value of the condition. Available only when you select a value field or a field that returns a number from the Select Field drop-down list.
  - **Color**
    Specifies the color that will be applied to the values which meet the condition.

    To change the color, click the color image button to display the color palette in which you can select a color, or click **More Colors** to customize the color within a wider range in the Pick a Color dialog, or click **More Fill Effects** to customize in the Fill Effects dialog. You can also input the hexadecimal value of a color directly in the text box after #.
  - **Transparency**
    Specifies the transparency for the color of the condition.
  - Adds a new condition line.
  - Removes the selected condition line.
  - **Label**
    If checked, you can modify the condition expression of the selected condition line which will be shown as the legend entry label.
  - **Value**
    Specifies to display data in the condition expression as value. Available only when you select a value field or a field that returns a number from the Select Field drop-down list.
  - **Percent**
    Specifies to display data in the condition expression in percent. Available only when you select a value field or a field that returns a number from the Select Field drop-down list.
- **Advanced**
  - **Condition**
    Displays the condition you have defined in the Edit Condition dialog.
  - **Color**
    Specifies the color that will be applied to the values which meet the condition.

    To change the color, click the color image button to display the color palette in which you can select a color, or click **More Colors** to customize the color within a wider range in the Pick a Color dialog, or click **More Fill Effects** to customize in the Fill Effects dialog. You can also input the hexadecimal value of a color directly in the text box after #.
  - **Transparency**
    Specifies the transparency for the color of the condition.
  - Opens the Edit Conditions dialog to create or edit a condition.
  - **Label**
    If checked, you can modify the condition expression of the selected condition line which will be shown as the legend entry label.
  - **Value**
    Specifies to display data in the condition expression as value. Available only when you select a value field or a field that returns a number from the Select Field drop-down list.
  - **Percent**
    Specifies to display data in the condition expression in percent. Available only when you select a value field or a field that returns a number from the Select Field drop-down list.

Use Multiple Colors with Condition

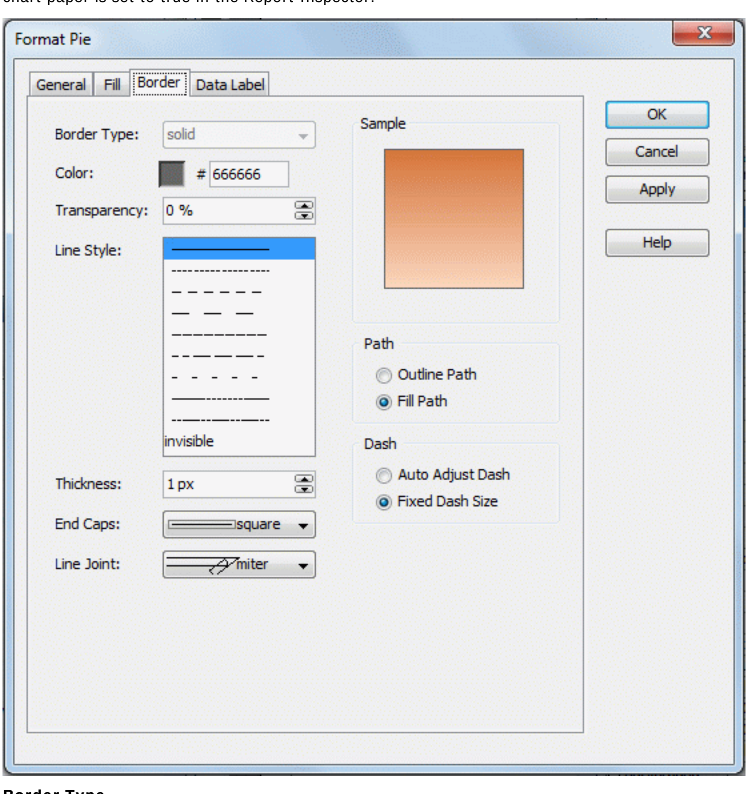
Not supported on pie chart.

Sample

Displays a preview sample of your selection.

Border tab

Specifies the properties for the border of the pies, which take effect only when the Border property on chart paper is set to true in the Report Inspector.



Border Type

Displays the type for border of the pies. The default value is solid, and cannot be changed.

Color

Specifies the color for the border of the pies.

To change the color, click the color image button to display the color palette in which you can select a color, or click **More Colors** to customize the color within a wider range in the Pick a Color dialog. You can also input the hexadecimal value of a color directly in the text box after #.

Transparency

Specifies the transparency for color of the border.

Line Style

Specifies the line style to apply to the border of the pies.

Thickness

Specifies the weight of the border.

End Caps

Specifies the ending style of the border line.

- **butt**
  Ends unclosed sub paths and dash segments with no added decoration.
- **round**
  Ends unclosed sub paths and dash segments with a round decoration that has a radius equal to half of the width of the pen.
- **square**
  Ends unclosed sub paths and dash segments with a square projection that extends beyond the end of the segment to a distance equal to half of the line width.

Line Joint

Specifies the line joint style for the border line.

- **miter**
  Joins path segments by extending their outside edges until they meet.
- **round**
  Joins path segments by rounding off the corner at a radius of half the line width.
- **bevel**
  Joins path segments by connecting the outer corners of their wide outlines with a straight segment.

Sample

Displays a preview sample of your selection.

Path

Specifies the fill pattern of the border line.

- **Outline Path**
  Specifies the fill pattern of the border line to be outline path.
- **Fill Path**
  Specifies the fill pattern of the border line to be whole path.

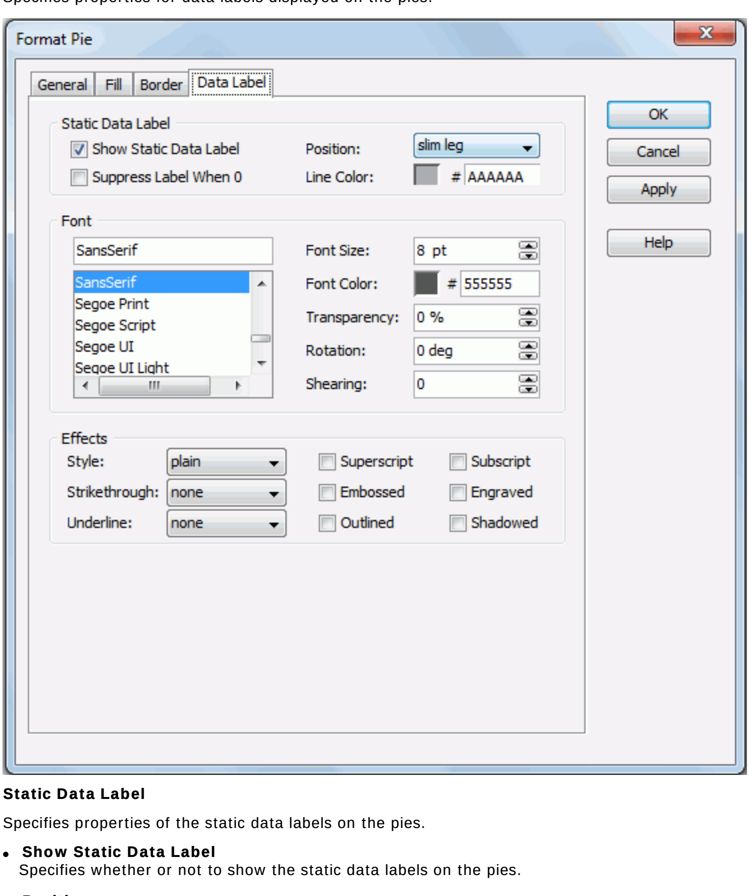
Dash

Specifies the dash size of border line.

- **Auto Adjusted Dash**
  If selected, the dash size will be adjusted automatically.
- **Fixed Dash Size**
  If selected, the dash size will be fixed size.

Data Label tab

Specifies properties for data labels displayed on the pies.



Static Data Label

Specifies properties of the static data labels on the pies.

- **Show Static Data Label**
  Specifies whether or not to show the static data labels on the pies.
- **Position**
  Specifies the position of the static data labels on the pies. Only when it is checked can the following static data label related properties take effect.
  - **auto fit**
    If selected, the static data labels will be displayed automatically.
  - **sticker**
    If selected, the static data labels will be displayed beside the pie sections.
  - **slim leg**
    If selected, the static data labels will be displayed beside the pie sections and pointed by thin lines.
  - **best fit**
    If selected, the static data labels will be displayed at the best fit position automatically.
  - **on slices**
    If selected, the static data labels will be displayed on the pie sections (slices).
- **Suppress Label When 0**
  If true, the static data label whose value is 0 will not be displayed on the chart.
- **Line Color**
  Specifies the color of the thin lines that are used to point to the static data labels.

  To change the color, click the color image button to display the color palette in which you can select a color, or click **More Colors** to customize the color within a wider range in the Pick a Color dialog. You can also input the hexadecimal value of a color directly in the text box after #.

Font

Specifies the font format of text in the data labels.

- **Font list**
  Lists all the available font faces that can be selected to apply to the text.
- **Font Size**
  Specifies the font size of the text.
- **Font Color**
  Specifies the font color of the text.

  To change the color, click the color image button to display the color palette in which you can select a color, or click **More Colors** to customize the color within a wider range in the Pick a Color dialog, or click **More Fill Effects** to customize in the Fill Effects dialog. You can also input the hexadecimal value of a color directly in the text box after #.
- **Transparency**
  Specifies the transparency of the text.
- **Rotation**
  Specifies the rotation angle of the text around its center, in degrees. The default value is 0.
- **Shearing**
  Specifies the gradient of the text.

Effects

Specifies the special effects of text in the data labels.

- **Style**
  Specifies the font style of the text. It can be one of the following: plain, bold, italic, and bold italic.
- **Strikethrough**
  Specifies the style of the horizontal line with which the text is struck through. It can be one of the following: none, thin line, bold line, and double lines.
- **Underline**
  Specifies the style of the horizontal line under the text. It can be one of the following: none, single, single lower, bold line, bold lower, double lines, bold double, patterned line, and bold patterned.
- **Superscript**
  Raises the text above the baseline and changes the text to a smaller font size, if a smaller size is available.
- **Embossed**
  Makes the text appear to be raised off the page in relief.
- **Outlined**
  Displays the inner and outer borders of each character.
- **Subscript**
  Lowers the text below the baseline and changes the text to a smaller font size, if a smaller size is available.
- **Engraved**
  Makes the text appear to be imprinted or pressed into the page.
- **Shadowed**
  Adds a shadow beneath and to the right of the text.

Related topics:

- Formatting the pies/donuts

Format Platform dialog

This dialog appears when you double-click the platform of a chart, or right-click a chart and then select Format Platform from the shortcut menu. The platform of a chart is the container of the other chart elements, such as the paper, legend, and labels. This dialog allows you to format the platform of the chart at any time, and consists of the following tabs:

- Fill tab
- Border tab

OK

Applies the changes and closes the dialog.

Cancel

Does not retain any changes and closes the dialog.

Apply

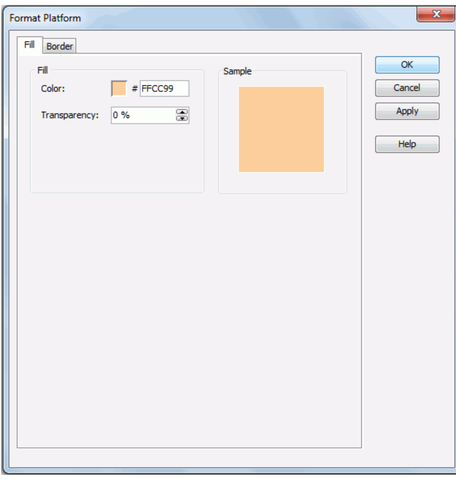
Applies all changes and leaves the dialog open.

Help

Displays the help document about this feature.

Fill tab

Specifies the color schema to fill the platform.



Color

Specifies the color with which to fill the platform.

To change the color, click the color image button to display the color palette in which you can select a color, or click **More Colors** to customize the color within a wider range in the Pick a Color dialog, or click **More Fill Effects** to customize in the Fill Effects dialog. You can also input the hexadecimal value of a color directly in the text box after # if you have changed the default color.

Transparency

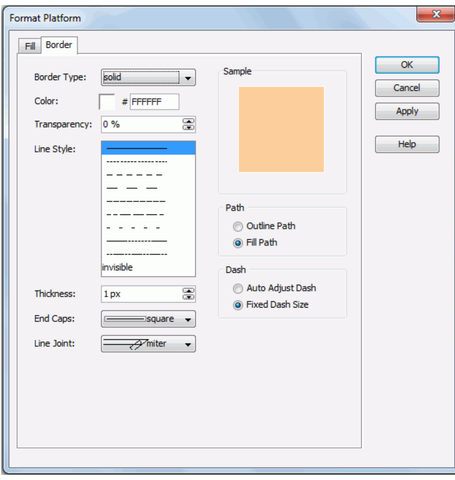
Specifies the transparency of the color.

Sample

Displays the selected color and its transparency effects.

Border tab

Specifies the properties for borders of the chart platform.



Border Type

Specifies the type for border of the platform.

- **none**
  The object has no visible border lines.
- **raised**
  The object has 3D borders that appear as if they are raised off the page.
- **recess**
  The object has 3D borders that appear as if they are pressed into the page.
- **shadow**
  The object has two shadowed borders, beneath and to the right of the object.
- **solid**
  The object has single-line borders.

Color

Specifies the color for border of the platform.

To change the color, click the color image button to display the color palette in which you can select a color, or click **More Colors** to customize the color within a wider range in the Pick a Color dialog. You can also input the hexadecimal value of a color directly in the text box after #.

Transparency

Specifies the transparency for color of the border.

Line Style

Specifies the line style to apply to the border of the platform.

Thickness

Specifies the thickness of the border, in pixels.

End Caps

Specifies the ending style of the border line.

- **butt**
  Ends unclosed sub paths and dash segments with no added decoration.
- **round**
  Ends unclosed sub paths and dash segments with a round decoration that has a radius equal to half of the width of the pen.
- **square**
  Ends unclosed sub paths and dash segments with a square projection that extends beyond the end of the segment to a distance equal to half of the line width.

Line Joint

Specifies the line joint style for the border line.

- **miter**
  Joins path segments by extending their outside edges until they meet.
- **round**
  Joins path segments by rounding off the corner at a radius of half the line width.
- **bevel**
  Joins path segments by connecting the outer corners of their wide outlines with a straight segment.
- **joint round**
  Joins path segments by rounding off the corner at the specified radius.

Path

Specifies the fill pattern of the border line.

- **Outline Path**
  Specifies the fill pattern of the border line to be outline path.
- **Fill Path**
  Specifies the fill pattern of the border line to be whole path.

Dash

Specifies the dash size of the border line.

- **Auto Adjusted Dash**
  If selected, the dash size will be adjusted automatically.
- **Fixed Dash Size**
  If selected, the dash size will be fixed size.

Radius

Specifies the radius for the border joint of the platform border line. This option is available only when the border line joint is set to joint round.

Related topics:

- Formatting the platform

Format Radar dialog

This dialog appears when you double-click radar in a chart, or right-click it and select Format Radar from the shortcut menu. It helps you to format radars in a chart, and consists of the following tabs:

- General tab
- Fill tab
- Border tab
- Data Label tab

OK
Applies the changes and closes the dialog.

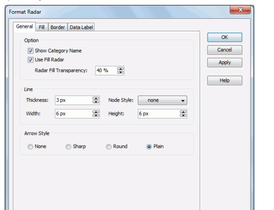
Cancel
Does not retain any changes and closes the dialog.

Apply
Applies all changes and leaves the dialog open.

Help
Displays the help document about this feature.

General tab

Specifies the general format of the radar chart.



Options

- **Show Category Name**
  Specifies whether to show the category names on the radar chart.
- **Use Fill Radar**
  Specifies whether to use colors to fill the radar areas.
  - **Radar Fill Transparency**
    Specifies the transparency for colors of the radar areas.

Line

- **Thickness**
  Specifies the thickness for the radar lines, in pixels.
- **Node Style**
  Specifies the type of node styles on the radar lines.
- **Width**
  Specifies the width for node styles on the radar lines, in pixels.
- **Height**
  Specifies the height for node styles on the radar lines, in pixels.

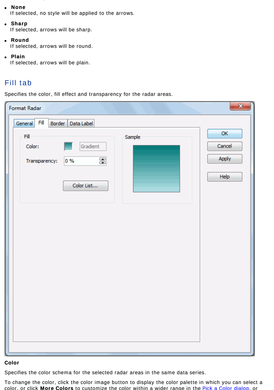
Arrow Style

Specifies the style for arrows on the lines.

- **None**
  If selected, no style will be applied to the arrows.
- **Sharp**
  If selected, arrows will be sharp.
- **Round**
  If selected, arrows will be round.
- **Plain**
  If selected, arrows will be plain.

Fill tab

Specifies the color, fill effect and transparency for the radar areas.



Color

Specifies the color schema for the selected radar areas in the same data series.

To change the color, click the color image button to display the color palette in which you can select a color, or click **More Colors** to customize the color within a wider range in the Pick a Color dialog, or click **More Fill Effects** to customize in the Fill Effects dialog. You can also input the hexadecimal value of a color directly in the text box after #. If you have changed the default color.

Transparency

Specifies the transparency of the color schema.

Color List

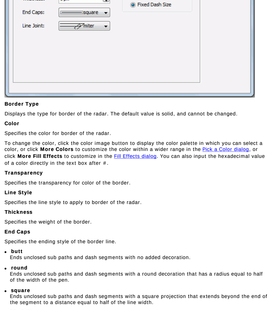
Opens the Color List dialog to modify color pattern for radar areas in the same data series.

Sample

Displays a preview sample of your selection.

Border tab

Specifies properties for border of the radar, which take effect only when the Border property on chart paper is set to true in the Report Inspector.



Border Type

Displays the type for border of the radar. The default value is solid, and cannot be changed.

Color

Specifies the color for border of the radar.

To change the color, click the color image button to display the color palette in which you can select a color, or click **More Colors** to customize the color within a wider range in the Pick a Color dialog, or click **More Fill Effects** to customize in the Fill Effects dialog. You can also input the hexadecimal value of a color directly in the text box after #.

Transparency

Specifies the transparency for color of the border.

Line Style

Specifies the line style to apply to border of the radar.

Thickness

Specifies the weight of the border.

End Caps

Specifies the ending style of the border line.

- **butt**
  Ends unclosed sub paths and dash segments with no added decoration.
- **round**
  Ends unclosed sub paths and dash segments with a round decoration that has a radius equal to half of the width of the pen.
- **square**
  Ends unclosed sub paths and dash segments with a square projection that extends beyond the end of the segment to a distance equal to half of the line width.

Line Joint

Specifies the line joint style for the border line.

- **miter**
  Joins path segments by extending their outside edges until they meet.
- **round**
  Joins path segments by rounding off the corner at a radius of half the line width.
- **bevel**
  Joins path segments by connecting the outer corners of their wide outlines with a straight segment.

Sample

Displays a preview sample of your selection.

Path

Specifies the fill pattern of the border line.

- **Outline Path**
  Specifies the fill pattern of the border line to be outline path.
- **Fill Path**
  Specifies the fill pattern of the border line to be whole path.

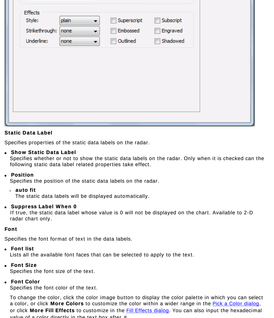
Dash

Specifies the dash size of border line.

- **Auto Adjusted Dash**
  If selected, the dash size will be adjusted automatically.
- **Fixed Dash Size**
  If selected, the dash size will be fixed size.

Data Label tab

Specifies properties for data labels displayed on the radar.



Static Data Label

Specifies properties of the static data labels on the radar.

- **Show Static Data Label**
  Specifies whether or not to show the static data labels on the radar. Only when it is checked can the following static data label related properties take effect.
- **Position**
  Specifies the position of the static data labels on the radar.
  - **auto fit**
    The static data labels will be displayed automatically.
- **Suppress Label When 0**
  If true, the static data label whose value is 0 will not be displayed on the chart. Available to 2-D radar chart only.

Font

Specifies the font format of text in the data labels.

- **Font list**
  Lists all the available font faces that can be selected to apply to the text.
- **Font Size**
  Specifies the font size of the text.
- **Font Color**
  Specifies the font color of the text.

  To change the color, click the color image button to display the color palette in which you can select a color, or click **More Colors** to customize the color within a wider range in the Pick a Color dialog, or click **More Fill Effects** to customize in the Fill Effects dialog. You can also input the hexadecimal value of a color directly in the text box after #.
- **Transparency**
  Specifies the transparency of the text.
- **Rotation**
  Specifies the rotation angle of the text around its center, in degrees. The default value is 0.
- **Shearing**
  Specifies the gradient of the text.

Effects

Specifies the special effects of text in the data labels.

- **Style**
  Specifies the font style of the text. It can be one of the following: plain, bold, italic, and bold italic.
- **Strikethrough**
  Specifies the style of the horizontal line with which the text is struck through. It can be one of the following: none, thin line, bold line, and double lines.
- **Underline**
  Specifies the style of the horizontal line under the text. It can be one of the following: none, single, single lower, bold line, bold lower, double lines, bold double, patterned line, and bold patterned.
- **Superscript**
  Raises the text above the baseline and changes the text to a smaller font size, if a smaller size is available.
- **Embossed**
  Makes the text appear to be raised off the page in relief.
- **Outlined**
  Displays the inner and outer borders of each character.
- **Subscript**
  Lowers the text below the baseline and changes the text to a smaller font size, if a smaller size is available.
- **Engraved**
  Makes the text appear to be imprinted or pressed into the page.
- **Shadowed**
  Adds a shadow beneath and to the right of the text.

Related topics

- Formatting the radar.

Format Rectangle dialog for library component

The dialog appears:

- When there is only one group in the heat map, right-click on the heat map, and then select Format Rectangle from the shortcut menu.
- When there are two or more groups in the heat map, right-click on the heat map, and then select a desired group field from the Format Rectangle submenu.

It helps you to format the rectangles of the specified group, and consists of the following tabs:

- Fill tab
- Separator tab
- Behaviors tab

OK

Applies the changes and closes the dialog.

Cancel

Does not retain any changes and closes the dialog.

Apply

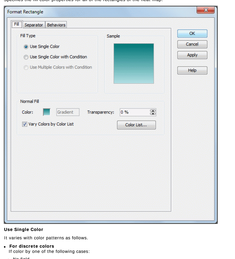
Applies all changes and leaves the dialog open.

Help

Displays the help document about this feature.

Fill tab

Specifies the fill color properties for all of the rectangles of the heat map.



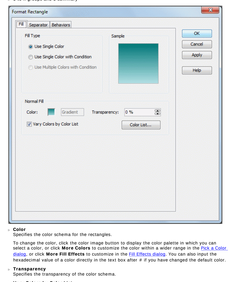
Use Single Color

It varies with color patterns as follows.

- **For discrete colors**

If color by one of the following cases:

  - No field
  - 1 to n groups
  - 1 to n groups and 1 summary



- **Color**

Specifies the color schema for the rectangles.

To change the color, click the color image button to display the color palette in which you can select a color, or click **More Colors** to customize the color within a wider range in the Pick a Color dialog, or click **More Fill Effects** to customize in the Fill Effects dialog. You can also input the hexadecimal value of a color directly in the text box after #. If you have changed the default color.

- **Transparency**

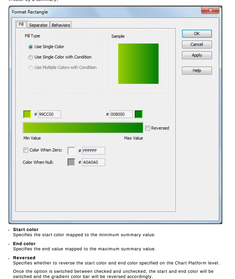
Specifies the transparency of the color schema.

- **Vary Colors by Color List**

Specifies whether or not to make the rectangles' colors vary. If checked, you can specify the color for each rectangle in the Color List dialog brought by clicking the Color List button.

- **For gradient colors**

If color by a summary.



- **Color**

Specifies the color schema for the rectangles.

To change the color, click the color image button to display the color palette in which you can select a color, or click **More Colors** to customize the color within a wider range in the Pick a Color dialog, or click **More Fill Effects** to customize in the Fill Effects dialog. You can also input the hexadecimal value of a color directly in the text box after #. If you have changed the default color.

- **Transparency**

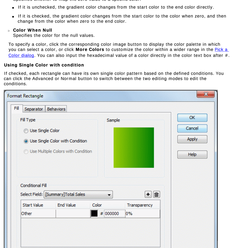
Specifies the transparency of the color schema.

- **Vary Colors by Color List**

Specifies whether or not to make the rectangles' colors vary. If checked, you can specify the color for each rectangle in the Color List dialog brought by clicking the Color List button.

- **For gradient colors**

If color by a summary.



- **Start color**

Specifies the start color mapped to the minimum summary value.

- **End color**

Specifies the end value mapped to the maximum summary value.

- **Reversed**

Specifies whether to reverse the start color and end color specified on the Chart Platform level.

Once the option is switched between checked and unchecked, the start and end color will be switched and the gradient color bar will be reversed accordingly.

If the option is unchecked, the start/end color on the tab is corresponding to the start/end color on the chart platform. Otherwise, the start/end color on the tab is corresponding to end/start color on the chart platform.

- **Color When Zero**

Specifies whether to map the zero value to a specific color.

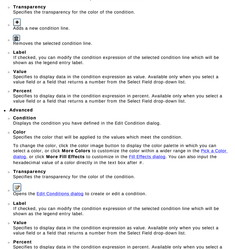
- If it is unchecked, the gradient color changes from the start color to the end color directly.
- If it is checked, the gradient color changes from the start color to the color when zero, and then changes from the color when zero to the end color.

- **Color When Null**

Specifies the color mapped for the null values.

If checked, each rectangle can have its own single color pattern based on the defined conditions. You can click the Advanced or Normal button to switch between the two editing modes to edit the conditions.

Using Single Color with condition



- **Normal**
  - **Select Field**

Lists all the available fields to which the conditional fill can be applied. Select the field on which you want to define the conditions from the drop-down list.

  - **Values**

Specifies the value of the condition. Available when you select a group by field or a field whose return value is not a number from the Select Field drop-down list.

  - **Start Value**

Specifies the start value of the condition. Available only when you select a value field or a field that returns a number from the Select Field drop-down list.

  - **End Value**

Specifies the end value of the condition. Available only when you select a value field or a field that returns a number from the Select Field drop-down list.

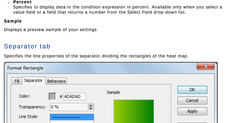
  - **Color**

Specifies the color that will be applied to the values which meet the condition.

To change the color, click the color image button to display the color palette in which you can select a color, or click **More Colors** to customize the color within a wider range in the Pick a Color dialog, or click **More Fill Effects** to customize in the Fill Effects dialog. You can also input the hexadecimal value of a color directly in the text box after #.

  - **Transparency**

Specifies the transparency for the color of the condition.



  - Adds a new condition line.
  - Removes the selected condition line.
  - **Label**

If checked, you can modify the condition expression of the selected condition line which will be shown as the legend entry label.

  - **Value**

Specifies to display data in the condition expression as value. Available only when you select a value field or a field that returns a number from the Select Field drop-down list.

  - **Percent**

Specifies to display data in the condition expression in percent. Available only when you select a value field or a field that returns a number from the Select Field drop-down list.

- **Advanced**
  - **Condition**

Displays the condition you have defined in the Edit Condition dialog.

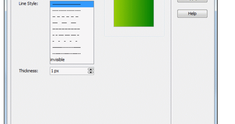
  - **Color**

Specifies the color that will be applied to the values which meet the condition.

To change the color, click the color image button to display the color palette in which you can select a color, or click **More Colors** to customize the color within a wider range in the Pick a Color dialog, or click **More Fill Effects** to customize in the Fill Effects dialog. You can also input the hexadecimal value of a color directly in the text box after #.

  - **Transparency**

Specifies the transparency for the color of the condition.



  - Opens the the Edit Conditions dialog to create or edit a condition.
  - **Label**

If checked, you can modify the condition expression of the selected condition line which will be shown as the legend entry label.

  - **Value**

Specifies to display data in the condition expression as value. Available only when you select a value field or a field that returns a number from the Select Field drop-down list.

  - **Percent**

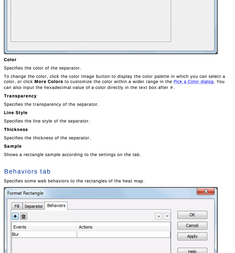
Specifies to display data in the condition expression in percent. Available only when you select a value field or a field that returns a number from the Select Field drop-down list.

Sample

Displays a preview sample of your settings.

Separator tab

Specifies the line properties of the separator dividing the rectangles of the heat map.



Color

Specifies the color of the separator.

To change the color, click the color image button to display the color palette in which you can select a color, or click **More Colors** to customize the color within a wider range in the Pick a Color dialog. You can also input the hexadecimal value of a color directly in the text box after #.

Transparency

Specifies the transparency of the separator.

Line Style

Specifies the line style of the separator.

Thickness

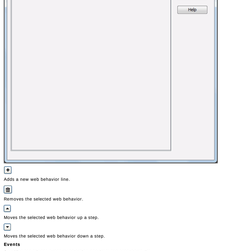
Specifies the thickness of the separator.

Sample

Shows a rectangle sample according to the settings on the tab.

Behaviors tab

Specifies some web behaviors to the rectangles of the heat map.



- Adds a new web behavior line.
- Removes the selected web behavior.
- Moves the selected web behavior up a step.
- Moves the selected web behavior down a step.

Events

Specifies the trigger event by selecting one from the drop-down list.

Actions

Specifies the action you want the event to trigger.

- Opens the Web Action List dialog to bind a web action to the event.

Related topics:

- Formatting the rectangles

Format Rectangle dialog for report

The dialog appears:

- When there is only one group in the heat map, right-click on the heat map, and then select Format Rectangle from the shortcut menu.
- When there are two or more groups in the heat map, right-click on the heat map, and then select a desired group field from the Format Rectangle submenu.

It helps you to format the rectangles of the specified group, and consists of the following tabs:

- Fill tab
- Separator tab

OK
Applies the changes and closes the dialog.

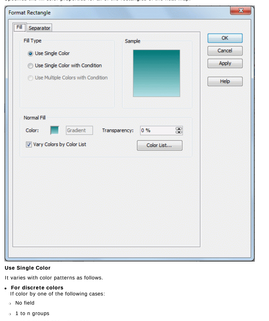
Cancel
Does not retain any changes and closes the dialog.

Apply
Applies all changes and leaves the dialog open.

Help
Displays the help document about this feature.

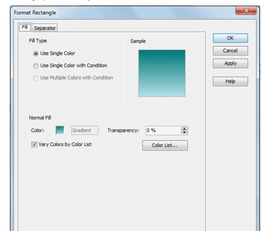
Fill tab

Specifies the fill color properties for all of the rectangles of the heat map.



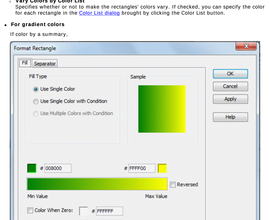
Use Single Color
It varies with color patterns as follows.

- **For discrete colors**
  If color by one of the following cases:
  - No field
  - 1 to n groups
  - 1 to n groups and 1 summary



  - **Color**
    Specifies the color schema for the rectangles.

    To specify a color, click the color image button to display the color palette in which you can select a color, or click **More Colors** to customize the color within a wider range in the Pick a Color dialog, or click **More Fill Effects** to customize in the Fill Effects dialog. You can also input the hexadecimal value of a color directly in the text box after #. If you have changed the default color.
  - **Transparency**
    Specifies the transparency of the color schema.
  - **Vary Colors by Color List**
    Specifies whether or not to make the rectangles' colors vary. If checked, you can specify the color for each rectangle in the Color List dialog brought by clicking the Color List button.
- **For gradient colors**
  If color by a summary:



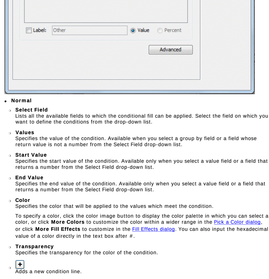
  - **Start Color**
    Specifies the start color mapped to the minimum summary value.
  - **End Color**
    Specifies the end value mapped to the maximum summary value.
  - **Reversed**
    Specifies whether to reverse the start color and end color specified on the Chart Platform level.

    Once the option is switched between checked and unchecked, the start and end color will be swatched and the gradient color bar will be reversed accordingly.

    If the option is unchecked, the start/end color on the tab is corresponding to the start/end color on the chart platform. Otherwise, the start/end color on the tab is corresponding to end/start color on the chart platform.
  - **Color When Zero**
    Specifies whether to map the zero value to a specific color.
    - If it is unchecked, the gradient color changes from the start color to the end color directly.
    - If it is checked, the gradient color changes from the start color to the color when zero, and then change from the color when zero to the end color.
  - **Color When Null**
    Specifies the color for the null values.

To specify a color, click the corresponding color image button to display the color palette in which you can select a color, or click **More Colors** to customize the color within a wider range in the Pick a Color dialog. You can also input the hexadecimal value of a color directly in the color text box after #.

Using Single Color with condition
If checked, each rectangle can have its own single color pattern based on the defined conditions. You can click the Advanced or Normal button to switch between the two editing modes to edit the conditions.



- **Normal**
  - **Select Field**
    Lists all the available fields to which the conditional fill can be applied. Select the field on which you want to define the conditions from the drop-down list.
  - **Values**
    Specifies the value of the condition. Available when you select a group by field or a field whose return value is not a number from the Select Field drop-down list.
  - **Start Value**
    Specifies the start value of the condition. Available only when you select a value field or a field that returns a number from the Select Field drop-down list.
  - **End Value**
    Specifies the end value of the condition. Available only when you select a value field or a field that returns a number from the Select Field drop-down list.
  - **Color**
    Specifies the color that will be applied to the values which meet the condition.

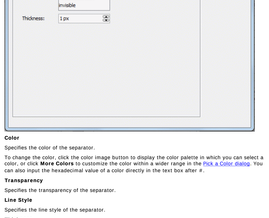
    To specify a color, click the color image button to display the color palette in which you can select a color, or click **More Colors** to customize the color within a wider range in the Pick a Color dialog, or click **More Fill Effects** to customize in the Fill Effects dialog. You can also input the hexadecimal value of a color directly in the text box after #.
  - **Transparency**
    Specifies the transparency for the color of the condition.
  -
    Adds a new condition line.
  -
    Removes the selected condition line.
  - **Label**
    If checked, you can modify the condition expression of the selected condition line which will be shown as the legend entry label.
  - **Value**
    Specifies to display data in the condition expression as value. Available only when you select a value field or a field that returns a number from the Select Field drop-down list.
  - **Percent**
    Specifies to display data in the condition expression in percent. Available only when you select a value field or a field that returns a number from the Select Field drop-down list.
- **Advanced**
  - **Condition**
    Displays the condition you have defined in the Edit Condition dialog.
  - **Color**
    Specifies the color that will be applied to the values which meet the condition.

    To specify a color, click the color image button to display the color palette in which you can select a color, or click **More Colors** to customize the color within a wider range in the Pick a Color dialog, or click **More Fill Effects** to customize in the Fill Effects dialog. You can also input the hexadecimal value of a color directly in the text box after #.
  - **Transparency**
    Specifies the transparency for the color of the condition.
  -
    Opens the Edit Conditions dialog to create or edit a condition.
  - **Label**
    If checked, you can modify the condition expression of the selected condition line which will be shown as the legend entry label.
  - **Value**
    Specifies to display data in the condition expression as value. Available only when you select a value field or a field that returns a number from the Select Field drop-down list.
  - **Percent**
    Specifies to display data in the condition expression in percent. Available only when you select a value field or a field that returns a number from the Select Field drop-down list.

Sample
Displays a preview sample of your settings.

Separator tab

Specifies the line properties of the separator dividing the rectangles of the heat map.



Color
Specifies the color of the separator.

To change the color, click the color image button to display the color palette in which you can select a color, or click **More Colors** to customize the color within a wider range in the Pick a Color dialog. You can also input the hexadecimal value of a color directly in the text box after #.

Transparency
Specifies the transparency of the separator.

Line Style
Specifies the line style of the separator.

Thickness
Specifies the thickness of the separator.

Sample
Shows a rectangle sample according to the settings on the tab.

Related topics:

- Formatting the rectangles

Format Rectangle Title dialog

The dialog appears:

- When there is only one group in the heat map, right-click on the heat map, and then select Format Rectangle Title from the shortcut menu.
- When there are two or more groups in the heat map, right-click on the heat map, and then select a desired group field from the Format Rectangle Title submenu.

It helps you to format the rectangle title of the specified group, and consists of the following tabs:

- General tab
- Font tab
- Border tab

OK

Applies the changes and closes the dialog.

Cancel

Does not retain any changes and closes the dialog.

Apply

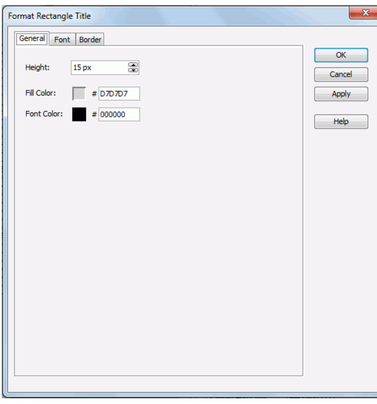
Applies all changes and leaves the dialog open.

Help

Displays the help document about this feature.

General tab

Specifies the height and color properties of the rectangle title.



Height

Specifies the height of the title.

Fill Color

Specifies the background color of the title.

To change the color, click the color image button to display the color palette in which you can select a color, or click **More Colors** to customize the color within a wider range in the Pick a Color dialog. You can also input color string directly in the text box after # in the format #RRGGBB.

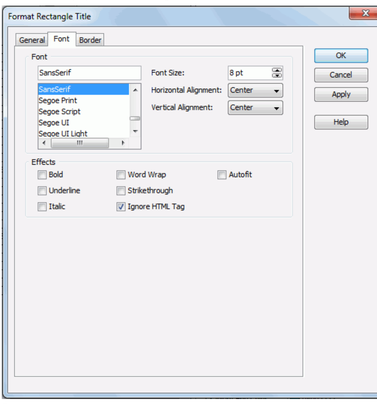
Font Color

Specifies the foreground color of the title.

To change the color, click the color image button to display the color palette in which you can select a color, or click **More Colors** to customize the color within a wider range in the Pick a Color dialog. You can also input color string directly in the text box after # in the format #RRGGBB.

Font tab

Specifies the font properties of the rectangle title.



Font

Specifies the font face of the text in the title.

Font Size

Specifies the font size of the text in the title.

Horizontal Alignment

Specifies the horizontal alignment mode of the text in the title.

Vertical Alignment

Specifies the vertical alignment mode of the text in the title.

Bold

Specifies whether to make the text bold or not.

Underline

Specifies whether the text will be underlined or not.

Italic

Specifies whether to make the text italic or not.

Word Wrap

Specifies whether or not to wrap the text to the title width.

Strikethrough

Specifies whether or not to attach a strikeout line to the text.

Ignore HTML Tag

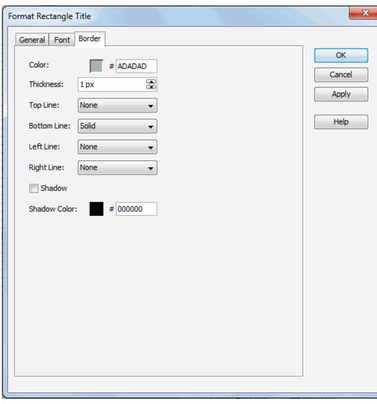
If this option is unchecked, JReport will parse HTML tag elements in the field value while the report is to be saved as an HTML file; or the field value will appear in the HTML file the same as that in JReport (HTML tag elements in the field value, if any, will not be parsed).

Autofit

Specifies whether or not to automatically adjust the width of the title.

Border tab

Specifies the border properties of the rectangle title.



Color

Specifies the border color.

To change the color, click the color image button to display the color palette in which you can select a color, or click **More Colors** to customize the color within a wider range in the Pick a Color dialog. You can also input color string directly in the text box after # in the format #RRGGBB.

Thickness

Specifies the border width.

Top Line

Specifies the style of the top border line.

Bottom Line

Specifies the style of the bottom border line.

Left Line

Specifies the style of the left border line.

Right Line

Specifies the style of the right border line.

Shadow

Specifies whether the borders will have a shadow effect or not.

Shadow Color

Specifies the color of the border shadow.

To change the color, click the color image button to display the color palette in which you can select a color, or click **More Colors** to customize the color within a wider range in the Pick a Color dialog. You can also input color string directly in the text box after # in the format #RRGGBB.

Related topics:

- Formatting the rectangle titles

Format Scatter dialog

This dialog appears when you double-click a scatter marker of a scatter chart, or right-click it and select Format Scatter from the shortcut menu. It helps you to format scatter markers in a scatter chart, and consists of the following tabs:

- General tab
- Fill tab
- Data Label tab

OK

Applies the changes and closes the dialog.

Cancel

Does not retain any changes and closes the dialog.

Apply

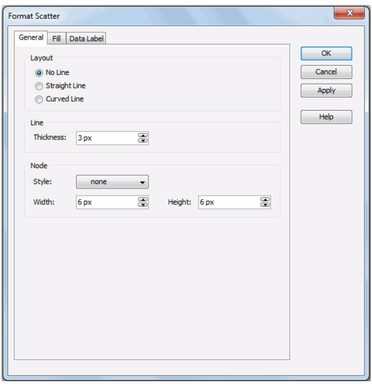
Applies all changes and leaves the dialog open.

Help

Displays the help document about this feature.

General tab

Specifies the general format of the scatter chart.



Layout

Specifies the layout for scatter markers in the chart.

- **No line**
  If selected, no lines will be used to joint scatter markers in the chart.
- **Straight line**
  If selected, straight lines will be used to joint scatter markers in the chart.
- **Curved line**
  If selected, curved lines will be used to joint scatter markers in the chart.

Line

Specifies properties for lines of the scatter chart.

- **Thickness**
  Specifies the thickness for lines in the scatter chart, in pixels.

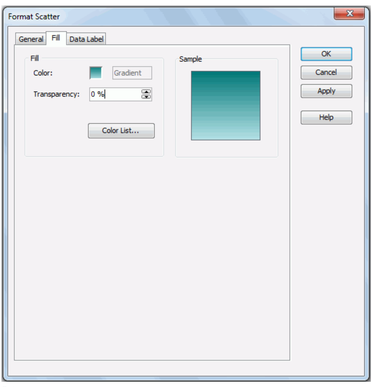
Node

Specifies properties for line nodes in the scatter chart.

- **Style**
  Specifies the node style for line nodes.
- **Width**
  Specifies the width for line nodes, in pixels.
- **Height**
  Specifies the height for line nodes, in pixels.

Fill tab

Specifies the color, fill effect and transparency of the scatter markers.



Color

Specifies the color schema for the selected scatter markers in the same data series.

To change the color, click the color image button to display the color palette in which you can select a color, or click **More Colors** to customize the color within a wider range in the Pick a Color dialog, or click **More Fill Effects** to customize in the Fill Effects dialog. You can also input the hexadecimal value of a color directly in the text box after # if you have changed the default color.

Transparency

Specifies the transparency of the color schema.

Color List

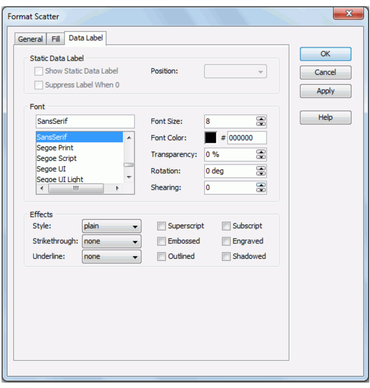
Opens the Color List dialog to modify color pattern for scatter markers in the same data series.

Sample

Displays a preview sample of your selection.

Data Label tab

Specifies properties for data labels displayed on the scatter markers.



Static Data Label

Specifies properties of the static data labels. Not supported on scatter chart.

Font

Specifies the font format of text in the data labels.

- **Font list**
  Lists all the available font faces that can be selected to apply to the text.
- **Font Size**
  Specifies the font size of the text.
- **Font Color**
  Specifies the font color of the text.

  To change the color, click the color image button to display the color palette in which you can select a color, or click **More Colors** to customize the color within a wider range in the Pick a Color dialog, or click **More Fill Effects** to customize in the Fill Effects dialog. You can also input the hexadecimal value of a color directly in the text box after #.
- **Transparency**
  Specifies the transparency of the text.
- **Rotation**
  Specifies the rotation angle of the text around its center, in degrees. The default value is 0.
- **Shearing**
  Specifies the gradient of the text.

Effects

Specifies the special effects of text in the data labels.

- **Style**
  Specifies the font style of the text. It can be one of the following: plain, bold, italic, and bold italic.
- **Strikethrough**
  Specifies the style of the horizontal line with which the text is struck through. It can be one of the following: none, thin line, bold line, and double lines.
- **Underline**
  Specifies the style of the horizontal line under the text. It can be one of the following: none, single, single lower, bold line, bold lower, double lines, bold double, patterned line, and bold patterned.
- **Superscript**
  Raises the text above the baseline and changes the text to a smaller font size, if a smaller size is available.
- **Embossed**
  Makes the text appear to be raised off the page in relief.
- **Outlined**
  Displays the inner and outer borders of each character.
- **Subscript**
  Lowers the text below the baseline and changes the text to a smaller font size, if a smaller size is available.
- **Engraved**
  Makes the text appear to be imprinted or pressed into the page.
- **Shadowed**
  Adds a shadow beneath and to the right of the text.

Related topics:

- Formatting the scatter

Format Series (Z) Axis dialog

This dialog appears when you double-click the series (Z) axis of a chart, or right-click a chart and then select Format Axis > Format Series (Z) Axis from the shortcut menu. It helps you to format the series (Z) axis of a chart, and consists of the following tabs:

- Axis tab
- Tick Mark tab
- Font tab
- Orientation tab
- Format tab

OK

Applies the changes and closes the dialog.

Cancel

Does not retain any changes and closes the dialog.

Apply

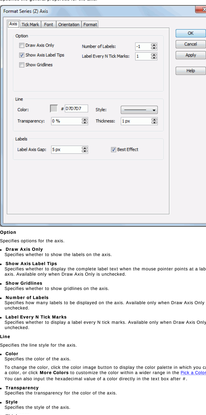
Applies all changes and leaves the dialog open.

Help

Displays the help document about this feature.

Axis tab

Specifies the general properties for the axis.



Option

Specifies options for the axis.

- **Draw Axis Only**
  Specifies whether to show the labels on the axis.
- **Show Axis Label Tips**
  Specifies whether to display the complete label text when the mouse pointer points at a label on the axis. Available only when Draw Axis Only is unchecked.
- **Show Gridlines**
  Specifies whether to show gridlines on the axis.
- **Number of Labels**
  Specifies how many labels to be displayed on the axis. Available only when Draw Axis Only is unchecked.
- **Label Every N Tick Marks**
  Specifies whether to display a label every N tick marks. Available only when Draw Axis Only is unchecked.

Line

Specifies the line style for the axis.

- **Color**
  Specifies the color of the axis.
  To change the color, click the color image button to display the color palette in which you can select a color, or click **More Colors** to customize the color within a wider range in the Pick a Color dialog. You can also input the hexadecimal value of a color directly in the text box after #.
- **Transparency**
  Specifies the transparency for the color of the axis.
- **Style**
  Specifies the style of the axis.
- **Thickness**
  Specifies the thickness of the axis.

Labels

Specifies properties for the labels on the axis.

- **Label Axis Gap**
  Specifies the distance between the axis and the labels.
- **Best Effect**
  Specifies whether to adjust the labels automatically to make them placed best.

Tick Mark tab

Specifies properties of the tick marks on the axis.

Tick Mark

Specifies the line properties of the tick marks.

- **Correlate with Axis**
  If checked, the line properties of the tick marks will correlate with that of the axis automatically. Only when it is unchecked can the following four properties take effect.
- **Color**
  Specifies the color of the tick marks.
  To change the color, click the color image button to display the color palette in which you can select a color, or click **More Colors** to customize the color within a wider range in the Pick a Color dialog. You can also input the hexadecimal value of a color directly in the text box after #.
- **Transparency**
  Specifies the transparency for the color of the tick marks.
- **Style**
  Specifies the type of the tick marks.
- **Thickness**
  Specifies the thickness of the tick marks.

Major Tick Mark

Specifies properties of the major tick marks on the axis.

- **None**
  If selected, major tick marks will not be shown on the axis and it will be meaningless to specify all the other major tick mark related properties.
- **Inside**
  If selected, major tick marks will be inside the chart.
- **Outside**
  If selected, major tick marks will be outside the chart.
- **Cross**
  If selected, major tick marks will be across the axis.
- **Major Tick Length**
  Specifies the length of the major tick marks on the axis.

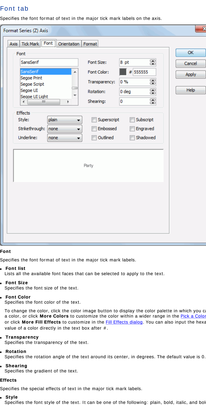
Minor Tick Mark

Specifies properties of the minor tick marks on the axis.

- **None**
  If selected, minor tick marks will not be shown on the axis and it will be meaningless to specify all the other minor tick mark related properties.
- **Inside**
  If selected, minor tick marks will be inside the chart.
- **Outside**
  If selected, minor tick marks will be outside the chart.
- **Cross**
  If selected, minor tick marks will be across the chart.
- **Minor Tick Length**
  Specifies the length of the minor tick marks on the axis.

Font tab

Specifies the font format of text in the major tick mark labels on the axis.



Font

Specifies the font format of text in the major tick mark labels.

- **Font list**
  Lists all the available font faces that can be selected to apply to the text.
- **Font Size**
  Specifies the font size of the text.
- **Font Color**
  Specifies the font color of the text.
  To change the color, click the color image button to display the color palette in which you can select a color, or click **More Colors** to customize the color within a wider range in the Pick a Color dialog or click **More Effects** to customize in the Fill Effects dialog. You can also input the hexadecimal value of a color directly in the text box after #.
- **Transparency**
  Specifies the transparency of the text.
- **Rotation**
  Specifies the rotation angle of the text around its center, in degrees. The default value is 0.
- **Shearing**
  Specifies the gradient of the text.

Effects

Specifies the special effects of text in the major tick mark labels.

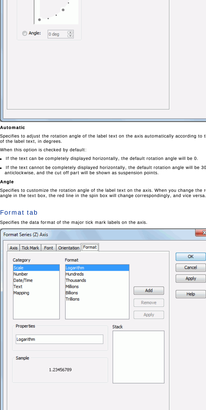
- **Style**
  Specifies the font style of the text. It can be one of the following: plain, bold, italic, and bold italic.
- **Strikethrough**
  Specifies the style of the horizontal line with which the text is struck through. It can be one of the following: none, thin line, bold line, double line, and double lines.
- **Underline**
  Specifies the style of the horizontal line under the text. It can be one of the following: none, single, single lower, bold line, bold lower, double lines, bold double, patterned line, and bold patterned.
- **Superscript**
  Lowers the text above the baseline and changes the text to a smaller font size, if a smaller size is available.
- **Embossed**
  Makes the text appear to be raised off the page in relief.
- **Outlined**
  Displays the inner and outer borders of each character.
- **Subscript**
  Lowers the text below the baseline and changes the text to a smaller font size, if a smaller size is available.
- **Engraved**
  Makes the text appear to be imprinted or pressed into the page.
- **Shadowed**
  Adds a shadow beneath and to the right of the text.

Sample

Displays the specified font and any text effects.

Orientation tab

Specifies the rotation angle of the major tick mark labels on the axis.



Automatic

Specifies to adjust the rotation angle of the label text on the axis automatically according to the length of the label text, in degrees.

When this option is checked by default:

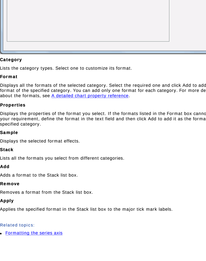
- If the text can be completely displayed horizontally, the default rotation angle will be 0.
- If the text cannot be completely displayed horizontally, the default rotation angle will be 30 anticlockwise, and the cut off part will be shown as suspension points.

Angle

Specifies to customize the rotation angle of the label text on the axis. When you change the rotation angle in the text box, the red line in the spin box will change correspondingly, and vice versa.

Format tab

Specifies the data format of the major tick mark labels on the axis.



Category

Lists the category types. Select one to customize its format.

Format

Displays all the formats of the selected category. Select the required one and click Add to add it as the format of the specified category. You can add only one format for each category. For more details about the formats, see A detailed chart property reference.

Properties

Displays the properties of the format you select. If the formats listed in the Format box cannot meet your requirement, define the format in the text field and then click Add to add it as the format of the specified category.

Sample

Displays the selected format effects.

Stack

Lists all the formats you select from different categories.

Add

Adds a format to the Stack list box.

Remove

Removes a format from the Stack list box.

Apply

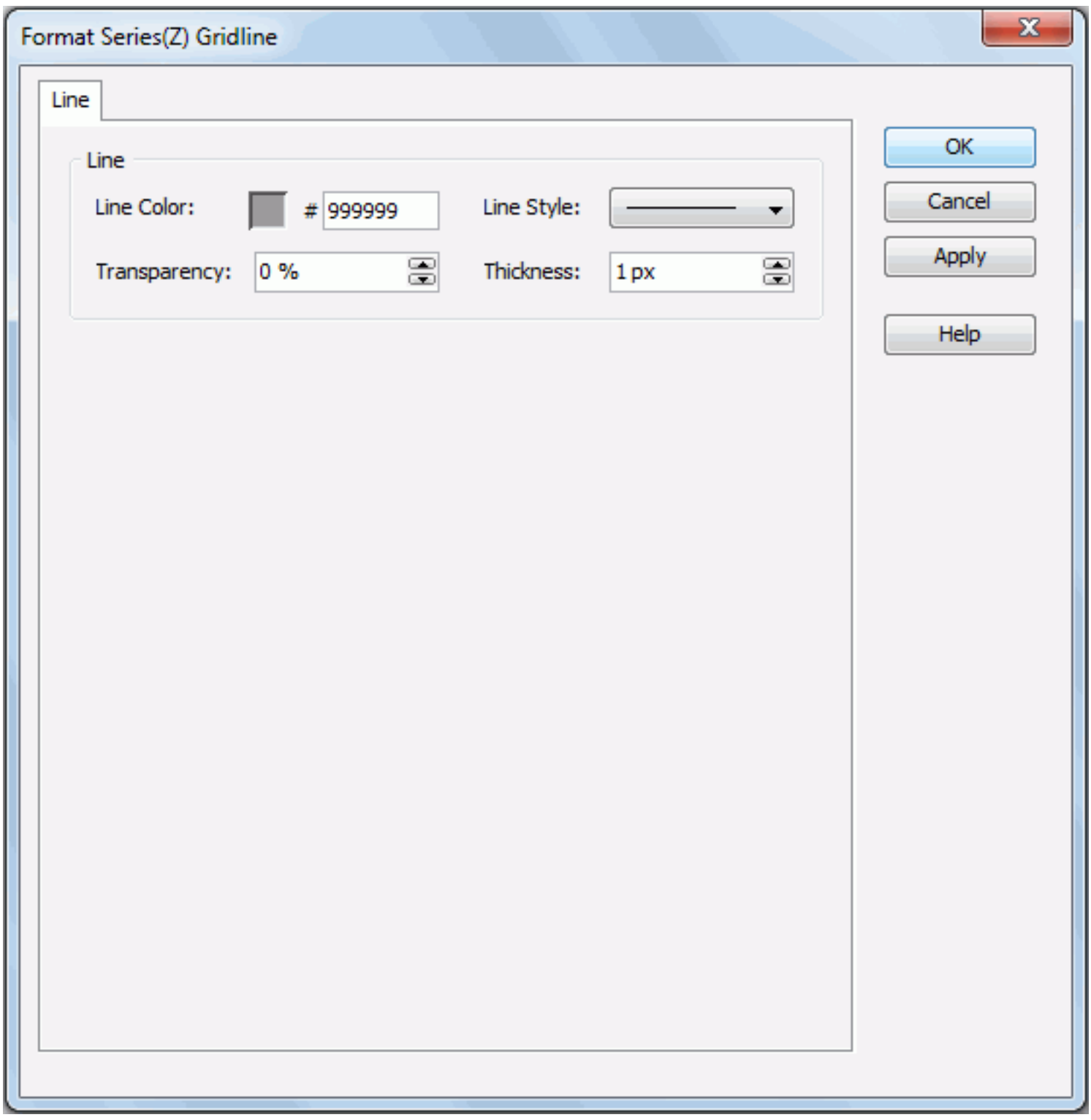
Applies the specified format in the Stack list box to the major tick mark labels.

Related topics:

- Formatting the series axis

Format Series (Z) Gridline dialog

This dialog appears when you double-click a series (Z) gridline of a chart, or right-click a chart and then select Format Gridlines > Series (Z) Gridline from the shortcut menu. It helps you to format the gridlines shown on the series (Z) axis of a chart.



The following are details about options in the dialog:

Line Color

Specifies the color schema of the gridlines.

To change the color, click the color image button to display the color palette in which you can select a color, or click **More Colors** to customize the color within a wider range in the Pick a Color dialog. You can also input the hexadecimal value of a color directly in the text box after #.

Line Style

Specifies the line style of the gridlines.

Transparency

Specifies the transparency of the color that is applied to the gridlines.

Thickness

Specifies the thickness of the gridlines.

OK

Applies the changes and closes the dialog.

Cancel

Does not retain any changes and closes the dialog.

Apply

Applies all changes and leaves the dialog open.

Help

Displays the help document about this feature.

Related topics:

- Formatting the gridlines

Format Stock dialog

This dialog appears when you double-click a stock of a chart, or right-click it and select Format Stock from the shortcut menu. It helps you to format stocks of a chart, and consists of the following tabs:

- General tab
- Border tab
- Data Label tab

OK

Applies the changes and closes the dialog.

Cancel

Does not retain any changes and closes the dialog.

Apply

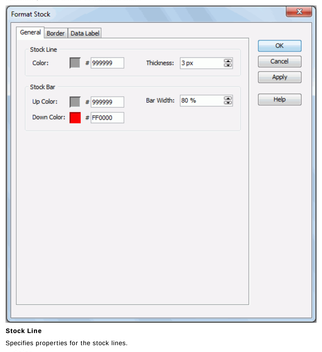
Applies all changes and leaves the dialog open.

Help

Displays the help document about this feature.

General tab

Specifies general format of the stock chart.



Stock Line

Specifies properties for the stock lines.

- **Color**
  Specifies the color schema for the stock lines.

  To change the color, click the color image button to display the color palette in which you can select a color, or click **More Colors** to customize the color within a wider range in the Pick a Color dialog. You can also input the hexadecimal value of a color directly in the text box after #.
- **Thickness**
  Specifies the thickness of the stock lines, in pixels.

Stock Bar

Specifies properties for stock bars of an Open-High-Low-Close stock chart, which requires four series of values in this order.

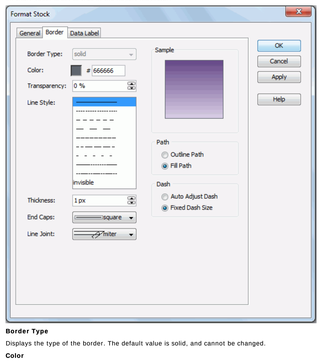
- **Up color**
  Specifies the color schema for stock bars that stand for the stocks that the Open value is lower than the Close value.

  To change the color, click the color image button to display the color palette in which you can select a color, or click **More Colors** to customize the color within a wider range in the Pick a Color dialog. You can also input the hexadecimal value of a color directly in the text box after #.
- **Down color**
  Specifies the color schema for stock bars that stand for the stocks that the Open value is higher than the Close value.

  To change the color, click the color image button to display the color palette in which you can select a color, or click **More Colors** to customize the color within a wider range in the Pick a Color dialog. You can also input the hexadecimal value of a color directly in the text box after #.
- **Bar Width**
  Specifies the width of the stock bars.

Border tab

Specifies the properties for border of the stock bars, which take effect only when the Border property on chart paper is set to true in the Report Inspector.



Border Type

Displays the type of the border. The default value is solid, and cannot be changed.

Color

Specifies the color of the border.

To change the color, click the color image button to display the color palette in which you can select a color, or click **More Colors** to customize the color within a wider range in the Pick a Color dialog. You can also input the hexadecimal value of a color directly in the text box after #.

Transparency

Specifies the transparency for color of the border.

Line Style

Specifies the line style to apply to the border.

Thickness

Specifies the weight of the border, in pixels.

End Caps

Specifies the ending style of the border line.

- **butt**
  Ends unclosed sub paths and dash segments with no added decoration.
- **round**
  Ends unclosed sub paths and dash segments with a round decoration that has a radius equal to half of the width of the pen.
- **square**
  Ends unclosed sub paths and dash segments with a square projection that extends beyond the end of the segment to a distance equal to half of the line width.

Line Joint

Specifies the line joint style for the border line.

- **miter**
  Joins path segments by extending their outside edges until they meet.
- **round**
  Joins path segments by rounding off the corner at a radius of half the line width.
- **bevel**
  Joins path segments by connecting the outer corners of their wide outlines with a straight segment.

Sample

Displays a preview sample of your selection.

Path

Specifies the fill pattern of the border line.

- **Outline Path**
  Specifies the fill pattern of the border line to be outline path.
- **Fill Path**
  Specifies the fill pattern of the border line to be whole path.

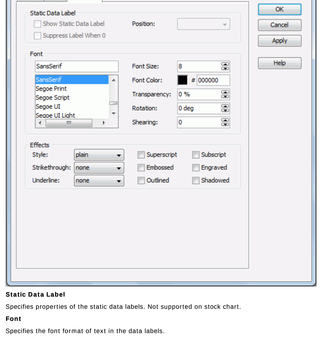
Dash

Specifies the dash size of border line.

- **Auto Adjusted Dash**
  If selected, the dash size will be adjusted automatically.
- **Fixed Dash Size**
  If selected, the dash size will be fixed size.

Data Label tab

Specifies properties for data labels displayed on the stock chart.



Static Data Label

Specifies properties of the static data labels. Not supported on stock chart.

Font

Specifies the font format of text in the data labels.

- **Font list**
  Lists all the available font faces that can be selected to apply to the text.
- **Font Size**
  Specifies the font size of the text.
- **Font Color**
  Specifies the font color of the text.

  To change the color, click the color image button to display the color palette in which you can select a color, or click **More Colors** to customize the color within a wider range in the Pick a Color dialog, or click **More Fill Effects** to customize in the Fill Effects dialog. You can also input the hexadecimal value of a color directly in the text box after #.
- **Transparency**
  Specifies the transparency of the text.
- **Rotation**
  Specifies the rotation angle of the text around its center, in degrees. The default value is 0.
- **Shearing**
  Specifies the gradient of the text.

Effects

Specifies the special effects of text in the data labels.

- **Style**
  Specifies the font style of the text. It can be one of the following: plain, bold, italic, and bold italic.
- **Strikethrough**
  Specifies the style of the horizontal line with which the text is struck through. It can be one of the following: none, thin line, bold line, and double lines.
- **Underline**
  Specifies the style of the horizontal line under the text. It can be one of the following: none, single, single lower, bold line, bold lower, double lines, bold double, patterned line, and bold patterned.
- **Superscript**
  Raises the text above the baseline and changes the text to a smaller font size, if a smaller size is available.
- **Embossed**
  Makes the text appear to be raised off the page in relief.
- **Outlined**
  Displays the inner and outer borders of each character.
- **Subscript**
  Lowers the text below the baseline and changes the text to a smaller font size, if a smaller size is available.
- **Engraved**
  Makes the text appear to be imprinted or pressed into the page.
- **Shadowed**
  Adds a shadow beneath and to the right of the text.

Related topics:

- Formatting the stock

Format Surface dialog

This dialog appears when you double-click a surface of a chart, or right-click it and then select Format Surface from the shortcut menu. It helps you to format surfaces in a chart, and consists of the following tabs:

- Fill tab
- Border tab
- Data Label tab

OK

Applies the changes and closes the dialog.

Cancel

Does not retain any changes and closes the dialog.

Apply

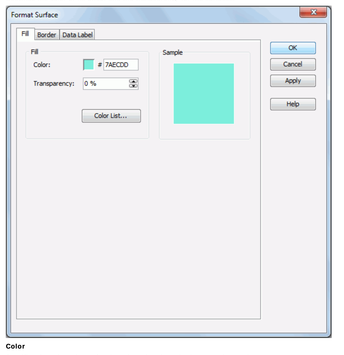
Applies all changes and leaves the dialog open.

Help

Displays the help document about this feature.

Fill tab

Specifies the color or fill effect and the transparency for the surfaces.



Color

Specifies the color schema for the selected surfaces in the same data series.

To change the color, click the color image button to display the color palette in which you can select a color, or click **More Colors** to customize the color within a wider range in the Pick a Color dialog, or click **More Fill Effects** to customize in the Fill Effects dialog. You can also input the hexadecimal value of a color directly in the text box after # if you have changed the default color.

Transparency

Specifies the transparency of the color schema.

Color List

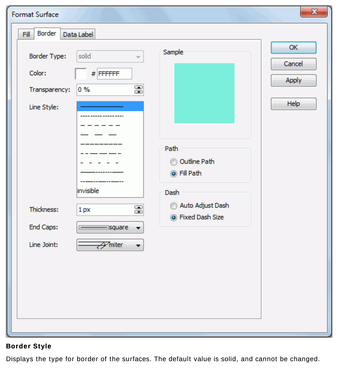
Opens the Color List dialog to modify color pattern for surfaces in the same data series.

Sample

Displays a preview sample of your selection.

Border tab

Specifies properties for border of the surfaces, which take effect only when the Border property on chart paper is set to true in the Report Inspector.



Border Style

Displays the type for border of the surfaces. The default value is solid, and cannot be changed.

Color

Specifies the color for border of the surfaces.

To change the color, click the color image button to display the color palette in which you can select a color, or click **More Colors** to customize the color within a wider range in the Pick a Color dialog. You can also input the hexadecimal value of a color directly in the text box after #.

Transparency

Specifies the transparency for color of the border.

Line Style

Specifies the line style to apply to border of the surfaces.

Thickness

Specifies the thickness of the border, in pixels.

End Caps

Specify the ending style of the border line.

- **butt**
  Ends unclosed sub paths and dash segments with no added decoration.
- **round**
  Ends unclosed sub paths and dash segments with a round decoration that has a radius equal to half of the width of the pen.
- **square**
  Ends unclosed sub paths and dash segments with a square projection that extends beyond the end of the segment to a distance equal to half of the line width.

Line Joint

Specifies the line joint style for the border line.

- **miter**
  Joins path segments by extending their outside edges until they meet.
- **round**
  Joins path segments by rounding off the corner at a radius of half the line width.
- **bevel**
  Joins path segments by connecting the outer corners of their wide outlines with a straight segment.

Sample

Displays a preview sample of your selection.

Path

Specifies the fill pattern of the border line.

- **Outline Path**
  Specify the fill pattern of the border line to be outline path.
- **Fill Path**
  Specify the fill pattern of the border line to be whole path.

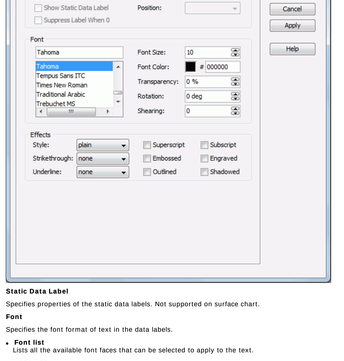
Dash

Specifies the dash size of the border line.

- **Auto Adjusted Dash**
  If selected, the dash size will be adjusted automatically.
- **Fixed Dash Size**
  If selected, the dash size will be fixed size.

Data Label tab

Specifies properties for data labels displayed on the surfaces.



Static Data Label

Specifies properties of the static data labels. Not supported on surface chart.

Font

Specifies the font format of text in the data labels.

- **Font list**
  Lists all the available font faces that can be selected to apply to the text.
- **Font Size**
  Specifies the font size of the text.
- **Font Color**
  Specifies the font color of the text.

  To change the color, click the color image button to display the color palette in which you can select a color, or click **More Colors** to customize the color within a wider range in the Pick a Color dialog, or click **More Fill Effects** to customize in the Fill Effects dialog. You can also input the hexadecimal value of a color directly in the text box after #.
- **Transparency**
  Specifies the transparency of the text.
- **Rotation**
  Specifies the rotation angle of the text around its center, in degrees. The default value is 0.
- **Shearing**
  Specifies the gradient of the text.

Effects

Specifies the special effects of text in the data labels.

- **Style**
  Specifies the font style of the text. It can be one of the following: plain, bold, italic, and bold italic.
- **Strikethrough**
  Specifies the style of the horizontal line with which the text is struck through. It can be one of the following: none, thin line, bold line, and double lines.
- **Underline**
  Specifies the style of the horizontal line under the text. It can be one of the following: none, single, single lower, bold line, bold lower, double lines, bold double, patterned line, and bold patterned.
- **Superscript**
  Raises the text above the baseline and changes the text to a smaller font size, if a smaller size is available.
- **Embossed**
  Makes the text appear to be raised off the page in relief.
- **Outlined**
  Displays the inner and outer borders of each character.
- **Subscript**
  Lowers the text below the baseline and changes the text to a smaller font size, if a smaller size is available.
- **Engraved**
  Makes the text appear to be imprinted or pressed into the page.
- **Shadowed**
  Adds a shadow beneath and to the right of the text.

Related topics:

- Formatting the surface

Format Text dialog

This dialog appears when you right-click some selected strings and then select Font or Paragraph (if the string is in a table cell, Format Text) from the shortcut menu. You can set the properties of the text in a paragraph and font of the text in the following two tabs:

- Paragraph tab
- Font tab

OK

Accepts all changes and closes the dialog.

Cancel

Does not retain any changes and closes the dialog.

Apply

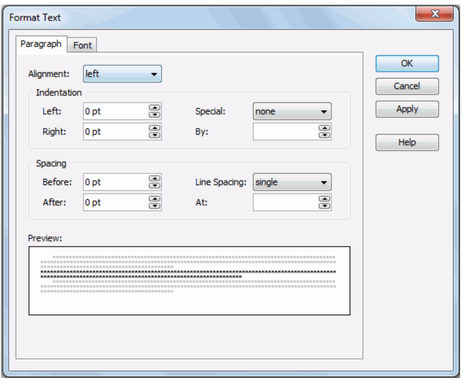
Applies all changes in this dialog.

Help

Displays the help document about this feature.

Paragraph tab

Specifies properties for the text in the paragraph.



Alignment

Specifies how paragraphs fit between the margins.

- **Left**
  Specifies to make the paragraph left-aligned.
- **Centered**
  Specifies to make the paragraph centered.
- **Right**
  Specifies to make the paragraph right-aligned.
- **Justified**
  Specifies to adjust the horizontal spacing so that text is aligned evenly along both the left and right margins. Justifying text creates a smooth edge on both sides.

Indentation

Specifies the distance of the paragraph from either the left or right margins.

- **Left**
  Specifies the distance of the paragraph from the left margin.
- **Right**
  Specifies the distance of the paragraph from the right margin.
- **Special**
  Specifies to make the paragraph specially indented.
  - **None**
    If selected, no special indent rule will be applied in this paragraph.
  - **First line**
    If selected, use the First line indent in the paragraph with the length you specified in the By text box.
  - **Hanging**
    If selected, use the Hanging indent in the paragraph with the length you specified in the By text box.

Spacing

Specifies the spaces between lines in the paragraph.

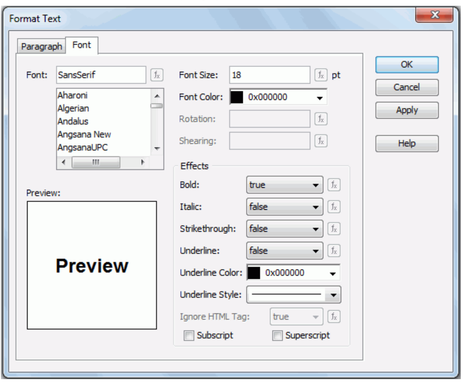
- **Before**
  Specifies the distance with N lines before the specified paragraph.
- **After**
  Specifies the distance with N lines after the specified paragraph.
- **Line Spacing**
  Specifies the amount of vertical space between lines of text in a paragraph.
  - **Single**
    Accommodates the largest font in that line, plus a small amount of extra space. The amount of extra space varies depending on the font used.
  - **1.5 lines**
    One-and-one-half times that of single line spacing.
  - **Double**
    Twice that of single line spacing.
  - **At least**
    Minimum line spacing that is needed to fit the largest font or graphic on the line.
  - **Exactly**
    Fixed line spacing that JReport Designer does not adjust.
  - **Multiple**
    Line spacing that is increased or decreased by a percentage that you specify. For example, setting line spacing to 1.2 will increase the space by 20 percent.
- **At**
  Effective only when you have selected At least, Exactly or Multiple**.** it specifies the distances between lines.

Preview

Displays a preview sample of your selection.

Font tab

Specifies the font format of text in the paragraph.



Font list

Lists all the available font faces for the text.

Font Size

Specifies the font size of the text.

Font Color

Specifies the font color of the text.

Rotation

Specifies the rotation angle of the text around its center, in degrees. Not supported on text in the paragraph.

Shearing

Specifies the gradient of text. Not supported on text in the paragraph.

Preview

Displays a preview sample of your selection.

Effects

Specifies special effect for the text.

- **Bold**
  Specifies whether the text to be bold or not.
- **Italic**
  Specifies whether the text to be italic or not.
- **Strikethrough**
  Specifies whether or not to draw a line through the text.
- **Underline**
  Specifies whether to use the horizontal line under the text or not.
- **Underline Color**
  Specifies the color for the horizontal line under the text.
- **Underline Style**
  Specifies the style of the horizontal line under the text.
- **Ignore HTML Tag**
  If true, the text will appear in the HTML file the same as that in JReport Designer (HTML tag elements in the field value, if any, will not be parsed); or HTML tag elements of the text will be parsed while the report is to be saved as an HTML file. Not supported on text in the paragraph.
- **Subscript**
  Specifies whether to make the specified text as subscript.
- **Superscript**
  Specifies whether to make the specified text as superscript.

Format Value (Y) Axis dialog

This dialog appears when you double-click the value (Y) axis of a chart, or right-click a chart and then select Format Axes > Format Value (Y) Axis from the shortcut menu. It helps you to format the value (Y) axis of a chart, and consists of the following tabs:

- Axis tab
- Tick Mark tab
- Font tab
- Orientation tab
- Format tab

OK

Applies the changes and closes the dialog.

Cancel

Does not retain any changes and closes the dialog.

Apply

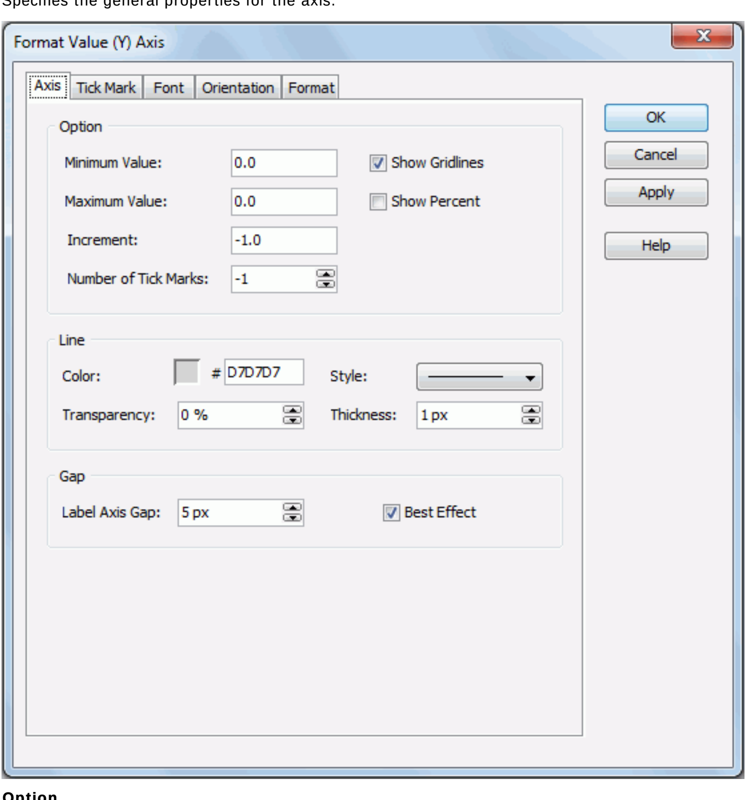
Applies all changes and leaves the dialog open.

Help

Displays the help document about this feature.

Axis tab

Specifies the general properties for the axis.



Option

Specifies the options for the axis.

- **Minimum Value**
  Specifies the minimum value that is to be displayed on the axis.
- **Maximum Value**
  Specifies the maximum value that is to be displayed on the axis.
- **Increment**
  Specifies the difference between two adjacent values on the axis.
- **Number of Tick Marks**
  Specifies how many tick marks to be displayed on the axis.
- **Show Gridlines**
  Specifies whether to show gridlines on the axis.
- **Show Percent**
  Specifies whether to show the value labels on the axis in percent. Only applies to radar, bullet, bar/bench, line, and area chart that are not 100% stacked type.

Line

Specifies the line style for the axis.

- **Color**
  Specifies the color of the axis.

  To change the color, click the color image button to display the color palette in which you can select a color, or click **More Colors** to customize the color within a wider range in the Pick a Color dialog. You can also input the hexadecimal value of a color directly in the text box after #.
- **Style**
  Specifies the style of the axis.
- **Transparency**
  Specifies the transparency for the color of the axis.
- **Thickness**
  Specifies the thickness of the axis, in pixels.

Gap

Specifies the gap properties for the labels on the axis.

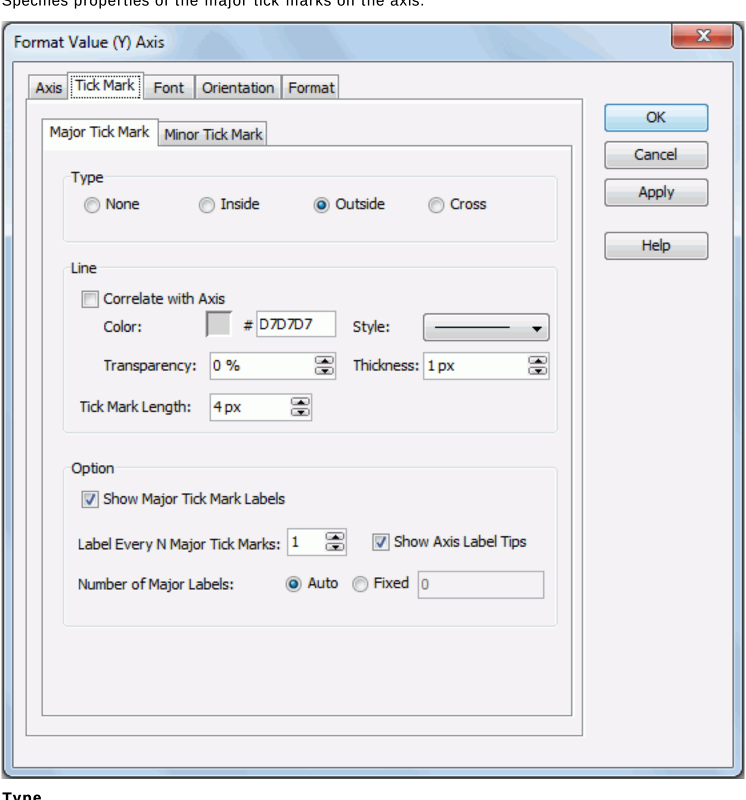
- **Label Axis Gap**
  Specifies the distance between the label and the axis, in pixels.
- **Best Effect**
  Specifies whether to adjust the labels on the axis automatically to make them placed best.

Tick Mark tab

The tab consists of two sub tabs: major tick mark tab, and minor tick mark tab.

Major Tick Mark tab

Specifies properties of the major tick marks on the axis.



Type

Specifies the type of the major tick marks on the axis.

- **None**
  If selected, major tick marks will not be shown on the axis and it will be meaningless to specify all the other major tick mark related properties.
- **Inside**
  If selected, major tick marks will be inside the chart.
- **Outside**
  If selected, major tick marks will be outside the chart.
- **Cross**
  If selected, major tick marks will be across the axis.

Line

Specifies the line properties of the major tick marks on the axis.

- **Correlate with Axis**
  If checked, the line properties of the major tick marks will correlate with that of the axis automatically.
  - **Color**
    Specifies the color of the major tick marks.

    To change the color, click the color image button to display the color palette in which you can select a color, or click **More Colors** to customize the color within a wider range in the Pick a Color dialog. You can also input the hexadecimal value of a color directly in the text box after #.
  - **Transparency**
    Specifies the transparency for the color of the major tick marks.
  - **Style**
    Specifies the type of the major tick marks.
  - **Thickness**
    Specifies the thickness of the major tick marks, in pixels.
- **Tick Mark Length**
  Specifies the length of the major tick marks, in pixels.

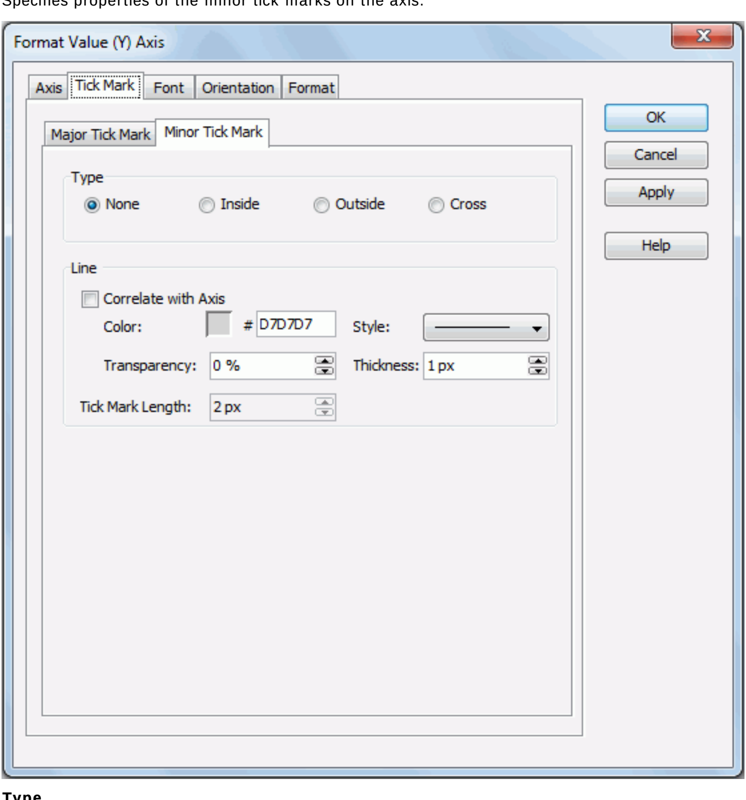
Option

Specifies the other properties of the major tick mark labels on the axis.

- **Show Major Tick Mark Labels**
  Specifies whether to display the labels of the major tick marks on the axis. If checked, the following properties will be enabled.
- **Label Every N Major Tick Marks**
  Specifies the frequency at which the major tick marks will be labeled.
- **Show Axis Label Tips**
  Specifies whether to display the complete label text when the mouse pointer points at a label on the axis.
- **Number of Major Labels**
  Specifies how many major tick mark labels to be displayed on the axis.
  - **Auto**
    If checked, all major tick mark labels will be shown.
  - **Fixed**
    If checked, you can specify the number of the major tick mark labels to be displayed on the axis.

Minor Tick Mark tab

Specifies properties of the minor tick marks on the axis.



Type

Specifies the type of the minor tick marks on the axis.

- **None**
  If selected, minor tick marks will not be shown on the axis and it will be meaningless to specify all the other minor tick mark related properties.
- **Inside**
  If selected, minor tick marks will be inside the chart.
- **Outside**
  If selected, minor tick marks will be outside the chart.
- **Cross**
  If selected, minor tick marks will be across the axis.

Line

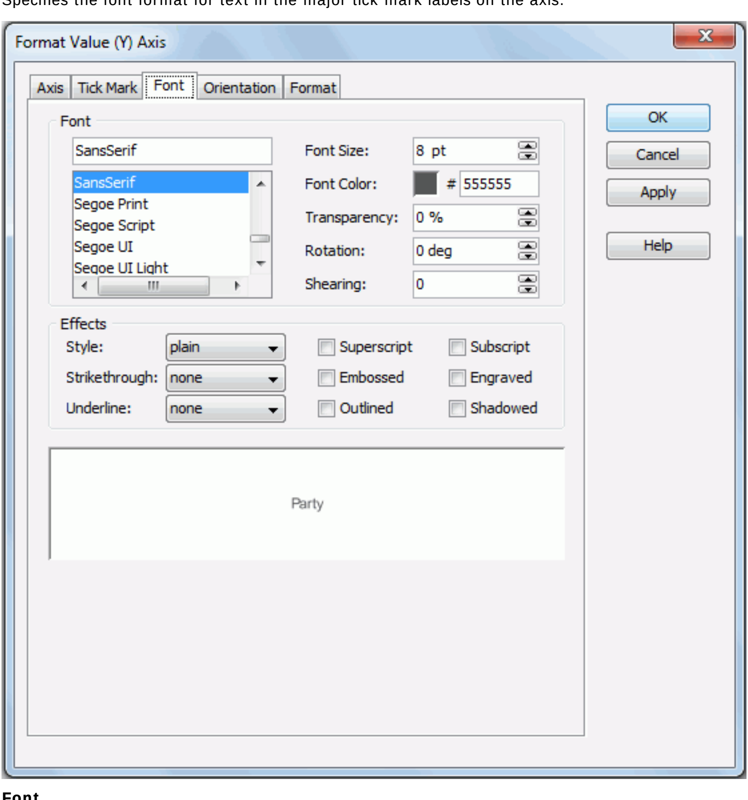
Specifies the line properties of the minor tick marks on the axis.

- **Correlate with Axis**
  If checked, the line properties of the minor tick marks will correlate with that of the axis automatically.
  - **Color**
    Specifies the color of the minor tick marks.

    To change the color, click the color image button to display the color palette in which you can select a color, or click **More Colors** to customize the color within a wider range in the Pick a Color dialog. You can also input the hexadecimal value of a color directly in the text box after #.
  - **Style**
    Specifies the type of the minor tick marks.
  - **Transparency**
    Specifies the transparency for the color of the minor tick marks.
  - **Thickness**
    Specifies the thickness of the minor tick marks, in pixels.
- **Tick Mark Length**
  Specifies the length of the minor tick marks, in pixels.

Font tab

Specifies the font format for text in the major tick mark labels on the axis.



Font

Specifies the font format of text in the major tick mark labels.

- **Font list**
  Lists all the available font faces that can be selected to apply to the text.
- **Font Size**
  Specifies the font size of the text.
- **Font Color**
  Specifies the font color of the text.

  To change the color, click the color image button to display the color palette in which you can select a color, or click **More Colors** to customize the color within a wider range in the Pick a Color dialog, or click **More Fill Effects** to customize in the Fill Effects dialog. You can also input the hexadecimal value of a color directly in the text box after #.
- **Transparency**
  Specifies the transparency of the text.
- **Rotation**
  Specifies the rotation angle of the text around its center, in degrees. The default value is 0.
- **Shearing**
  Specifies the gradient of the text.

Effects

Specifies the special effects of text in the major tick mark labels.

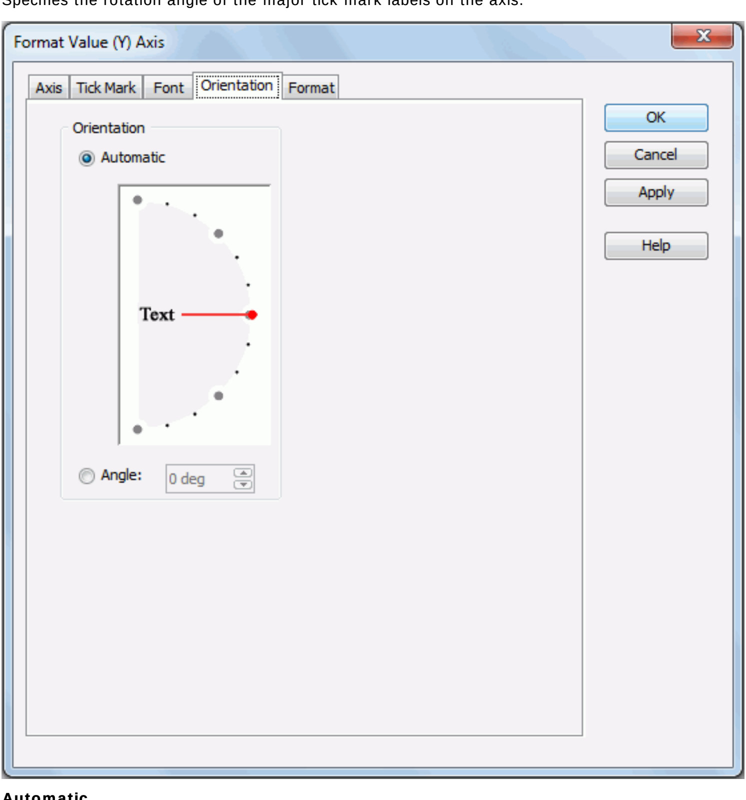
- **Style**
  Specifies the font style of the text. It can be one of the following: plain, bold, italic, and bold italic.
- **Strikethrough**
  Specifies the style of the horizontal line with which the text is struck through. It can be one of the following: none, thin line, bold line, and double lines.
- **Underline**
  Specifies the style of the horizontal line under the text. It can be one of the following: none, single, single lower, bold line, bold lower, double lines, bold double, patterned line, and bold patterned.
- **Superscript**
  Raises the text above the baseline and changes the text to a smaller font size, if a smaller size is available.
- **Embossed**
  Makes the text appear to be raised off the page in relief.
- **Outlined**
  Displays the inner and outer borders of each character.
- **Subscript**
  Lowers the text below the baseline and changes the text to a smaller font size, if a smaller size is available.
- **Engraved**
  Makes the text appear to be imprinted or pressed into the page.
- **Shadowed**
  Adds a shadow beneath and to the right of the text.

Sample

Displays the specified font and any text effects.

Orientation tab

Specifies the rotation angle of the major tick mark labels on the axis.



Automatic

Specifies whether to adjust the rotation angle of the label text on the axis automatically according to the length of the label text, in degrees.

When this option is checked by default:

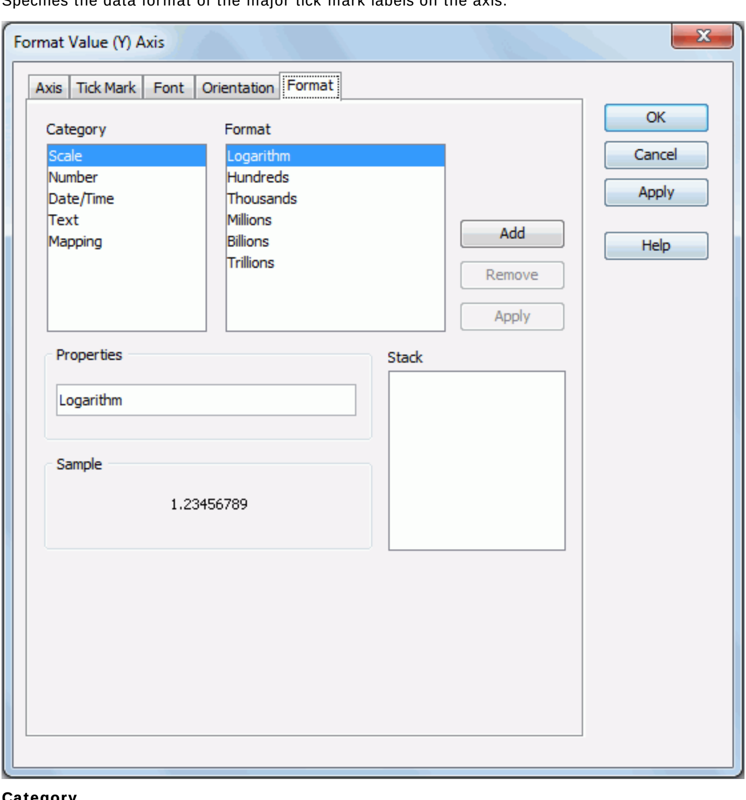
- If the text can be completely displayed horizontally, the default rotation angle will be 0.
- If the text cannot be completely displayed horizontally, the default rotation angle will be 30 anticlockwise, and the cut off part will be shown as suspension points.

Angle

Specifies to customize the rotation angle of the label text on the axis. When you change the rotation angle in the text box, the red line in the spin box will change correspondingly, and vice versa.

Format tab

Specifies the data format of the major tick mark labels on the axis.



Category

Lists the category types. Select one to customize its format.

Format

Displays all the formats of the selected category. Select the required one and click Add to add it as the format of the specified category. You can add only one format for each category. For more details about the formats, see A detailed chart property reference.

Properties

Displays the properties of the format you select. If the formats listed in the Format box cannot meet your requirement, define the format in the text field and then click Add to add it as the format of the specified category.

Sample

Displays the selected format effects.

Stack

Lists all the formats you select from different categories.

Add

Adds a format to the Stack list box.

Remove

Removes a format from the Stack list box.

Apply

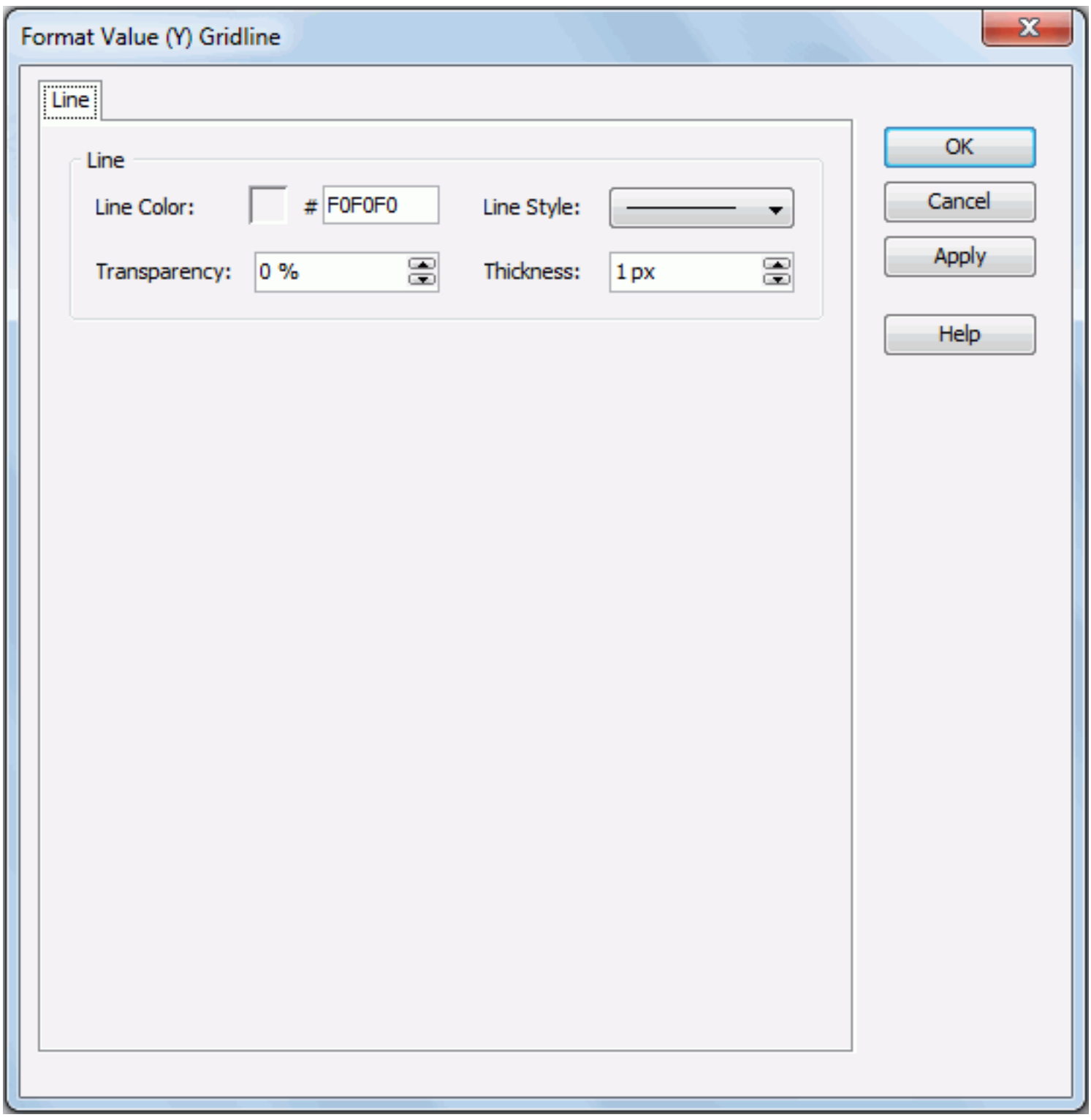
Applies the specified format in the Stack list box to the major tick mark labels.

Related topics:

- Formatting the value axis

Format Value (Y) Gridline dialog

This dialog appears when you double-click a value (Y) gridline of a chart, or right-click a chart and then select Format Gridlines > Value (Y) Gridline from the shortcut menu. It helps you to format the gridlines shown on the value (Y) axis of a chart.



The following are details about options in the dialog:

Line Color

Specifies the color schema of the gridlines.

To change the color, click the color image button to display the color palette in which you can select a color, or click **More Colors** to customize the color within a wider range in the Pick a Color dialog. You can also input the hexadecimal value of a color directly in the text box after #.

Line Style

Specifies the line style of the gridlines.

Transparency

Specifies the transparency of the color that is applied to the gridlines.

Thickness

Specifies the thickness of the gridlines, in pixels.

OK

Applies the changes and closes the dialog.

Cancel

Does not retain any changes and closes the dialog.

Apply

Applies all changes and leaves the dialog open.

Help

Displays the help document about this feature.

Related topics:

- Formatting the gridlines

Format Value (Y2) Axis dialog

This dialog appears when you double-click the value (Y2) axis of a combo chart, or right-click a combo chart and then select Format Axes > Format Value (Y2) Axis from the shortcut menu. It helps you to format the value (Y2) axis of a combo chart, and consists of the following tabs:

- Axis tab
- Tick Mark tab
- Font tab
- Orientation tab
- Format tab

OK

Applies the changes and closes the dialog.

Cancel

Does not retain any changes and closes the dialog.

Apply

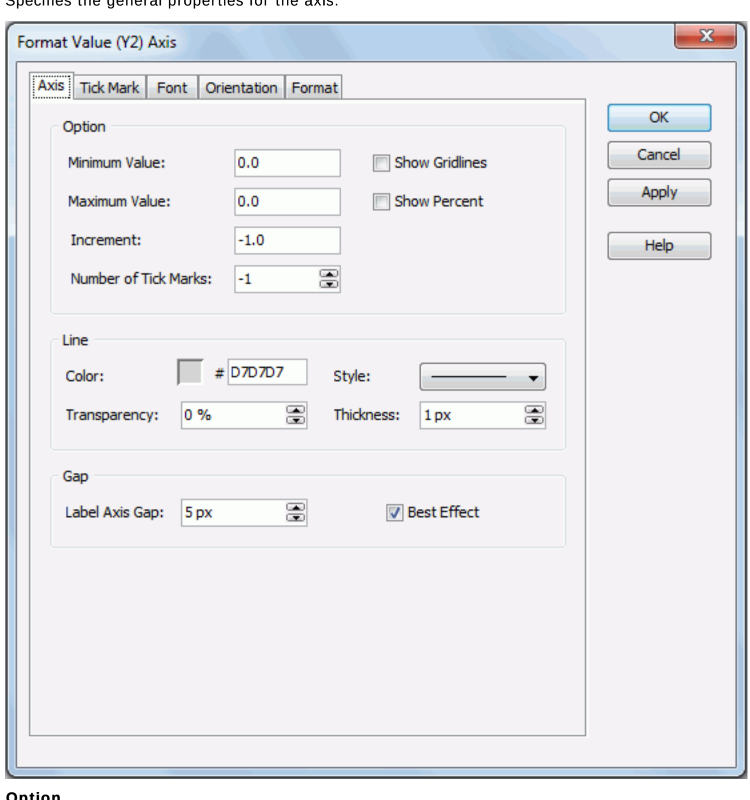
Applies all changes and leaves the dialog open.

Help

Displays the help document about this feature.

Axis tab

Specifies the general properties for the axis.



Option

Specifies the options for the axis.

- **Minimum Value**
  Specifies the minimum value that is to be displayed on the axis.
- **Maximum Value**
  Specifies the maximum value that is to be displayed on the axis.
- **Increment**
  Specifies the difference between two adjacent values on the axis.
- **Number of Tick Marks**
  Specifies how many tick marks to be displayed on the axis.
- **Show Gridlines**
  Specifies whether to show gridlines on the axis.
- **Show Percent**
  Specifies whether to show the value labels on the axis in percent. Only applies to bar/bench, line and area chart that are not 100% stacked type.

Line

Specifies the line style for the axis.

- **Color**
  Specifies the color of the axis.

  To change the color, click the color image button to display the color palette in which you can select a color, or click **More Colors** to customize the color within a wider range in the Pick a Color dialog. You can also input the hexadecimal value of a color directly in the text box after #.
- **Style**
  Specifies the style of the axis.
- **Transparency**
  Specifies the transparency for the color of the axis.
- **Thickness**
  Specifies the thickness of the axis, in pixels.

Gap

Specifies the gap properties for the labels on the axis.

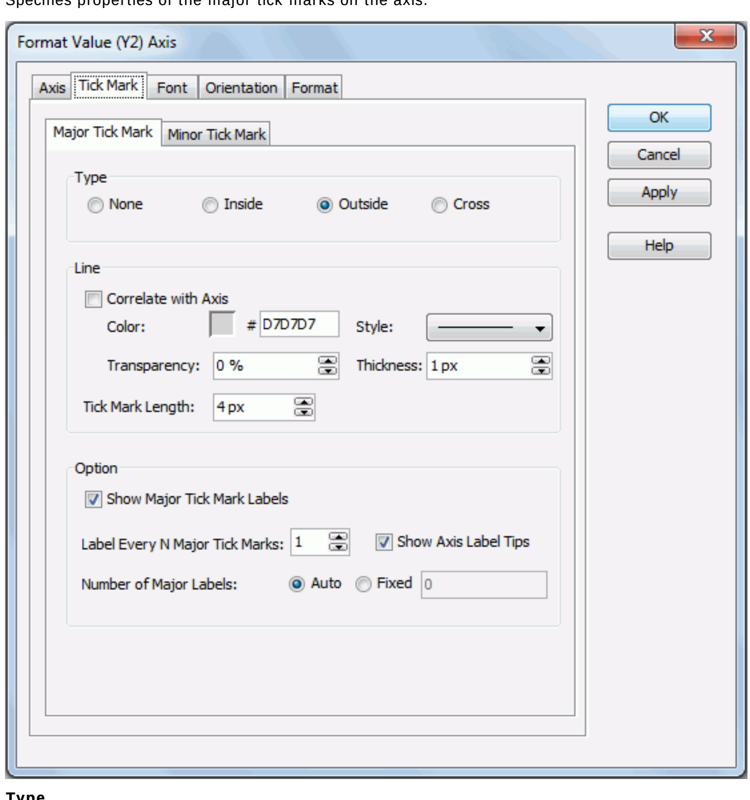
- **Label Axis Gap**
  Specifies the distance between the label and the axis, in pixels.
- **Best Effect**
  Specifies whether to adjust the labels on the axis automatically to make them placed best.

Tick Mark tab

The tab consists of two sub tabs: major tick mark tab, and minor tick mark tab.

Major Tick Mark tab

Specifies properties of the major tick marks on the axis.



Type

Specifies the type of the major tick marks on the axis.

- **None**
  If selected, major tick marks will not be shown on the axis and it will be meaningless to specify all the other major tick mark related properties.
- **Inside**
  If selected, major tick marks will be inside the chart.
- **Outside**
  If selected, major tick marks will be outside the chart.
- **Cross**
  If selected, major tick marks will be across the axis.

Line

Specifies the line properties of the major tick marks on the axis.

- **Correlate with Axis**
  If checked, the line properties of the major tick marks will correlate with that of the axis automatically.
  - **Color**
    Specifies the color of the major tick marks.

    To change the color, click the color image button to display the color palette in which you can select a color, or click **More Colors** to customize the color within a wider range in the Pick a Color dialog. You can also input the hexadecimal value of a color directly in the text box after #.
  - **Transparency**
    Specifies the transparency for the color of the major tick marks.
  - **Style**
    Specifies the type of the major tick marks.
  - **Thickness**
    Specifies the thickness of the major tick marks, in pixels.
- **Tick Mark Length**
  Specifies the length of the major tick marks, in pixels.

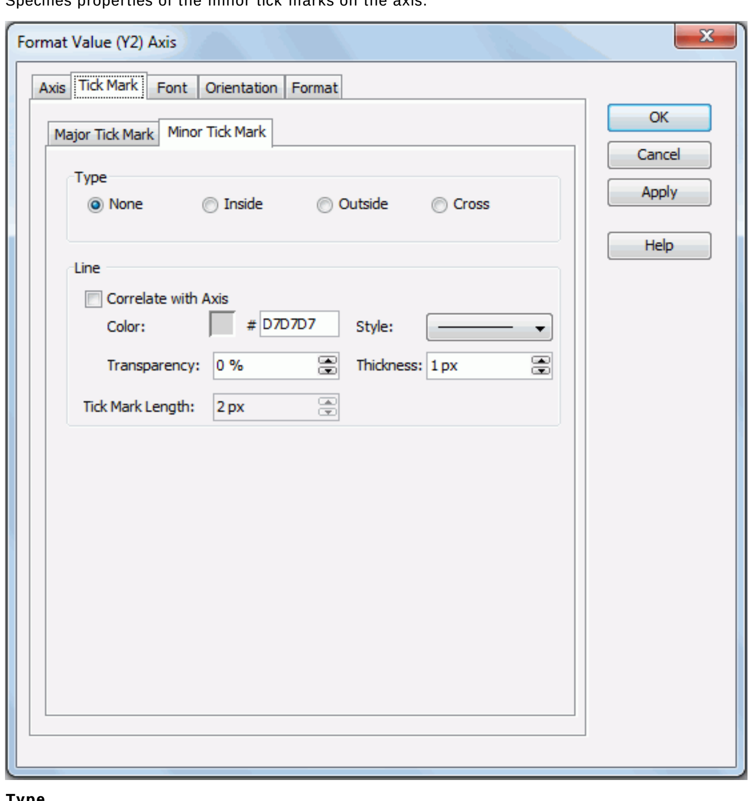
Option

Specifies the other properties of the major tick mark labels on the axis.

- **Show Major Tick Mark Labels**
  Specifies whether to display the labels of the major tick marks on the axis. If checked, the following three properties will be enabled.
- **Label Every N Major Tick Marks**
  Specifies the frequency at which the major tick marks will be labeled.
- **Show Axis Label Tips**
  Specifies whether to display the complete label text when the mouse pointer points at a label on the axis.
- **Number of Major Labels**
  Specifies how many major tick mark labels to be displayed on the axis.
  - **Auto**
    If checked, all major tick mark labels will be shown.
  - **Fixed**
    If checked, you can specify the number of the major tick mark labels to be displayed on the axis.

Minor Tick Mark tab

Specifies properties of the minor tick marks on the axis.



Type

Specifies the type of the minor tick marks on the axis.

- **None**
  If selected, minor tick marks will not be shown on the axis and it will be meaningless to specify all the other minor tick mark related properties.
- **Inside**
  If selected, minor tick marks will be inside the chart.
- **Outside**
  If selected, minor tick marks will be outside the chart.
- **Cross**
  If selected, minor tick marks will be across the axis.

Line

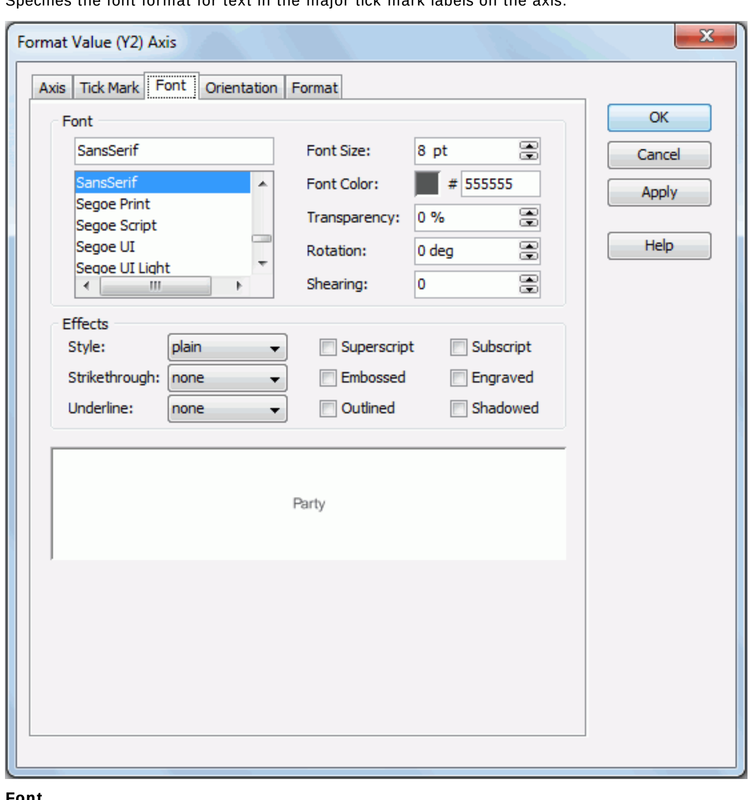
Specifies the line properties of the minor tick marks on the axis.

- **Correlate with Axis**
  If checked, the line properties of the minor tick marks will correlate with that of the axis automatically.
  - **Color**
    Specifies the color of the minor tick marks.

    To change the color, click the color image button to display the color palette in which you can select a color, or click **More Colors** to customize the color within a wider range in the Pick a Color dialog. You can also input the hexadecimal value of a color directly in the text box after #.
  - **Style**
    Specifies the type of the minor tick marks.
  - **Transparency**
    Specifies the transparency for the color of the minor tick marks.
  - **Thickness**
    Specifies the thickness of the minor tick marks, in pixels.
- **Tick Mark Length**
  Specifies the length of the minor tick marks, in pixels.

Font tab

Specifies the font format for text in the major tick mark labels on the axis.



Font

Specifies the font format of text in the major tick mark labels.

- **Font list**
  Lists all the available font faces that can be selected to apply to the text.
- **Font Size**
  Specifies the font size of the text.
- **Font Color**
  Specifies the font color of the text.

  To change the color, click the color image button to display the color palette in which you can select a color, or click **More Colors** to customize the color within a wider range in the Pick a Color dialog, or click **More Fill Effects** to customize in the Fill Effects dialog. You can also input the hexadecimal value of a color directly in the text box after #.
- **Transparency**
  Specifies the transparency of the text.
- **Rotation**
  Specifies the rotation angle of the text around its center, in degrees. The default value is 0.
- **Shearing**
  Specifies the gradient of the text.

Effects

Specifies the special effects of text in the major tick mark labels.

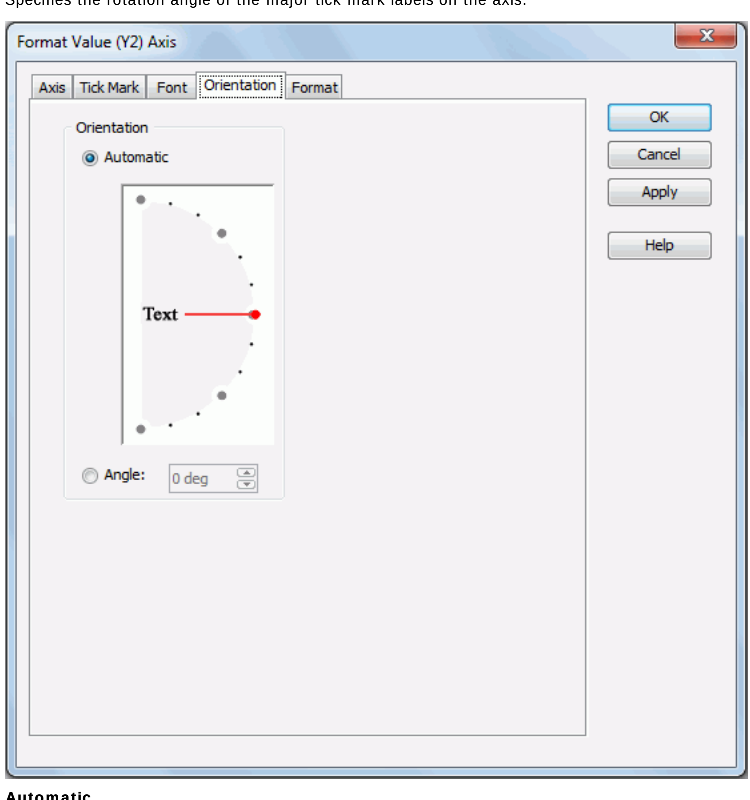
- **Style**
  Specifies the font style of the text. It can be one of the following: plain, bold, italic, and bold italic.
- **Strikethrough**
  Specifies the style of the horizontal line with which the text is struck through. It can be one of the following: none, thin line, bold line, and double lines.
- **Underline**
  Specifies the style of the horizontal line under the text. It can be one of the following: none, single, single lower, bold line, bold lower, double lines, bold double, patterned line, and bold patterned.
- **Superscript**
  Raises the text above the baseline and changes the text to a smaller font size, if a smaller size is available.
- **Embossed**
  Makes the text appear to be raised off the page in relief.
- **Outlined**
  Displays the inner and outer borders of each character.
- **Subscript**
  Lowers the text below the baseline and changes the text to a smaller font size, if a smaller size is available.
- **Engraved**
  Makes the text appear to be imprinted or pressed into the page.
- **Shadowed**
  Adds a shadow beneath and to the right of the text.

Sample

Displays the specified font and any text effects.

Orientation tab

Specifies the rotation angle of the major tick mark labels on the axis.



Automatic

Specifies whether to adjust the rotation angle of the label text on the axis automatically according to the length of the label text, in degrees.

When this option is checked by default:

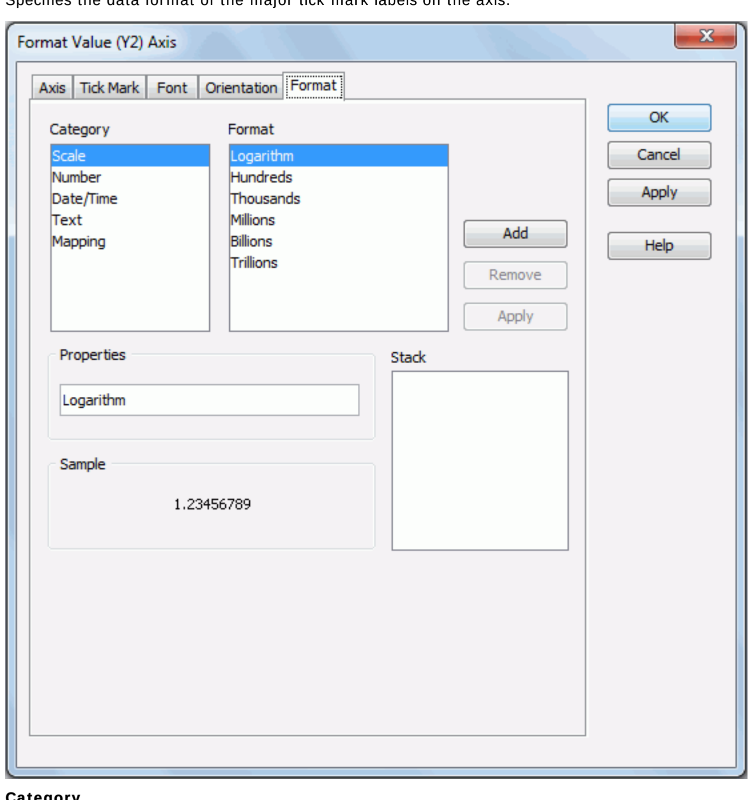
- If the text can be completely displayed horizontally, the default rotation angle will be 0.
- If the text cannot be completely displayed horizontally, the default rotation angle will be 30 anticlockwise, and the cut off part will be shown as suspension points.

Angle

Specifies to customize the rotation angle of the label text on the axis. When you change the rotation angle in the text box, the red line in the spin box will change correspondingly, and vice versa.

Format tab

Specifies the data format of the major tick mark labels on the axis.



Category

Lists the category types. Select one to customize its format.

Format

Displays all the formats of the selected category. Select the required one and click Add to add it as the format of the specified category. You can add only one format for each category. For more details about the formats, see A detailed chart property reference.

Properties

Displays the properties of the format you select. If the formats listed in the Format box cannot meet your requirement, define the format in the text field and then click Add to add it as the format of the specified category.

Sample

Displays the selected format effects.

Stack

Lists all the formats you select from different categories.

Add

Adds a format to the Stack list box.

Remove

Removes a format from the Stack list box.

Apply

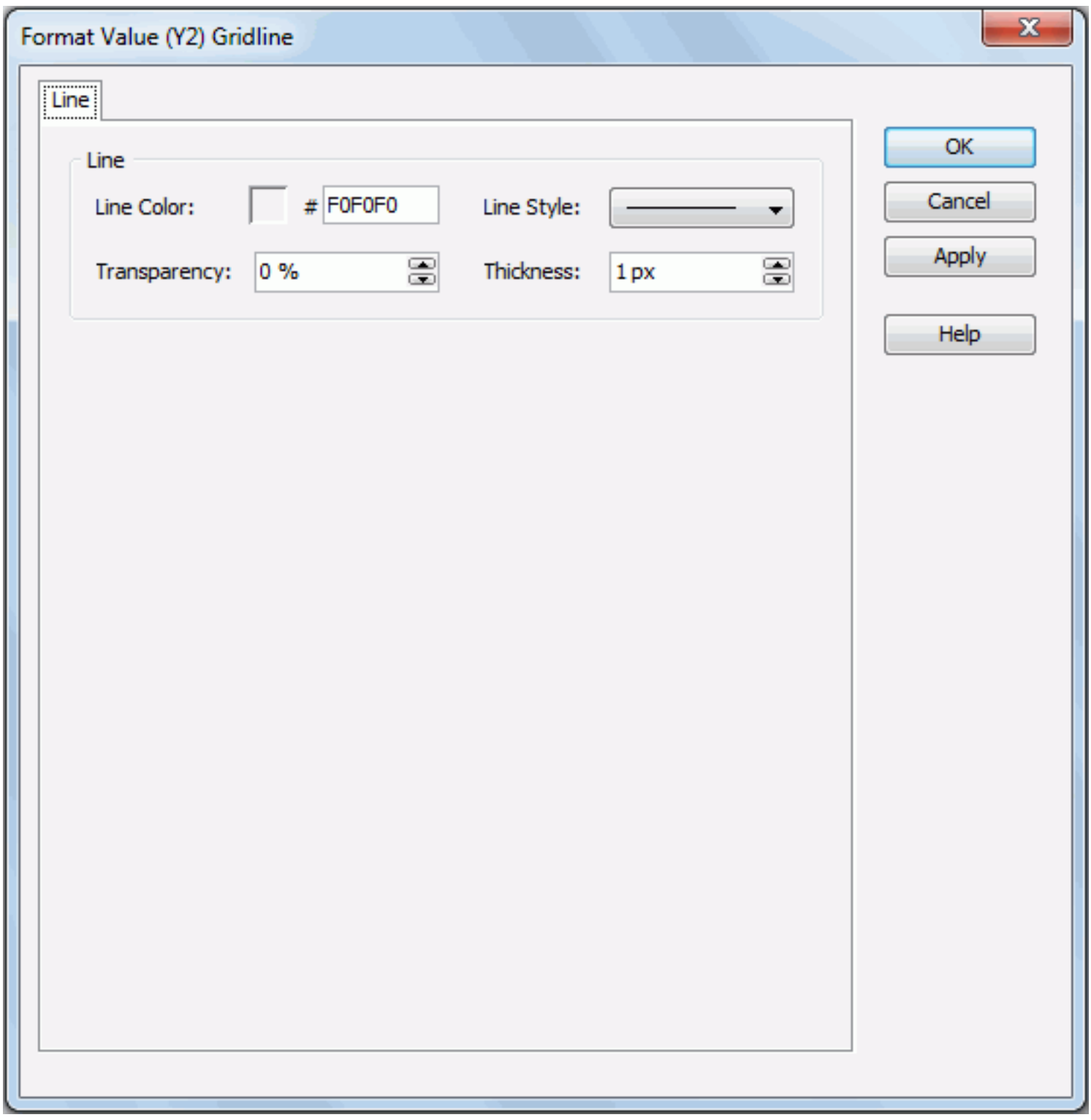
Applies the specified format in the Stack list box to the major tick mark labels.

Related topics:

- Formatting the value axis

Format Value (Y2) Gridline dialog

This dialog appears when you double-click a value (Y2) gridline of a chart, or right-click a chart and then select Format Gridlines > Value (Y2) Gridline from the shortcut menu. It helps you to format the gridlines shown on the value (Y2) axis of a chart.



The following are details about options in the dialog:

Line Color

Specifies the color schema of the gridlines.

To change the color, click the color image button to display the color palette in which you can select a color, or click **More Colors** to customize the color within a wider range in the Pick a Color dialog. You can also input the hexadecimal value of a color directly in the text box after #.

Line Style

Specifies the line style of the gridlines.

Transparency

Specifies the transparency of the color that is applied to the gridlines.

Thickness

Specifies the thickness of the gridlines, in pixels.

OK

Applies the changes and closes the dialog.

Cancel

Does not retain any changes and closes the dialog.

Apply

Applies all changes and leaves the dialog open.

Help

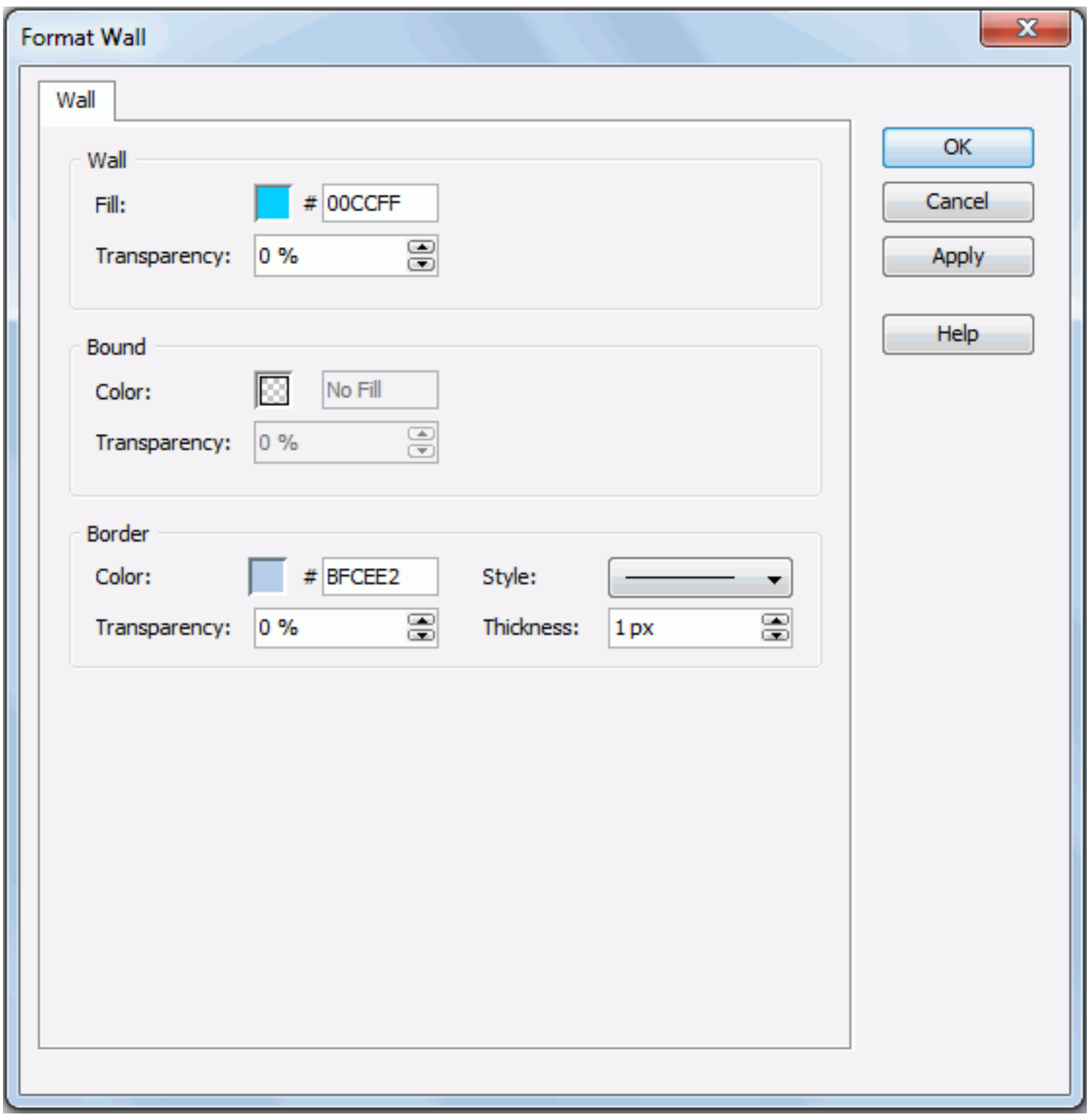
Displays the help document about this feature.

Related topics:

- Formatting the gridlines

Format Wall dialog

This dialog appears when you double-click the wall of a chart, or right-click a chart and then select Format Walls > Format Wall from the shortcut menu. It helps you to format the wall of a chart.



The following are details about options in the dialog:

Wall

Specifies the color or the fill effect of the wall of the chart.

- **Fill**
  Specifies the color schema to fill the wall.

  To change the color, click the color image button to display the color palette in which you can select a color, or click **More Colors** to customize the color within a wider range in the Pick a Color dialog, or click **More Fill Effects** to customize in the Fill Effects dialog. You can also input the hexadecimal value of a color directly in the text box after # if you have changed the default color.

- **Transparency**
  Specifies the transparency of the color schema that is applied to the wall of the chart.

Bound

Specifies the fill color or effect to compound with the basic fill which is set in the Wall area.

- **Color**
  Specifies the fill color schema or effect to compound with the basic fill which is set for the walls of a chart in the Wall area.

  To change the color, click the color image button to display the color palette in which you can select a color, or click **More Colors** to customize the color within a wider range in the Pick a Color dialog, or click **More Fill Effects** to customize in the Fill Effects dialog. You can also input the hexadecimal value of a color directly in the text box after # if you have changed the default color.

- **Transparency**
  Specifies the transparency of the color schema that is applied here.

Border

Specifies the color and its transparency for the border of the wall.

- **Color**
  Specifies the color schema for the border of the wall.

  To change the color, click the color image button to display the color palette in which you can select a color, or click **More Colors** to customize the color within a wider range in the Pick a Color dialog. You can also input the hexadecimal value of a color directly in the text box after #.

- **Transparency**
  Specifies the transparency of the color schema that is applied to the border of the wall.

- **Style**
  Specifies the style for the border of the wall.

- **Thickness**
  Specifies the thickness for the border of the wall, in pixels.

OK

Applies the changes and closes the dialog.

Cancel

Does not retain any changes and closes the dialog.

Apply

Applies all changes and leaves the dialog open.

Help

Displays the help document about this feature.

Related topics:

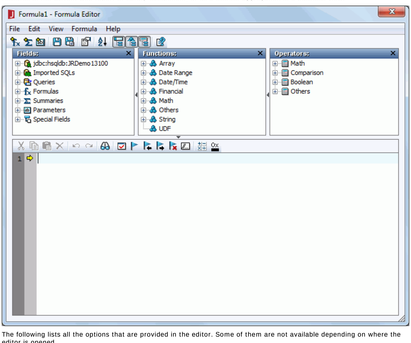
- Formatting the wall/floor

Formula Editor window

This window appears when you do either of the following:

- Right-click the Formulas node, select Add Formula from the shortcut menu, input a name for the new formula and then click OK, or right-click a formula and click Edit Formula on its shortcut menu in the Catalog Browser.
- In the Resource View panel of the JReport Designer main window, or in the Resources box of the report wizard, expand the Dynamic Resources > Formulas node, click <Add Formula>gt;.
- Click the Settings button in the General tab of the Add View Element dialog or Edit View Element dialog.

It helps you to write a formula using the available resources, which may be used generally in JReport, as a dynamic resource in the current web report or library component, or as a custom aggregation in a business view.



The following lists all the options that are provided in the editor. Some of them are not available depending on where the editor is opened.

File menu

- **New Formula**
  Creates a new formula.
- **New Summary**
  Opens the New Summary dialog to create a new summary.
- **New Parameter**
  Opens the Create Parameter dialog to create a new parameter.
- **Import UDF Classes**
  Opens the UDF Classes dialog to import user defined formula classes.
- **Save**
  Saves the formula.
- **Save As**
  Saves the formula with a different name.
- **Close**
  Closes the window.

Edit menu

- **Undo**
  Undoes an action.
- **Redo**
  Redoes an action.
- **Cut**
  Cuts the selected text in the editing panel.
- **Copy**
  Copies the selected text in the editing panel.
- **Paste**
  Pastes the text that was previously cut or copied in the editing panel.
- **Delete**
  Deletes the selected text in the editing panel.
- **Search**
  Brings out the Find/Replace dialog. You can then search for text in the editing panel, and replace the found text with different text.

View menu

- **Fields Panel**
  Specifies whether to show the Fields panel.
- **Functions Panel**
  Specifies whether to show the Functions panel.
- **Operators Panel**
  Specifies whether to show the Operators panel.
- **Sort Tree**
  Sorts functions in the Functions panel and fields in the Fields panel by names.

Formula menu

- **Check**
  Tests the syntax of your formula. If the syntax is incorrect, JReport gives the opportunity to correct the errors.
- **Add Bookmark**
  Adds a bookmark to a specific position.
- **Go to Previous Bookmark**
  Goes to the previous bookmark.
- **Go to Next Bookmark**
  Goes to the next bookmark.
- **Clear all Bookmarks**
  Clears all of the bookmarks.
- **Comment/Uncomment**
  Adds or removes comments.
- **Add Operators**
  Specifies a general operator to be used in the editing panel.
- **Color Converter**
  A color palette is provided for you to insert the HEX code of a color simply by clicking the corresponding color in the color palette instead of inputting the HEX code manually. If the color palette cannot meet your requirement, click More Color to open the Pick a Color dialog, then you can set a full range of color in the dialog.
- **Auto Finish**
  Enables the automatic input of the other part of a sign pair right after you type in the first part when editing formula.
- **Formula References**
  Shows the formulas that reference the selected UDF function in the Functions panel.
- **Properties**
  Shows properties of the formula in the Formula Properties dialog.
  - **Data Type**
    The data type of the returned value of the formula.
  - **Precision**
    The precision of the returned value of the formula. If the precision is set to 0, all characters of the returned value will be displayed; if it is set to N, only the first N characters will be displayed.
  - **Scale**
    The number of digits to the right of the decimal point for the returned value of the formula.

Help menu

- **Formula Editor help**
  Displays the help document about this feature.

Toolbar

Lists the commands on the toolbar.

- **New Formula**
  Creates a new formula.
- **New Summary**
  Creates a new summary.
- **New Parameter**
  Creates a new parameter.
- **Save**
  Saves the formula.
- **Save As**
  Saves the formula with a different name.
- **Properties**
  Specifies the precision and scale properties of the formula.
- **Sort Tree**
  Sorts functions in the Functions panel and fields in the Fields panel by names.
- **Fields Panel**
  Specifies whether to show the Fields panel.
- **Functions Panel**
  Specifies whether to show the Functions panel.
- **Operators Panel**
  Specifies whether to show the Operators panel.
- **Formula Editor Help**
  Displays the help document about this feature.
- **Use As**
  Available when it is a dynamic formula. It specifies to use the formula as one of the following types: Group, Detail or Aggregation.

  Whether a dynamic formula can be used as a certain type depends on the following rule:
  - Any formula can be used as Detail.
  - Record level pass 1 formula can be used as Group.
  - Comparable and record level pass 1 formula can be used as Group.
  - Group level formula can be used as Aggregation.
- **Cut**
  Cuts the selected text in the editing panel.
- **Copy**
  Copies the selected text in the editing panel.
- **Paste**
  Pastes the text that was previously cut or copied in the editing panel.
- **Delete**
  Deletes the selected text in the editing panel.
- **Undo**
  Undoes an action.
- **Redo**
  Cancels undoing an action.
- **Search**
  Searches for text in the editing panel, and replace the found text with different text.
- **Check**
  Tests the syntax of your formula. If the syntax is incorrect, JReport gives the opportunity to correct the errors.
- **Add Bookmark**
  Adds a bookmark to a specific position.
- **Go to Previous Bookmark**
  Goes to the previous bookmark.
- **Go to Next Bookmark**
  Goes to the next bookmark.
- **Clear All Bookmarks**
  Clears all of the bookmarks.
- **Comment/Uncomment**
  Adds or removes comments.
- **Add Operators**
  Selects a general operator to be used in the editing panel.
- **Color Converter**
  A color palette is provided for you to insert the HEX code of a color simply by clicking the corresponding color in the color palette instead of inputting the HEX code manually.

Fields panel

Displays a list of those fields that are available for formulas. Those fields may contain table columns, formulas, summaries, parameters and special fields. Select one field and double-click it to insert it into the editing panel at the insertion point.

Functions panel

Displays a list of JReport functions and imported user defined formula functions that are available for formulas. When you select one function and double-click it, JReport will insert the selected function into the editing panel at the insertion point completely with its required syntax items (parentheses, commas, and so on). For the usage of these functions, refer to section Built-in functions and User-defined formula functions.

Operators panel

Displays a list of operators that are available for formulas. Select one operator and double-click it to insert the selected operator into the editing panel at the insertion point. For the usage of these operators, refer to section Operators.

Formula text panel

In this panel, you can build and edit your formula. There are several ways to work with formula:

- Select formula components from the Fields, Functions and Operators panels on the top of Formula Editor, and then double-click the components, JReport will then insert them in the formula;
- Type your formula in the editing panel directly;
- Use the above two methods together;
- Paste formula text from the text document of other programs.

The line numbers of the formula are marked in the left of the panel to help you check it easily.

Related topics:

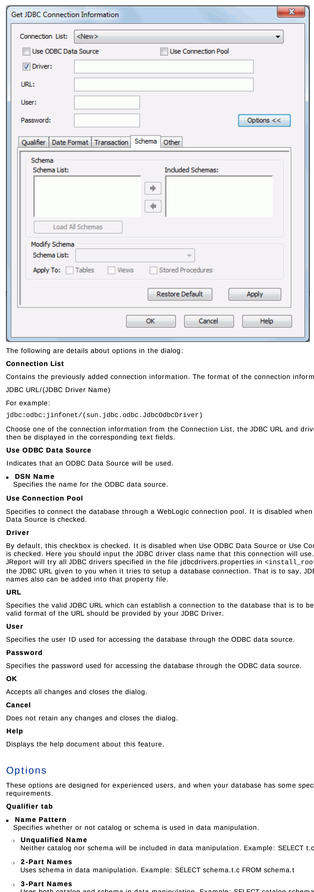
- Creating a formula
- Creating and using dynamic formulas
- Adding a view element

Get JDBC Connection Information dialog

The dialog appears when you do one of the following:

- Select JDBC or HIVE and click OK in the New Data Source dialog.
- In the Data tab of the Catalog Browser, right-click a data source and select Add JDBC Connection or Add HIVE Connection from the shortcut menu.
- In the Data tab of the Catalog Browser, right-click an existing JDBC or HIVE connection and select Edit Connection from the shortcut menu.
- In the Reports tab of the Catalog Browser, right-click the root folder Reports and select Export from the shortcut menu.

The dialog is used to get or create JDBC or HIVE connection information. It lists the database objects which are accessible through JDBC or JDBC-ODBC Bridge.



The following are details about options in the dialog:

Connection List

Contains the previously added connection information. The format of the connection information is:

JDBC URL/(JDBC Driver Name)

For example:

`jdbc:odbc:jinfonet/(sun.jdbc.odbc.JdbcOdbcDriver)`

Choose one of the connection information from the Connection List, the JDBC URL and driver name will then be displayed in the corresponding text fields.

Use ODBC Data Source

Indicates that an ODBC Data Source will be used.

- **DSN Name**
  Specifies the name for the ODBC data source.

Use Connection Pool

Specifies to connect the database through a WebLogic connection pool. It is disabled when Use ODBC Data Source is checked.

Driver

By default, this checkbox is checked. It is disabled when Use ODBC Data Source or Use Connection Pool is checked. Here you should input the JDBC driver class name that this connection will use. Otherwise, JReport will try all JDBC drivers specified in the file jdbcdrivers.properties in `<install_root>\bin` for the JDBC URL given to you when it tries to setup a database connection. That is to say, JDBC driver's names also can be added into that property file.

URL

Specifies the valid JDBC URL which can establish a connection to the database that is to be used. The valid format of the URL should be provided by your JDBC Driver.

User

Specifies the user ID used for accessing the database through the ODBC data source.

Password

Specifies the password used for accessing the database through the ODBC data source.

OK

Accepts all changes and closes the dialog.

Cancel

Does not retain any changes and closes the dialog.

Help

Displays the help document about this feature.

Options

These options are designed for experienced users, and when your database has some special requirements.

Qualifier tab

- **Name Pattern**
  Specifies whether or not catalog or schema is used in data manipulation.
  - **Unqualified Name**
    Neither catalog nor schema will be included in data manipulation. Example: SELECT t.c FROM t
  - **2-Part Names**
    Uses schema in data manipulation. Example: SELECT schema.t.c FROM schema.t
  - **3-Part Names**
    Uses both catalog and schema in data manipulation. Example: SELECT catalog.schema.t.c from catalog.schema.t

  This is very useful when the database contains tables with the same names in different catalogs and/or schemas. If you only use an unqualified name, the SQL statement generated by JReport will not contain the catalog and schema name, and therefore the result will be unpredictable.
- **Quote Qualifier**
  Specifies the characters, and then the qualifier name which contains the characters that will not be quoted.

  In JReport, you need quote with the character defined by the DBMS. The default character is double quotation marks (" "). Two properties exist for the Quote Qualifier.
  - **Default (JDBC)**
    If selected, the program will get the extra name characters from JDBC.
  - **User Defined**
    If selected, you can modify the quote character according to the database system being used.
    - **Extra Characters**
      Specifies the extra characters for identifier names.
    - **Quote Character**
      Specifies the quote character.
- **Modify Qualifier**
  Edits the current qualifier.
  - **Qualifier**
    Lists all the qualifiers available. You can change the current qualifier by selecting one from the drop-down list. The default value is the first one in the list. If the database do not support qualifier, there will be no drop-down list, and you cannot input the qualifier name.
  - **Apply To**
    - **Tables**
      If checked, the modified qualifier will be applied to all tables in the connection.
    - **Views**
      If checked, the modified qualifier will be applied to all views in the connection.
    - **Imported SQLs**
      If checked, the modified qualifier will be applied to all imported SQLs in the connection
    - **Stored Procedures**
      If checked, the modified qualifier will be applied to all stored procedures in the connection.

      Note: All the changes you made in the Modify Qualifier box will not be recorded in the Get JDBC Connection Information dialog. That is to say, if you change any value in the box and apply it, when you open the dialog next time, it will keep to the original.
- **Restore Default**
  Restores values in the tab to defaults.
- **Apply**
  Accept the changes.

Date Format tab

Different database systems use different Date and Time formats for Date and Time values. The Date Format tab is for specifying the format, so that when JReport sends the SQL statement with Date or Time parameters, they can be correctly translated into the format of your database system.

For example, in an Oracle database, by default, the Date format is 'dd-MMM-yy'. However, in a Sybase database system, the Date format is 'yyyy-MM-dd'. Thus, the Date format in the connection object is required to be modified accordingly, in order for the JReport generated SQL statements to function correctly with the database.

- **Date Format**
  Specifies the date format.
- **Time Format**
  Specifies the time format.
- **Timestamp Format**
  Specifies the timestamp format.

Symbol	Meaning	Presentation
y	year	number
M	month in year	number
d	day in month	number
h	hour in day (0~23)	number
m	minute in hour	number
s	second in minute	number
S	millisecond	number

Transaction tab

- **Read Only**
  - **Default**
    Accepts the settings from the database that you are connecting to.
  - **Read Only**
    Users will only be able to access the database in read-only mode.
  - **Read & Write**
    Users will be able to access the database in read-write mode.
- **Transaction Mode**
  - **Default**
    Accepts the settings from the database that you are connecting to.
  - **None**
    Transactions are not supported.
  - **Read Uncommitted**
    Transactions are not isolated from each other. If the DBMS supports other transaction isolation levels, it ignores whatever mechanism it uses to implement those levels. So that they do not adversely affect other transactions, transactions running at the Read Uncommitted level are usually read-only.

    Note: When Read Only is set and Transaction Isolation to be Read Uncommitted, it provides faster access to the data but if a database transaction is rolled back the read will retrieve an invalid row and result in an inaccurate report.
- **Read Committed**
  When one transaction is reading, updating or deleting a row, other transactions cannot update or delete it until this transaction has been committed or rolled back.
- **Repeatable Read**
  When one transaction is updating, deleting several rows or inserting new rows into them, the other transactions cannot also update or delete these rows, until this transaction has been committed or rolled back.
- **Serializable**
  When one transaction is updating, deleting several rows or inserting new rows into them, the other transactions cannot update, delete these rows, or insert any new rows into these rows, until this transaction has been committed or rolled back.

Schema tab

- **Schema**
  Enables you to choose the schemas you want and makes JReport show the chosen schemas in the Add Table dialog and Add View dialog.
  - **Schema List**
    Lists the schemas for you to choose.
  - **Included Schemas**
    Shows the selected schemas.
  - **Load All Schemas**
    Enables JReport Designer to show all the schemas in database.
  - Adds the selected schemas to the Included Schemas box.
  - Removes the selected schemas from the Included Schemas box.
- **Modify Schema**
  Edits the current schema.
  - **Schema List**
    Lists all the schemas available.
  - **Apply To**
    - **Tables**
      If checked, the selected schema will be applied to all tables in the connection.
    - **Views**
      If checked, the selected schema will be applied to all views in the connection.
    - **Stored Procedures**
      If checked, the selected schema will be applied to all stored procedures in the connection.

      Note: All the changes you made in the Modify Schema box will not be recorded in the Get JDBC Connection Information dialog. That is to say, if you change any value in the box and apply it, when you open the dialog next time, it will keep to the original.
- **Restore Default**
  Restores values in the tab to defaults.
- **Apply**
  Accepts the changes.

Other tab

When parameters are used to build queries, there may be special characters in the parameter values like a back slash (\), and you would like them to be interpreted literally, rather than as a special character. Then you can replace the special characters with another string according to your database. For example, for MySQL database, use \\ to represent \.

- **Characters to Be Replaced**
  Specifies the characters that need to be replaced in parameter values when the parameter values are passed onto queries.
- **Replaced By**
  Specifies the characters to replace with.

Related topics:

- Setting up a JDBC connection
- Exporting the report information to a table

Geographic Map Wizard for library component

This wizard appears when you right-click a geographic map and then select Geographic Map Wizard from the shortcut menu. It helps you modify a geographic map that has been created.

The wizard consists of the following screens:

- Data screen
- Display screen

Back
Goes back to the previous screen.

Next
Goes to the next screen.

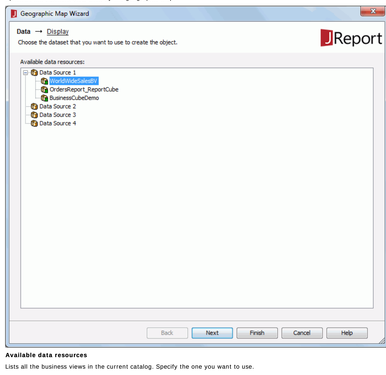
Finish
Finishes modifying the geographic map and closes this dialog.

Cancel
Does not retain changes and closes this dialog.

Help
Displays the help document about this feature.

Data screen

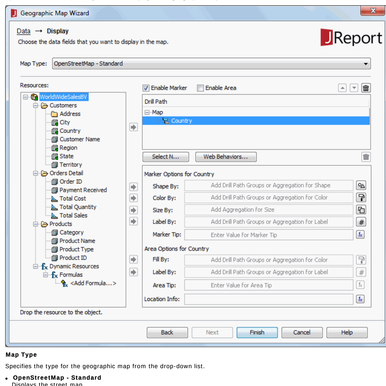
Specifies the business view used by the geographic map.



Available data resources
Lists all the business views in the current catalog. Specify the one you want to use.

Display screen

Specifies the data fields that you want to display in the geographic map.



Map Type
Specifies the type for the geographic map from the drop-down list.

- **OpenStreetMap - Standard**
  Displays the street map.
- **OpenCycleMap - Cycle Map**
  Displays the cycle route map.
- **OpenCycleMap - Transport Map**
  Displays the transport map.
- **Google Maps - Roadmap**
  Displays Google road map.
- **Google Maps - Satellite**
  Displays Google Earth satellite images.
- **Google Maps - Terrain**
  Displays a physical map based on terrain information.

Resources
Lists all the available data resources. You can also create dynamic formulas to use in the map.

Adds the selected field to the geographic map.

Enable Marker
Enables specifying the marker properties for all the groups.

Enable Area
Enables specifying the area properties for all the groups.

One option between Enable Marker and Enable Area must be checked at least and Enable Marker is checked by default.

Drill Path
Lists the fields that are used to group data in the geographic map.

Moves the specified group one step up.

Moves the specified group one step down.

Removes the selected fields that are not required from the geographic map.

Select N
Opens the Select N dialog to specify the Select N condition for the selected group level.

Web Behaviors
Opens the Web Behaviors dialog to specify some web behaviors to the markers and areas of the selected group level.

Marker Options
Specifies the marker properties of the selected group in the Drill Path box. Available only when Enable Marker is checked.

- **Shape By**
  Specifies the field used to control the shapes of the markers for the selected group level. Only the group field that represents the currently selected group level, or a group field of a higher level than the current level in the Drill Path box, or an aggregation field can be added.
  - Specifies the shapes of the markers.
    - If no field is added, select a shape from the shape list, or check and click the Advanced option to open the Advanced dialog to specify an image as the shape of the markers.
    - If a field is added, opens the Edit Shapes dialog to specify shapes for the markers.
- **Color By**
  Specifies the field used to control the colors of the markers for the selected group level. Only the group field that represents the currently selected group level, or group fields of higher levels than the current level in the Drill Path box, or an aggregation field can be added. When more than one group field is added, you can adjust the order of the fields by dragging and dropping them in the text box.
  - Specifies the colors of the markers.
    - If no field is added, select a color from the color palette or click **More Colors** on the color palette to open the Pick a Color dialog to specify a color for the markers.
    - If one or more group fields are added, opens the Edit Color dialog to specify colors for the markers.
    - If an aggregation field is added, opens the Edit Gradient Color dialog to specify a gradient color for the markers.
- **Size By**
  Specifies the field used to control the size of the markers for the selected group level. Only an aggregation field can be added.
  - Specifies the size of the markers.
    - **Zoom percentage bar**
      Specifies the zoom percentage based on the default size.
    - **Reset**
      Specifies whether to restore the default size.
- **Label By**
  Specifies the field to control the labels shown above the markers for the selected group level. When there are more than one field, you can adjust the order of the fields by dragging and dropping them in the text box.
  - Specifies the font format of the label text.
    - Arial
      Specifies the font of the label text.
    - 9
      Specifies the font size of the label text.
    - **B**
      Specifies whether to make the label text bold.
    - **I**
      Specifies whether to make the label text italic.
    - **U**
      Specifies whether to underline the text.
    - **A**
      Specifies the color of the label text.
    - **$**
      Specifies the format for the field you currently select from the preview box below.
    - **Preview box**
      Displays the specified font format effects.
    - **OK**
      Applies the specified font format and closes the dialog.
- **Marker Tip**
  Specifies the tip information when you hover the mouse on the markers at runtime.

Area Options
Specifies the area properties of the selected group in the Drill Path box. Available only when Enable Area is checked.

- **Fill By**
  Specifies the field to control the colors that are used to fill the areas of the selected group level. Only the group field that represents the currently selected group level, or group fields of higher levels than the current level in the Drill Path box, or an aggregation field can be added. When more than one group field is added, you can adjust the order of the fields by dragging and dropping them in the text box.
  - Specifies the colors to fill the areas.
    - If no field is added, select a color from the color palette or click **More Colors** on the color palette to open the Pick a Color dialog to specify a color for the areas.
    - If one or more group fields are added, opens the Edit Color dialog to specify colors for the areas.
    - If an aggregation field is added, opens the Edit Gradient Color dialog to specify a gradient color for the areas.
- **Label By**
  Specifies the field to control the labels shown above the areas for the selected group level. When more than one field is added, you can adjust the orderof the fields by dragging and dropping.
  - Specifies the font format of the label text.
    - Arial
      Specifies the font of the label text.
    - 9
      Specifies the font size of the label text.
    - **B**
      Specifies whether to make the label text bold.
    - **I**
      Specifies whether to make the label text italic.
    - **U**
      Specifies whether to underline the text.
    - **A**
      Specifies the color of the label text.
    - **$**
      Specifies the format for the field you currently selected from the preview box below.
    - **Preview box**
      Displays the specified font format effects.
    - **OK**
      Applies the specified font format and closes the dialog.
- **Area Tip**
  Specifies the tip information when you hover the mouse on the areas at runtime.

Location Info
Specifies the mapping location of the markers or areas for the selected group by field in the Drill Path box. For each group level that you want to show on the map, specify a DBfield or formula that contains the value matching the folder name in `<install_root>\gisinfo\geocode` used by the geographic map. For example you may group by Country and State. For State, you would want to use a formula fStateCountry that concatenates state and country so Google Maps, OpenStreetMap or OpenCycleMap knows which state to use such as "New York, USA" or "California, USA".

Related topics:

- Inserting a geographic map to a library component

Geographic Map Wizard for page report

This wizard appears when you right-click a geographic map and then select Geographic Map Wizard from the shortcut menu. It helps you modify a geographic map that has been created.

The wizard consists of the following screens:

- Data screen
- Group screen
- Layout screen
- Marker screen

Back

Goes back to the previous screen.

Next

Goes to the next screen.

Finish

Finishes modifying the geographic map and closes this dialog.

Cancel

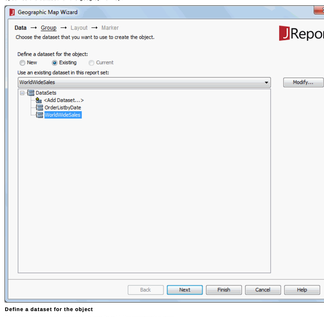
Does not retain changes and closes this dialog.

Help

Displays the help document about this feature.

Data screen

Specifies the dataset for the geographic map.



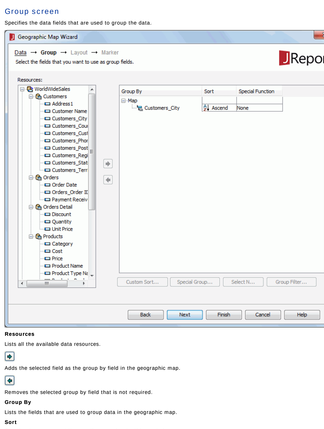
Define a dataset for the object

Specifies the dataset you want to use to edit the geographic map.

- **New**
  If checked, select a data source from the catalog resources to create a dataset that will be used to build the geographic map. When you choose to create the dataset from a query, you can click the Modify button to edit the query in the Query Editor if required.
- **Existing**
  If checked, select a dataset from the ones existing in the open report to create the geographic map. Click the Modify button to edit the dataset in the Dataset Editor if required.
- **Current**
  If checked, the current dataset used by the parent object will be applied to the geographic map.

Group screen

Specifies the data fields that are used to group the data.



Resources

Lists all the available data resources.

Adds the selected field as the group by field in the geographic map.

Removes the selected group by field that is not required.

Group By

Lists the fields that are used to group data in the geographic map.

Sort

Specifies how groups at the specific group level will be sorted.

- **Ascend**
  Groups will be sorted in an ascending order (A, B, C).
- **Descend**
  Groups will be sorted in a descending order (C, B, A).
- **No Sort**
  Groups will be sorted in the original order in database.
- **Special Group**
  Opens the User Defined Group dialog to define grouping information.

  For example, if you placed a field named Region for grouping, and this field contains all 50 states of the United States, and if you want to see the data between Maryland (MD) and New York (NY), you can define the criteria by selecting the between operator to further define your grouping information.
- **Custom Sort**
  Opens the Custom Sort dialog to set how groups will be sorted.

Special Function

If the group by field is of Numeric/String/Date/Time type, you can select a special function for the field in the Special Function column to further specify to which level the data will be grouped by.

If Customize is selected, the Customized Function dialog will be displayed, in which you can set the function by your own.

Custom Sort

Specifies how to sort the groups. Activated only when you have clicked Custom Sort from the Sort column to define the sorting manner of groups for the selected group level.

Special Group

Specifies how to group your information. Activated only when you have clicked Special Group from the Sort column to define a special group.

Select N

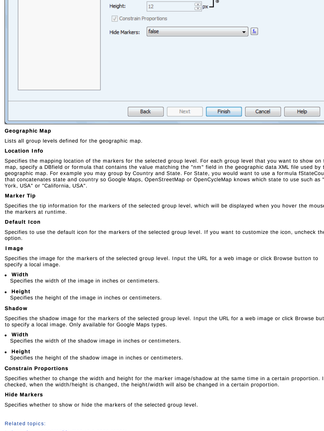
Opens the Select N dialog to specify the Select N condition.

Group Filter

Opens the Group Filter dialog to specify the group filter condition.

Layout screen

Specifies the layout of geographic map elements.



Geographic Data

Specifies the geographical data for the geographic map with an XML file. If the XML file that you specify does not exist it will be created in the <install_root>\gisinfo directory. The geographic data XML file is used to map the geographic latitude and longitude coordinates from Google Maps, OpenStreetMap or OpenCycleMap based on the location information specified for each group in the Marker tab. When the location information key string of a group value matches the "nm" field in the XML file, you will get the related latitude and longitude value for this group value.

You can also import an XML file which contains predefined geographic data including latitude and longitude for the specific "nm" field. By default, the XML file is located in the <install_root>\gisinfo directory. If not, the file will be copied to this directory after it is imported. When you publish the report from Designer you can specify the gisinfo files to publish to the Server from the Geographic Information tab.

If you do not publish the geographical information it will be created automatically when you first run the report.

Map Type

Specifies the type for the geographic map from the drop-down list.

- **Google Maps - Roadmap**
  Displays Google road map.
- **Google Maps - Satellite**
  Displays Google Earth satellite images.
- **Google Maps - Terrain**
  Displays a physical map based on terrain information.
- **OpenStreetMap - Standard**
  Displays the street map.
- **OpenCycleMap - Cycle Map**
  Displays the cycle-route map.
- **OpenCycleMap - Transport Map**
  Displays the transport map.

Marker screen

Specifies properties of the markers bound with the group levels.

Geographic Map

Lists all group levels defined for the geographic map.

Location Info

Specifies the mapping location of the markers for the selected group level. For each group level that you want to show on the map, specify a DBField or formula that contains the value matching the "nm" field in the geographic data XML file used by the geographic map. For example you may group by Country and State. For State, you would want to use a formula fStateCountry that concatenates state and country so Google Maps, OpenStreetMap or OpenCycleMap knows which state to use such as "New York, USA" or "California, USA".

Marker Tip

Specifies the tip information for the markers of the selected group level, which will be displayed when you hover the mouse on the markers at runtime.

Default Icon

Specifies to use the default icon for the markers of the selected group level. If you want to customize the icon, uncheck the option.

Image

Specifies the image for the markers of the selected group level. Input the URL for a web image or click Browse button to specify a local image.

- **Width**
  Specifies the width of the image in inches or centimeters.
- **Height**
  Specifies the height of the image in inches or centimeters.

Shadow

Specifies the shadow image for the markers of the selected group level. Input the URL for a web image or click Browse button to specify a local image. Only available for Google Maps types.

- **Width**
  Specifies the width of the shadow image in inches or centimeters.
- **Height**
  Specifies the height of the shadow image in inches or centimeters.

Constrain Proportions

Specifies whether to change the width and height for the marker image/shadow at the same time in a certain proportion. If checked, when the width/height is changed, the height/width will also be changed in a certain proportion.

Hide Markers

Specifies whether to show or hide the markers of the selected group level.

Related topics:

- Inserting a geographic map to a page report

Geographic Map Wizard for web report

This wizard appears when you right-click a geographic map and then select Geographic Map Wizard from the shortcut menu. It helps you modify a geographic map that has been created.

The wizard consists of the following screens:

- Data screen
- Display screen

Back

Goes back to the previous screen.

Next

Goes to the next screen.

Finish

Finishes modifying the geographic map and closes this dialog.

Cancel

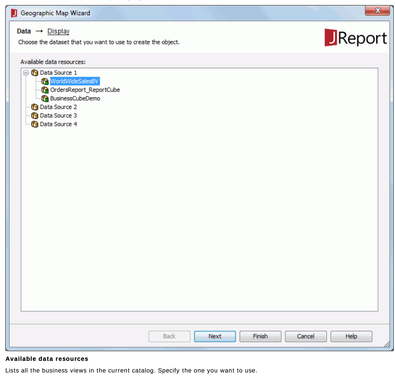
Does not retain changes and closes this dialog.

Help

Displays the help document about this feature.

Data screen

Specifies the business view used by the geographic map.

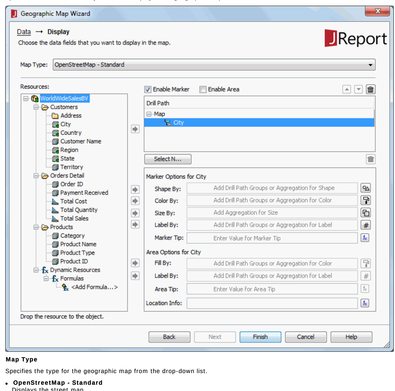


Available data resources

Lists all the business views in the current catalog. Specify the one you want to use.

Display screen

Specifies the data fields that you want to display in the geographic map.



Map Type

Specifies the type for the geographic map from the drop-down list.

- **OpenStreetMap - Standard**
  Displays the street map.
- **OpenCycleMap - Cycle Map**
  Displays the cycle route map.
- **OpenCycleMap - Transport Map**
  Displays the transport map.
- **Google Maps - Roadmap**
  Displays Google road map.
- **Google Maps - Satellite**
  Displays Google Earth satellite images.
- **Google Maps - Terrain**
  Displays a physical map based on terrain information.

Resources

Lists all the available data resources. You can also create dynamic formulas to use in the map.

Adds the selected field to the geographic map.

Enable Marker

Enables specifying the marker properties for all the groups.

Enable Area

Enables specifying the area properties for all the groups.

One option between Enable Marker and Enable Area must be checked at least and Enable Marker is checked by default.

Drill Path

Lists the fields that are used to group data in the geographic map.

Moves the specified group one step up.

Moves the specified group one step down.

Removes the selected fields that are not required from the geographic map.

Select N

Opens the Select N dialog to specify the Select N condition for the selected group level.

Marker Options

Specifies the marker properties of the selected group in the Drill Path box. Available only when Enable Marker is checked.

- **Shape By**
  Specifies the field used to control the shapes of the markers for the selected group level. Only the group field that represents the currently selected group level, or a group field of a higher level than the current level in the Drill Path box, or an aggregation field can be added.
  - Specifies the shapes of the markers.
  - If no field is added, select a shape from the shape list, or check and click the Advanced option to open the Advanced dialog to specify an image as the shape of the markers.
  - If a field is added, opens the Edit Shapes dialog to specify shapes for the markers.
- **Color By**
  Specifies the field used to control the colors of the markers for the selected group level. Only the group field that represents the currently selected group level, or group fields of higher levels than the current level in the Drill Path box, or an aggregation field can be added. When more than one group field is added, you can adjust the order of the fields by dragging and dropping them in the text box.
  - Specifies the colors of the markers.
  - If no field is added, select a color from the color palette or click **More Colors** on the color palette to open the Pick a Color dialog to specify a color for the markers.
  - If one or more group fields are added, opens the Edit Color dialog to specify colors for the markers.
  - If an aggregation field is added, opens the Edit Gradient Color dialog to specify a gradient color for the markers.
- **Size By**
  Specifies the field used to control the size of the markers for the selected group level. Only an aggregation field can be added.
  - Specifies the size of the markers.
  - **Zoom percentage bar**
    Specifies the zoom percentage based on the default size.
  - **Reset**
    Specifies whether to restore the default size.
- **Label By**
  Specifies the field to control the labels shown above the markers for the selected group level. When there are more than one field, you can adjust the order of the fields by dragging and dropping them in the text box.
  - Specifies the font format of the label text.
    - Arial
      Specifies the font of the label text.
    - 8
      Specifies the font size of the label text.
    - **B**
      Specifies whether to make the label text bold.
    - *I*
      Specifies whether to make the label text italic.
    - U
      Specifies whether to underline the text.
    - A
      Specifies the color of the label text.
    - $
      Specifies the format for the field you currently select from the preview box below.
    - **Preview box**
      Displays the specified font format effects.
    - **OK**
      Applies the specified font format and closes the dialog.
- **Marker Tip**
  Specifies the tip information when you hover the mouse on the markers at runtime.

Area Options

Specifies the area properties of the selected group in the Drill Path box. Available only when Enable Area is checked.

- **Fill By**
  Specifies the field to control the colors that are used to fill the areas of the selected group level. Only the group field that represents the currently selected group level, or group fields of higher levels than the current level in the Drill Path box, or an aggregation field can be added. When more than one group field is added, you can adjust the order of the fields by dragging and dropping them in the text box.
  - Specifies the colors to fill the areas.
  - If no field is added, select a color from the color palette or click **More Colors** on the color palette to open the Pick a Color dialog to specify a color for the areas.
  - If one or more group fields are added, opens the Edit Color dialog to specify colors for the areas.
  - If an aggregation field is added, opens the Edit Gradient Color dialog to specify a gradient color for the areas.
- **Label By**
  Specifies the field to control the labels shown above the areas for the selected group level. When more than one field is added, you can adjust the order of the fields by dragging and dropping.
  - Specifies the font format of the label text.
    - Arial
      Specifies the font of the label text.
    - 8
      Specifies the font size of the label text.
    - **B**
      Specifies whether to make the label text bold.
    - *I*
      Specifies whether to make the label text italic.
    - U
      Specifies whether to underline the text.
    - A
      Specifies the color of the label text.
    - $
      Specifies the format for the field you currently selected from the preview box below.
    - **Preview box**
      Displays the specified font format effects.
    - **OK**
      Applies the specified font format and closes the dialog.
- **Area Tip**
  Specifies the tip information when you hover the mouse on the areas at runtime.

Location Info

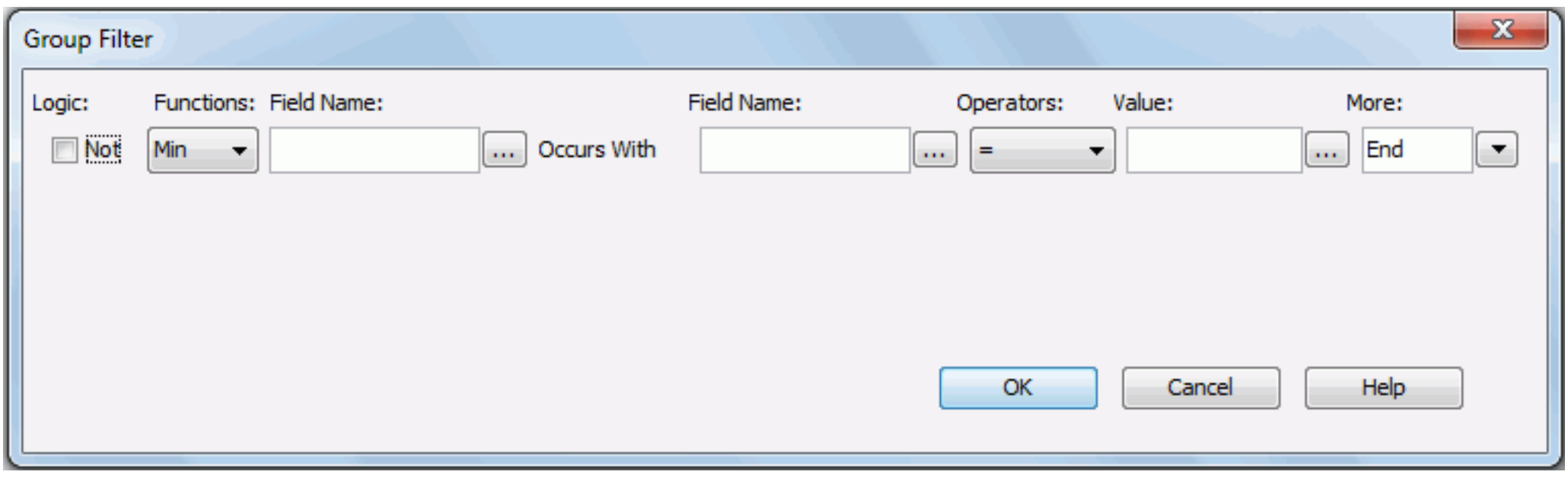
Specifies the mapping location of the markers or areas for the selected group by field in the Drill Path box. For each group level that you want to show on the map, specify a DBfield or formula that contains the value matching the folder name in `<install_root>\gisinfo\geocode` used by the geographic map. For example you may group by Country and State. For State, you would want to use a formula StateCountry that concatenates state and country so Google Maps, OpenStreetMap or OpenCycleMap knows which state to use such as "New York, USA" or "California, USA".

Related topics:

- Inserting a geographic map to a web report

Group Filter dialog

This dialog appears when you click the Group Filter button in the Group screen of the report wizard. It helps you to create and modify more advanced filtering conditions based on the selected group.



The following are details about options in the dialog:

Logic

Specifies whether to evaluate the records that are not in the record aggregate.

Functions

Specifies the function to aggregate on a certain field.

- **Min**
  Evaluates the records that have the lowest values for the selected field.
- **Max**
  Evaluates the records that have the largest values for the selected field.
- **Exist**
  Checks if there exists at least one qualified record.

Field Name (before Occur With)

Specifies the field you want to aggregate on.

Field Name (after Occur With)

Specifies the criteria of the filter.

Expression includes not only the DBField, but also formulas and parameters. You can manually type the strings supported by the database or click to select column names or fields values in the Expressions dialog.

Where,

- **+**
  Adds numbers or fields together in the Expression menu.
- **-**
  Subtracts numbers or fields together in the Expression menu.
- *****
  Multiplies numbers or fields together in the Expression menu.
- **/**
  Divides numbers or fields together in the Expression menu.
- **=**
  Equates fields together.
- **"**
  Places quotations marks on long character strings or names that have blanks. For example, places quotes on values, such as "New York" and "Washington DC".
- **||**
  Places fields together in the same Expression menu. For example: "New York" || "Washington DC".
- **( )**
  Places the fields in parentheses.

Operators

Specifies the operator to compose the filter expression. It can be =, >, <, >=, <= or <>.

Value

Specifies the value used in the filter condition.

You can also use a parameter for runtime input. You can manually type the value or parameter name, or, for parameters, click ... to select the parameter names in the Expressions dialog.

More

Lists some additional commands.

- **END**
  Finishes entering the condition
- **And**
  Inserts the logic operator AND which is applied to this line and the next line.
- **OR**
  Inserts the logic operator OR which is applied to this line and the next line.
- **Insert Row**
  Inserts a single Expression row.
- **Delete Row**
  Deletes a single Expression row.

OK

Accepts the changes and closes the dialog.

Cancel

Closes the dialog, leaving any changes unsaved.

Help

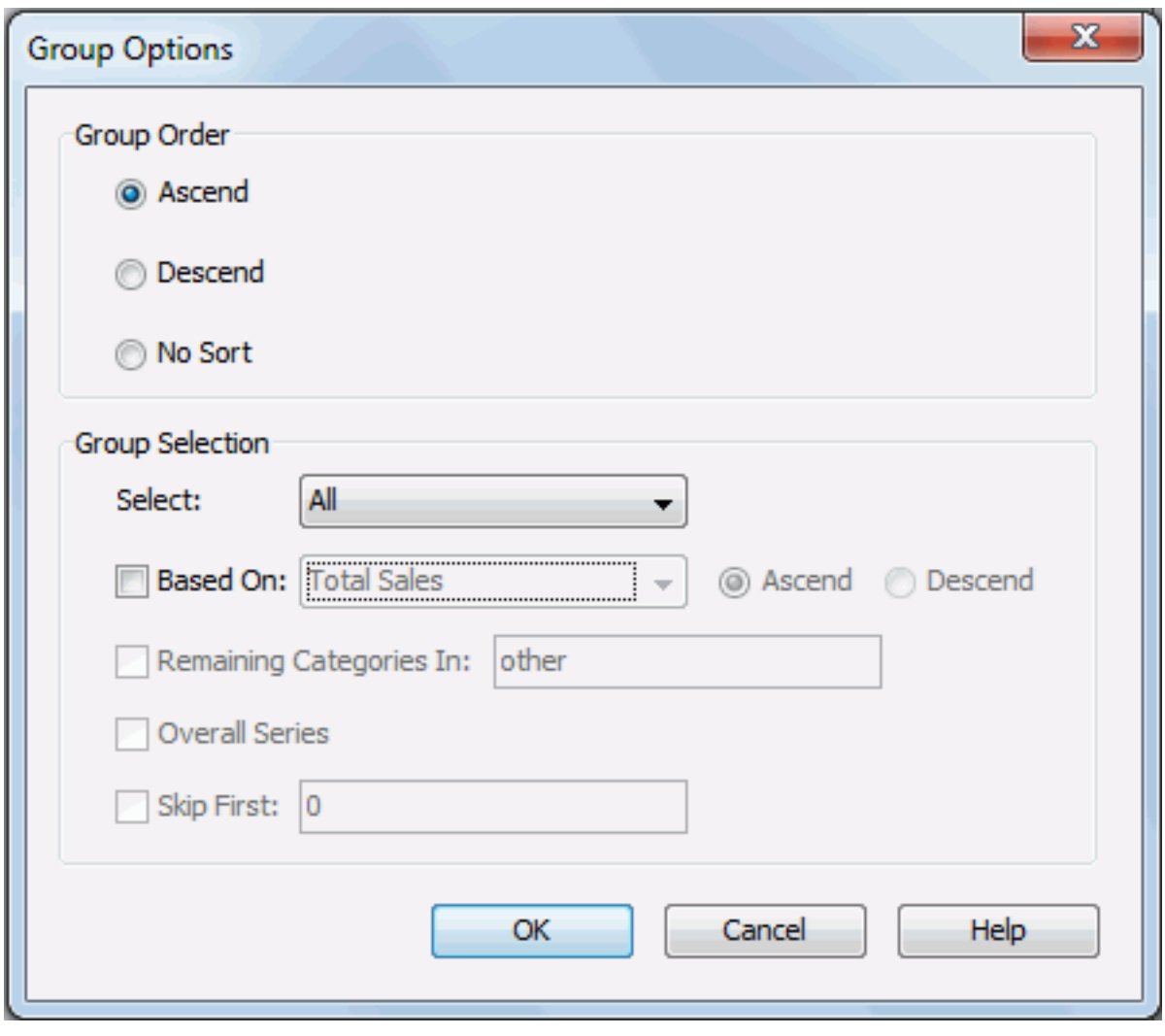
Displays the help document about this feature.

Related topics:

- Filtering the groups

Group Options dialog

The dialog appears when you click the Order/Select N button below the Groups box in the Display screen of the chart wizard when the chart type is heat map. It helps you to specify the sort order of the group values and define the number of the group values that will be displayed in a chart.



The following are details about options in the dialog:

Group Order

Specifies in which manner to sort the group values.

- **Ascend**
  Values will be sorted in an ascending order (A, B, C).
- **Descend**
  Values will be sorted in a descending order (C, B, A).
- **No Sort**
  Values will be sorted in the original order in database.

Group Selection

Specifies the number of group values that will be displayed in the chart.

- **Select**
  Specifies the Select N condition to define the number of the group values that will be displayed.
  - **All**
    If selected, all group values in the chart will be displayed.
  - **Top N**
    If selected, specify a number in the field to the right and the first N group values in the chart will be displayed. You can also select a parameter from the drop-down list, which returns an integer, and then specify a value for the parameter when viewing the report to dynamically define the Top N condition.
  - **Bottom N**
    If selected, specify a number in the field to the right and the last N group values in the chart will be displayed. You can also select a parameter from the drop-down list which returns an integer, and then specify a value for the parameter when viewing the report to dynamically define the Bottom N condition.
- **Based On**
  If checked, the group values will be sorted by values of the summary selected with the direction specified. If unchecked, the group values will be sorted by the order specified in the Group Order box of the dialog.
- **Remaining Categories In**
  Enabled only when Top N or Bottom N is selected from the Select drop-down list. Check this option and then type a character string in the text field to group all the group values beyond the top/bottom N range.
- **Overall Series**
  Not supported on group values.
- **Skip First**
  If you check the Skip First option and input a number M in the text field to the right, then the first M group values in the chart will be skipped and the Select N condition will take effect beginning with the M+1 value. The skipped values will be included in the remaining group together with all the group values beyond the top/bottom N range. Enabled when Top N or Bottom N is selected from the Select drop-down list.

OK

Accepts the changes and exits the dialog.

Cancel

Discards the changes and closes the dialog.

Help

Displays the help document about this feature.

Related topics:

- Setting Order/Select N condition for category/series/group field

Horizontal Banded Wizard

This wizard appears when you select the Horizontal Banded layout in the New Page Report Tab dialog or New Page Report dialog and then click OK. It helps you to create a report with a horizontal banded object in it, and consists of the following screens:

- Data screen
- Display screen
- Group screen
- Summary screen
- Chart screen
- Filter screen
- Style screen

Back
Goes back to the previous screen.

Next
Goes to the next screen.

Finish
Finishes creating the horizontal banded report and closes this wizard.

Cancel
Does not retain changes and closes this wizard.

Help
Displays the help document about this feature.

Data screen

Specifies the dataset used to create the banded object.

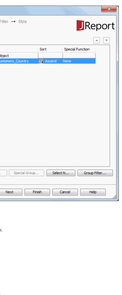
Define a dataset for the object
Specifies the dataset on which the banded object will be built.

New
If checked, select a data source from the catalog resources to create a dataset that will be used to build the banded object. When you choose to create the dataset from a query, you can click the Modify button to edit the query in the Query Editor if needed.

Existing
If checked, select a dataset from the ones existing in the open report to create the banded object. Click the Modify button to edit the dataset in the Dataset Editor if required.

Display screen

Specifies the data fields displayed in the banded object.



Resources
Lists all the available data resources.

Resources
Lists all the available data resources.

Adds the selected field as the group field in the banded object.

Removes the selected group field in the banded object.

Moves the selected group one step up.

Moves the selected group one step down.

Group By
Lists the fields that are used to group data in the banded object.

Sort
Specifies how groups at the specific group level will be sorted.

- **Ascend**
  Groups will be sorted in an ascending order (A, B, C).
- **Descend**
  Groups will be sorted in a descending order (C, B, A).
- **None**
  Groups will be sorted in the original order in database.

- **Special Group**
  Opens the User Defined Group dialog to define grouping information.

  For example, if you placed a field named Region for grouping, and this field contains all 50 states of the United States, and if you want to place the data between Maryland (MD) and New York (NY), you can define the criteria by selecting the between operator in the User Defined Group dialog to define your grouping information.
- **Custom Sort**
  Opens the Custom Sort dialog to set how groups will be sorted.

Special Function
If the group by field is of Numeric/String/Date/Time type, you can select a special function for the field in the Special Function column to further specify in which level the data will be grouped by.

If Customize is selected, the Customized Function dialog will be displayed, in which you can set the function by your own.

Custom Sort
Specifies how to sort the groups. Activated only when you have chosen Custom Sort from the Sort column to define the sorting manner of groups for the selected group level.

Special Group
Specifies how to group your information. Activated only when you have chosen Special Group from the Special Function column to define a special group.

Select N
Opens the Select N dialog to specify the Select N condition.

Group Filter
Opens the Group Filter dialog to specify the group filter condition.

Summary screen

Specifies the fields on which to create aggregate functions.

Resources
Lists all the available data resources.

Adds the selected field as the summary field to the banded object.

Removes the selected summary field that is not required.

Moves the selected summary field one step up.

Moves the selected summary field one step down.

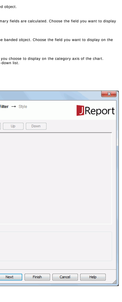
Summarized Fields
Lists all the fields that you want to display and to be summarized in the banded object.

Aggregate Function
Specifies the function to summarize the selected field.

Break Field
Displays a defined field on which the summary will be calculated. If a summary field is added below the Banded Object node, the break field is not and the summary will be calculated on the whole dataset.

Chart screen

Specifies to create a chart together with the banded object, which will be placed in the banded header panel. This screen is available only when there is at least one group by field and one summary field which is calculated on one group by field in the banded object.



No Chart
Specifies not to create a chart.

Bar Chart
Specifies to create a Clustered Bar 2-D chart together with the banded object.

Line Chart
Specifies to create a Line 2-D chart together with the banded object.

Pie Chart
Specifies to create a Clustered Pie chart together with the banded object.

Category
Lists the group by fields of the banded object on which the summary fields are calculated. Choose the field you want to display the category (X) axis of the chart from the drop-down list.

Series
Lists the fields that have been added as the group by fields of the banded object. Choose the field you want to display on the series (Z) axis of the chart from the drop-down list.

Show Values
Lists the summary fields which are calculated based on the field you choose to display on the category axis of the chart. Choose the value you want to display in the chart from the drop-down list.

Filter screen

Specifies to filter data displayed in the banded object.

For details about options in the screen, see Edit Filter dialog.

Style screen

Specifies the style of the banded object.

Style
Lists all the available styles that can be applied to the banded object.

Preview
Displays the selected layout and style effects.

Create a Banded Report
Creates a restricted version of a flow layout report, one that can contain a single horizontal banded object only. This layout option is intended to provide compatibility with Report Section 7 report templates.

Create a Pivot Layout Report Containing a Banded Object
Creates a report that contains a horizontal banded object mapped to the specified dataset. This layout option is specified by default.

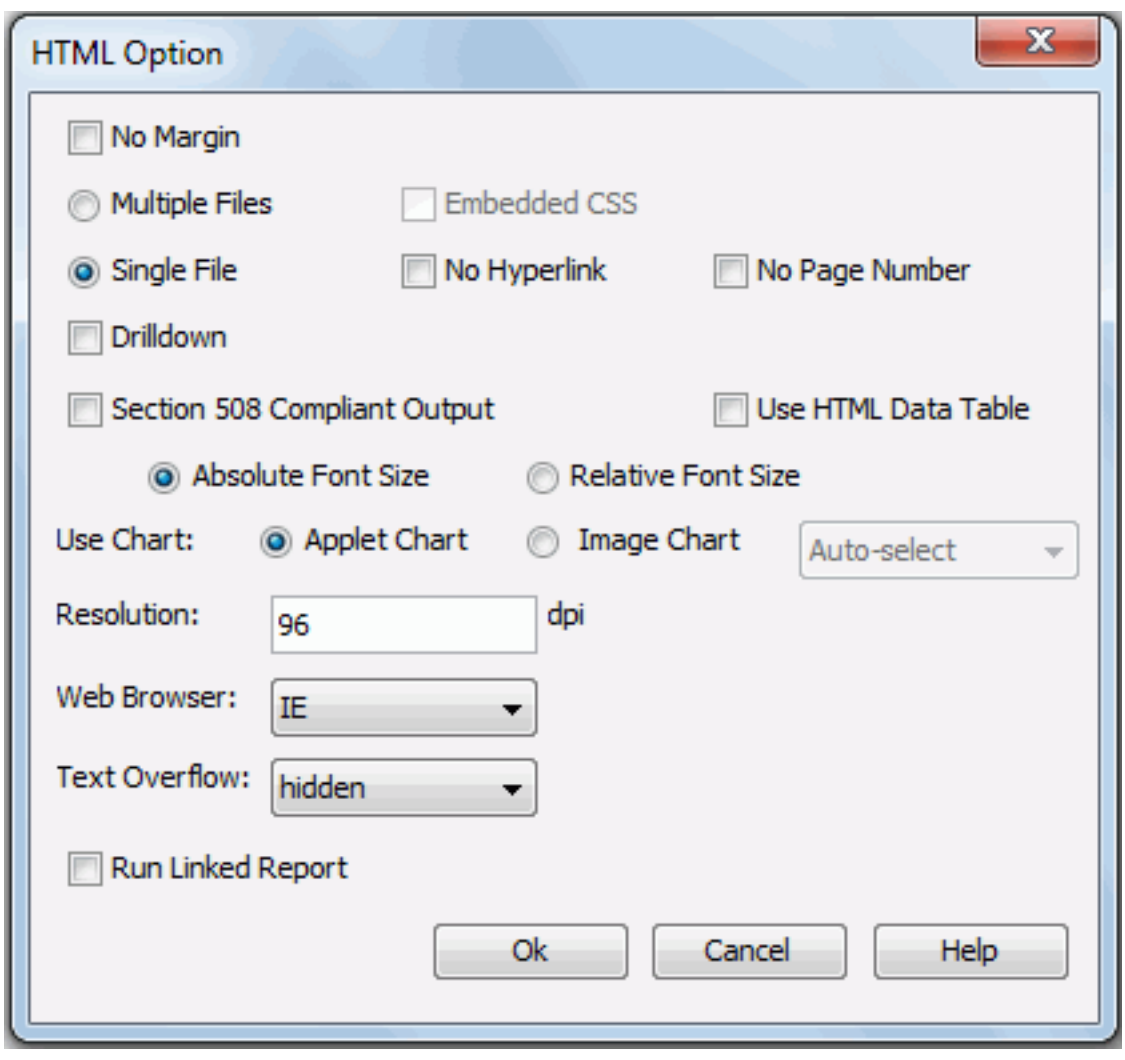
Page Setup
Opens the Page Setup dialog to specify page properties.

Related topics:

- Creating a banded report

HTML Option dialog

The dialog appears when you set the Default Format for Viewing Report property of a report tab or a web report to HTML and then click ... in the value cell in the Report Inspector. It helps you to specify the parameters for running a report in HTML format.



The following are details about options in the dialog:

No Margin

Removes the margins you originally set while designing the report.

Multiple Files

Generates the report result to multiple HTML files. JReport designates a serial number for each HTML page. For example, if you named a 3-page report as "sales", JReport will create three files called sales_1.html, sales_2.html, and sales_3.html.

- **Embedded CSS**
  Specifies to embed the cascading style sheet (CSS) in the HTML files; otherwise, the .css file will be generated individually.

Single File

Generates the report result to a single HTML file.

- **No Hyperlink**
  If checked, there will be no hyperlinks for navigating previous and next pages on the navigation bar of the HTML file.
- **No Page Number**
  If checked, there will be no page number information showing the current page number and total page number on the navigation bar of the HTML file.

Drilldown

Generates the report result into an HTML file with the Drilldown feature enabled. The Drilldown feature enables you to inspect certain items for further detailed data.

Section 508 Compliant Output

If checked, the accessibility attributes defined for the report elements via the Report Inspector will be exported to the HTML result which is Section 508 compliant. See Accessibility for more information.

When Section 508 Compliant Output is checked, the Use HTML Data Table and Relative Font Size options will be checked and disabled. The output will be Section 508 compliant including HTML data table, accessible attributes, and relative font feature.

Use HTML Data Table

Specifies whether the table and crosstab components will be output as table objects in the HTML result.

Absolute Font Size

Generates the report result using an absolute font size, which means that the font size is fixed, and cannot be adjusted according to the font size settings in the web browser.

Relative Font Size

Generates the report result using a relative font size. The font size can be adjusted according to the font size settings in the web browser.

Use Chart

- **Applet Chart**
  Uses a Java applet to display the charts in an HTML result file.
- **Image Chart**
  If selected, charts will be displayed as images. You can specify the image type from the drop-down list. The options are:
  - **Auto-select**
    If selected, the image format will be detected to JPG or GIF by the JReport system automatically. If the image colors are less than 256 colors, GIF will be applied; otherwise, it is JPG.
  - **GIF**
    If selected, the GIF format will be applied, which is a lossless compression technique and supports only 256 colors. GIF is better than JPG for images with only a few distinct colors, such as line drawings, black and white images and small text that is only a few pixels high.
  - **JPEG**
    If selected, the JPEG format will be applied, which is supported on the Web. JPEG is a lossy compression technique that is designed to compress color and grayscale continuous-tone images. JPEG images support 16 million colors and are best suited for photographs and complex graphics.
  - **PNG**
    If selected, the PNG format will be applied, which provides a portable, legally unencumbered, well-compressed (effectively 100 percent lossless compression), well-specified standard for lossless bitmapped image file. PNG supports indexed-color images of up to 256 colors and shows a more interchangeable, flexible and robust function than GIF.

Resolution

Specifies the resolution of the HTML result to zoom in/out, in DPI. The default value is obtained from the operation system, which is the resolution of your monitor, for example, 72 DPI on Unix or 96 on Windows. You can set higher/lower value to zoom in/out.

Web Browser

Specifies the web browser for which the HTML result adapts.

Text Overflow

Specifies whether the text overflow is visible or hidden.

Run Linked Report

If the report is linked with other reports, you can check whether or not to generate the linked reports in the HTML result. If you are only interested in the primary report, leave this option unchecked. Generating linked reports at the same time, especially when the linked reports contain a large amount of data, will cause performance issue.

OK

Applies the changes and closes the dialog.

Cancel

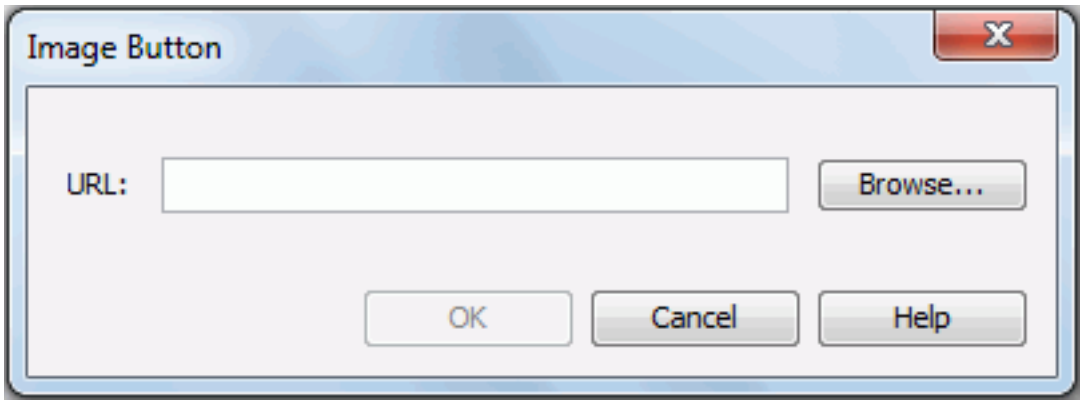
Cancels the changes and exits the dialog.

Help

Displays the help document about this feature.

Image Button dialog

This dialog appears when you right-click a navigation item and select Button Type > Image Button from the shortcut menu. It helps you to specify the source of the image button.



URL

Specifies the source of the image button from a URL. You can click the Browse button to specify the path or input the URL directly in the text field.

OK

Applies the changes and closes the dialog.

Cancel

Does not retain any changes and closes the dialog.

Help

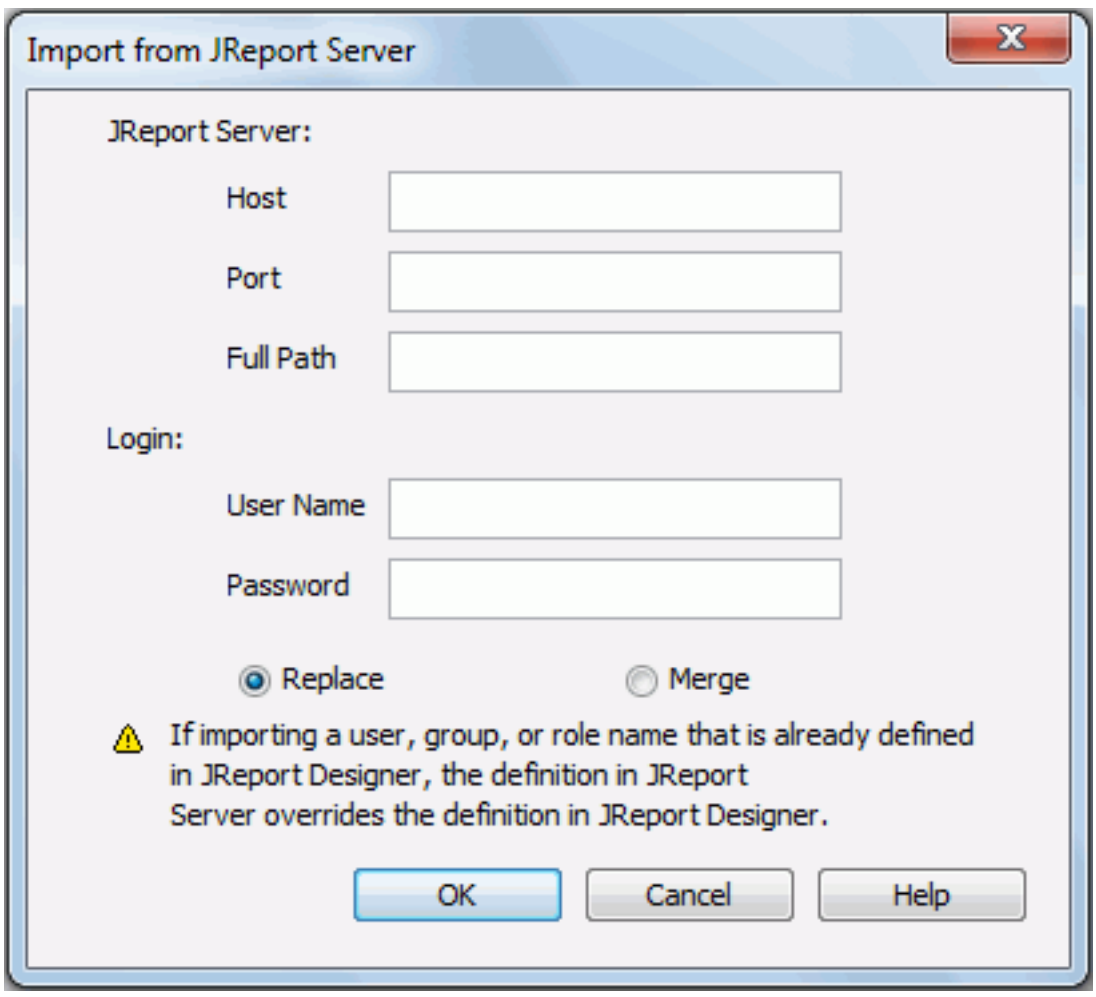
Displays the help document about this feature.

Related topics:

- Using navigation control to undo/redo value selection in filter controls

Import from JReport Server dialog

This dialog appears when you click Security > Import from JReport Server in the Security dialog. It helps you to import users, groups and roles from JReport Server.



The following are details about options in the dialog:

Host

Specifies the host of the remote web server.

Port

Specifies the port that the web server listens to.

Full path

It is /jrserver, which is used in the URL to access the servlet.

User Name

Specifies the user name for JReport Server.

Password

Specifies the password for the user name.

Replace

When there are users, groups and roles on the server with the same names as those on the local computer, the setting from the server will replace those on the local computer. If the names of the user/role are the same on both the Server and the local computer, you can use the server users/roles to override any user defined ones.

Note: Permission settings for user defined users or roles will be lost once they are replaced.

Merge

When there are users, groups and roles on the server with the same names as those defined on the local computer, the settings from the server will be maintained while being integrated with the permissions from the local computer.

OK

Imports users, groups and roles from JReport Server and closes the dialog.

Cancel

Does not retain the changes and closes the dialog.

Help

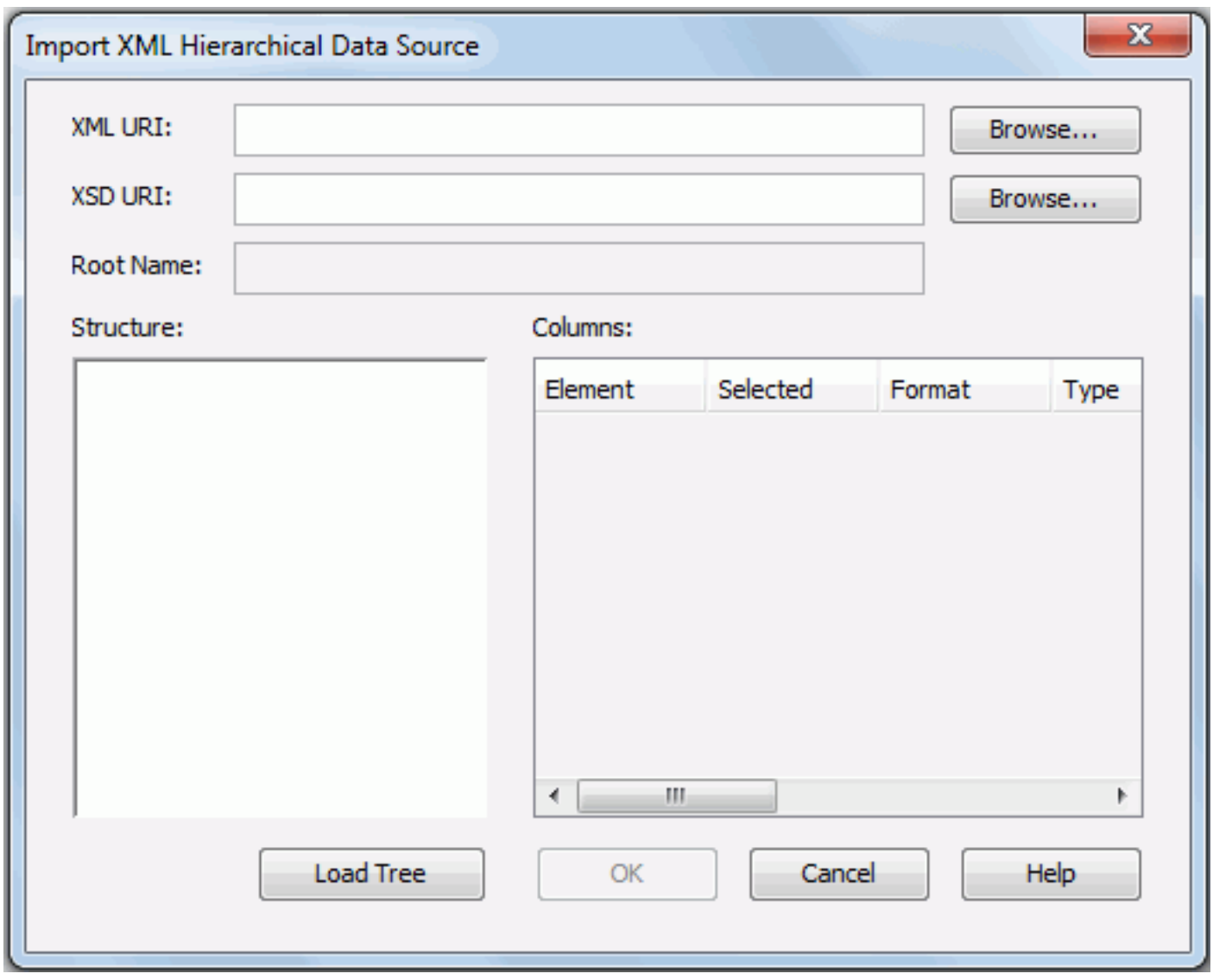
Displays the help document about this feature.

Related topics:

- Importing security information from JReport Server

Import XML Hierarchical Data Source dialog

This dialog appears when you right-click a data source and select Add XML Hierarchical Data Source from the shortcut menu in the Catalog Browser. It helps you to import an XML HDS into a catalog.



The following are details about options in the dialog:

XML URI

Specifies the location of the external XML format data source. Can be a local path or a web URL. Click Browse to find the file on the local disk.

XSD URI

Specifies the location of the schema file for the XML format data source. Can be a local path or a web URL. Click Browse to find the file on the local disk.

Root Name

Indicates the hierarchical data source root, starting from which JReport Designer imports the data.

Structure

Shows the structure of the hierarchical data source.

Columns

Specifies settings for the columns.

- **Element**
  Displays the field name in the selected node of table.
- **Selected**
  Specifies whether to import the field into the catalog.
- **Format**
  Specifies the format in which the field value is to be displayed in the report result. Varies for different types of data, and can be manually defined.
- **Type**
  Specifies the data type of the column.
- **Precision**
  Displays the default value for each SQL type automatically. Click the cell and modify the value if necessary.
- **Scale**
  Specifies the decimal place.
- **Nullable**
  Specifies whether the field is nullable. You have three options to choose: Nullable, No Nulls and Nullable Unknown.
- **Currency**
  Specifies whether to use the Currency type for this field or not.
- **Array**
  Specifies whether to use the Array type for this field or not.

Load Tree

Analyzes and loads data tree from the external data source file. The structure is shown in the Structure box after you click this button.

OK

Applies all changes here and closes the dialog.

Cancel

Does not retain changes and closes the dialog.

Help

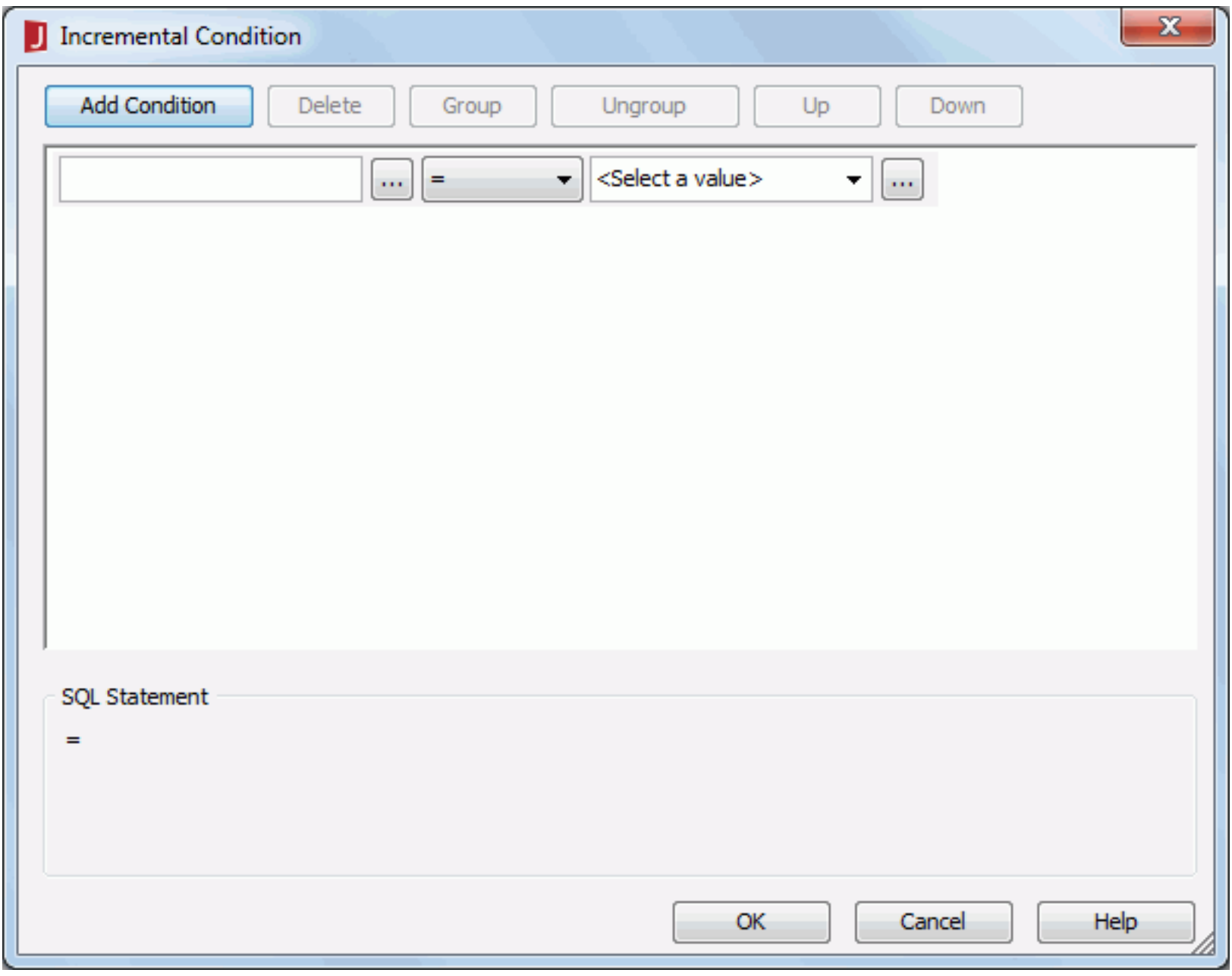
Displays the help document about this feature.

Related topics:

- Adding an HDS to a catalog

Incremental Condition dialog

This dialog appears when you click ... in the value cell of the Incremental Fetch property of a refresh object in a library component in the Report Inspector.
It helps you to specify conditions to filter the incremental data.



The following are details about options in the dialog:

Add Condition

Adds a condition line.

Delete

Deletes the selected condition line.

Group

Makes the selected conditions in one group.

Ungroup

Makes the selected condition ungrouped.

Up

Moves the selected condition or group up to a higher level.

Down

Moves the selected condition or group down to a lower level.

Field

Specifies the business view element on which the filter condition is based. Click ... next to the text box to select the business view element.

Operator

Specifies the operator to compose the condition expression.

Value

Specifies the value of how to filter the business view element. You can type in the value manually, click ... next to the text box to specify the value, or select a value from the drop-down list.

- **MaxExistValue**
  If selected, the maximum data value in the selected business view element from the last refresh time will be used as the value to filter the business view element.
- **MinExistValue**
  If selected, the minimum data value in the selected business view element from the last refresh time will be used as the value to filter the business view element.
- **LastRefreshTime**
  If selected, the previous refresh time will be used as the value to filter the business view element.

SQL

Statement Displays the SQL statement of the condition expression.

OK

Applies the changes and closes this dialog.

Cancel

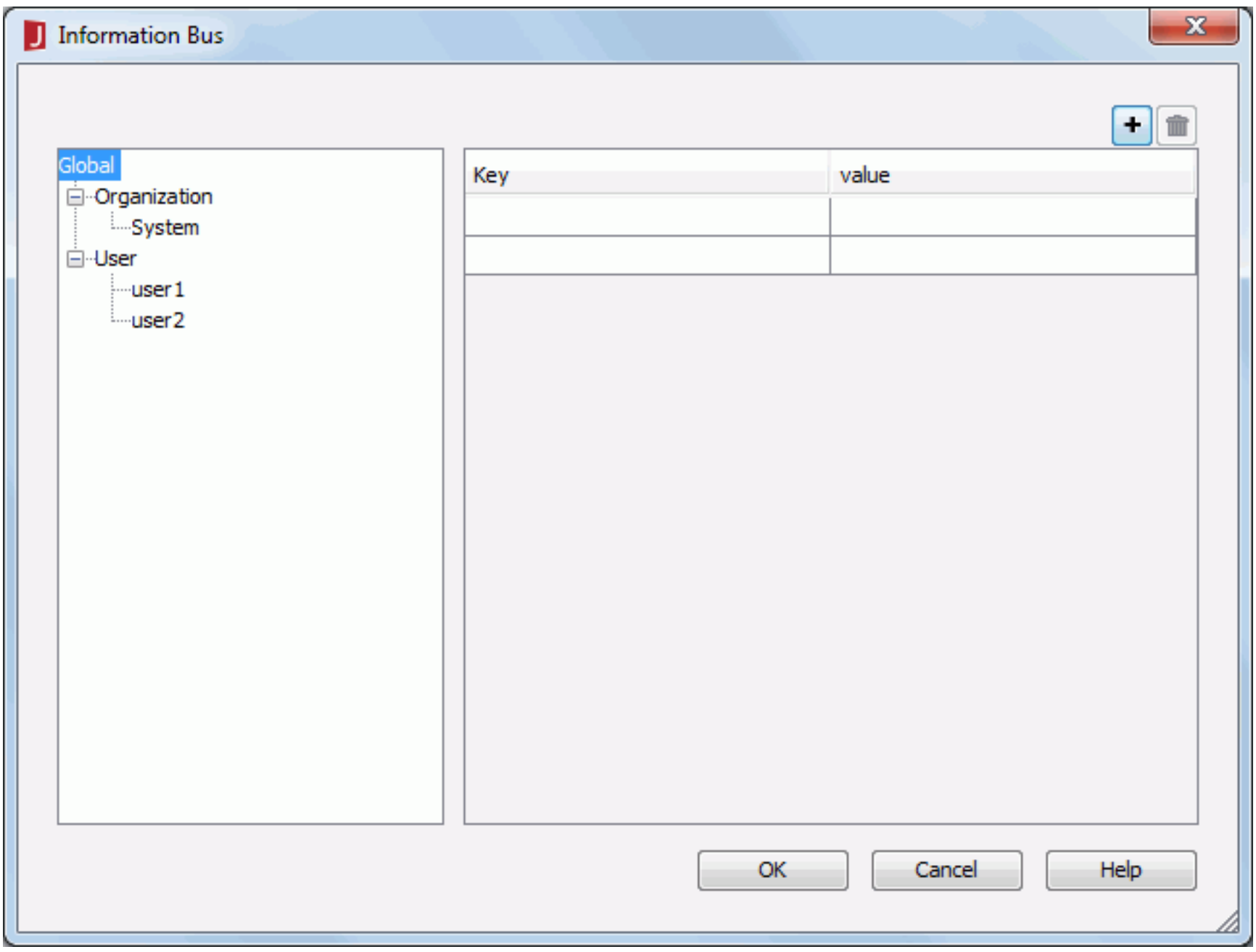
Cancels the changes and exits the dialog.

Help

Displays the help document about this feature.

Information Bus dialog

This dialog appears when you click File > Information Bus on the menu bar. It helps you to edit the Information Bus content saved with user information in the currently opened catalog.



The following are details about options in the dialog:

Users

Shows a tree structure according to the user definition in the currently opened catalog. The users here are the same as the users created for member-level security.

The information from global level, organization level and user level are listed here. System admin can get or put information in all information containers. Organization admin can get or put information in the global level information container, its own organization level information container and the user level information containers that belong to its organization. Organization user can get or put information in the global level information container, the organization level information container it belongs and its own user level information container.

Adds a new information line to the container.

Removes the selected information line.

Key

Shows the information key of the selected user.

Value

Shows the information string of the selected user.

OK

Applies the changes and closes this dialog.

Cancel

Cancels the changes and exits the dialog.

Help

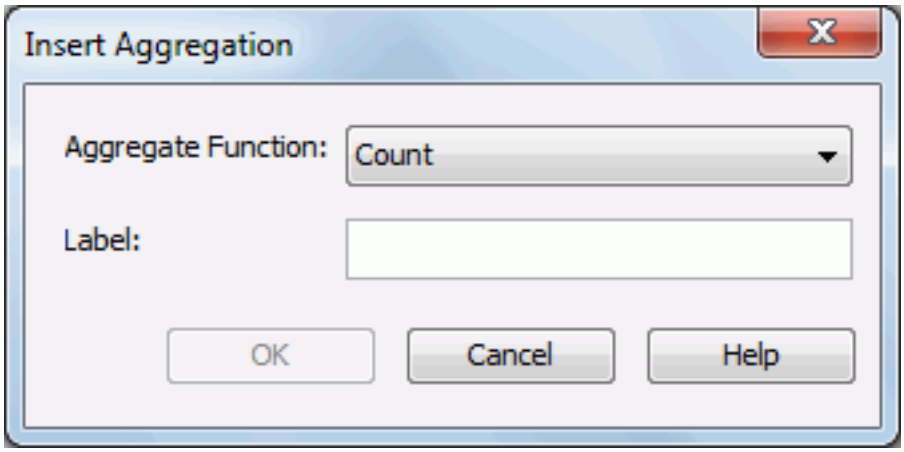
Displays the help document about this feature.

Related topics:

- Information Bus

Insert Aggregation dialog

The dialog appears when you drag a field that can be used as aggregate field from the Resource View panel to the cell above/under any subtotal or grand total of a crosstab. It helps you to insert the selected field as an aggregate field in the crosstab.



Aggregate Function

Specifies the function to summarize the selected field. For details about each function, see Math functions.

If the selected field has been predefined with an aggregate function, the option is Aggregation. The name of the field is displayed in the text box and cannot be edited.

Label

Specifies the display name of the aggregate field, which will label the newly created summaries in the crosstab. You can leave it blank.

OK

Inserts the aggregate field and closes the dialog.

Cancel

Does not retain any changes and closes the dialog.

Help

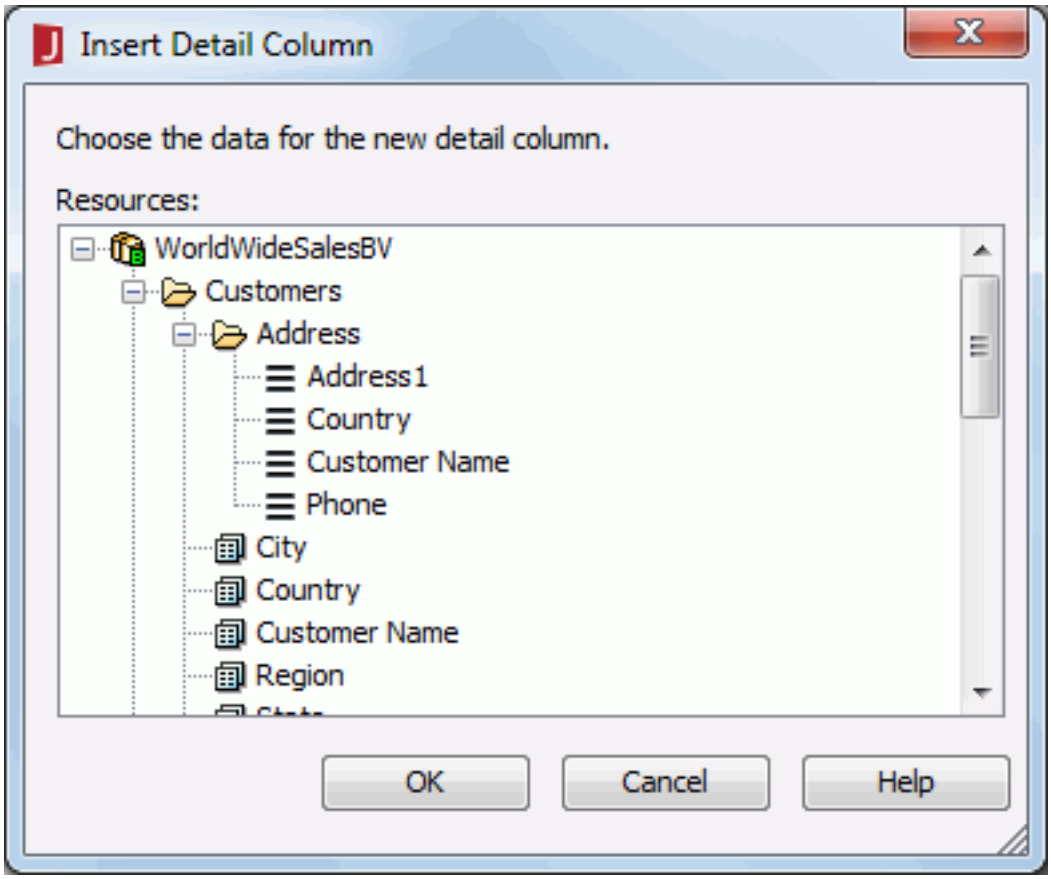
Displays the help document about this feature.

Related topics:

- Managing the aggregate fields in a crosstab

Insert Detail Column dialog

The dialog appears when you right-click a table or a table column and select Insert > Detail Column from the shortcut menu. It helps you to insert new detail columns into a table.



Resources

Lists all the available data resources for the new detail column. Select the resource you want from the list.

OK

Inserts the column with the specified resource and closes the dialog.

Cancel

Does not retain any changes and closes the dialog.

Help

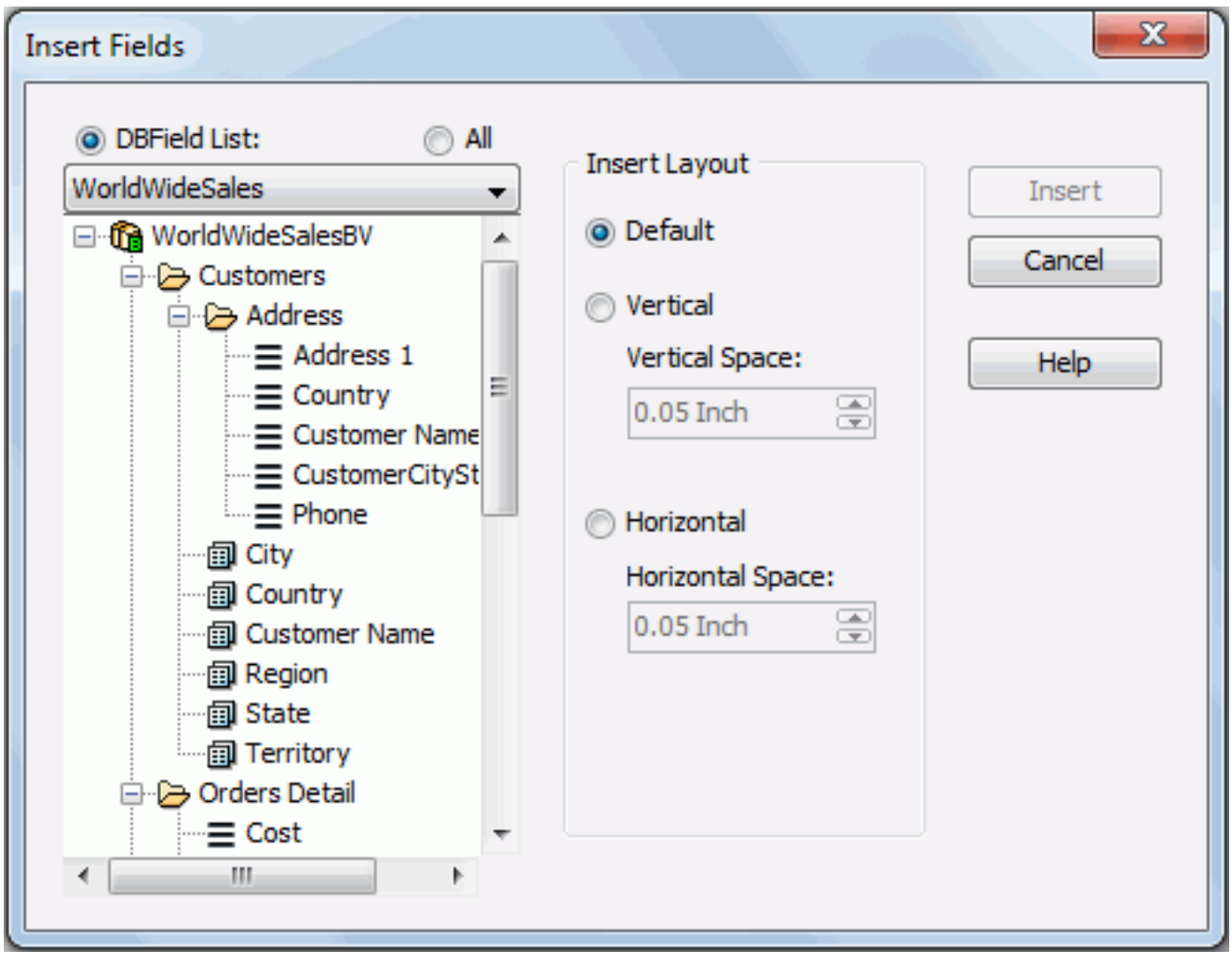
Displays the help document about this feature.

Related topics:

- Managing columns/rows in a table

Insert Fields dialog

This dialog appears when you click Insert > DBField/Formula/Parameter on the menu bar, or the corresponding buttons on the Insert toolbar, or the button in the Value column of the Display Type dialog when the display type is defined as List or Drop-down List, or the button in the Value column of the Web Options dialog when defining the web options of a list or drop-down list in the configuration panel of a library component. It helps you to insert fields into a report, or to add fields to control values of the list/drop-down list item.



The following are details about options in the dialog.

DBField List

Specifies to insert a field.

- **Dataset**
  This node is automatically expanded when inserting a DBField. It lists the tables and fields from the current dataset. If you want to insert DBField from another dataset, select the dataset from the dataset drop-down list. You can also click the <Choose Data From...> item from the dataset drop-down list to create a new dataset.

- **Formulas**
  This node is automatically expanded when inserting a formula field. It lists all the formulas in the current catalog. You can also click the <Add Formula...> item to create one.

- **Parameters**
  This node is automatically expanded when inserting a parameter field. It lists all the parameters in the current catalog. You can also click the <Add Parameter...> item to create one.

- **Dynamic Resources**
  This node is available only when you are defining values for a list or drop-down list in a library component. It lists all the dynamic formulas in the current library component. You can also click the <Add Formula...> item to create one.

All

This radio button is available only when the dialog is opened from the Display Type dialog or Web Options dialog. You can check it to add an All value to the list/drop-down list.

Insert Layout

When multiple fields are selected, you can arrange these objects horizontally or vertically by selecting Horizontal or Vertical, or by selecting the Default radio button to set it with the default layout.

- **Vertical Space**
  Specifies the space between fields that are arranged vertically, in inches.

- **Horizontal Space**
  Specifies the space between fields that are arranged horizontally, in inches.

Insert

Inserts the selected fields into the specified location in the report.

Cancel

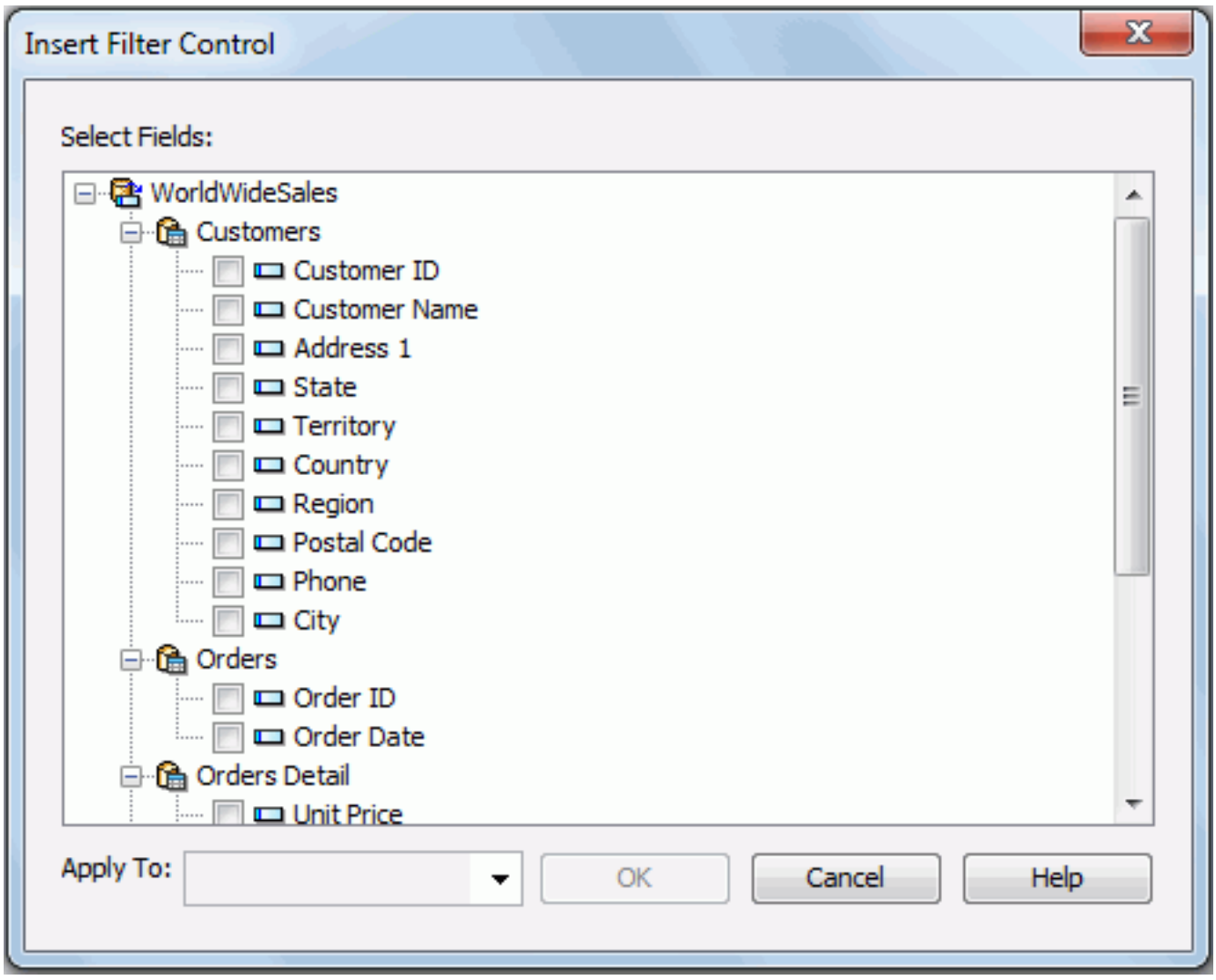
Does not retain any changes and closes the dialog.

Help

Displays the help document about this feature.

Insert Filter Control dialog

The dialog appears when you click Insert > Web Controls > Filter Control on the menu bar, drag Filter Control from the Toolbox panel, or click the down arrow on the Insert Control Objects button and select Filter Control from the drop-down list. It helps you to insert a filter control into a report for filtering component data.



The following are details about options in the dialog:

Select Fields

Specifies the fields to bind to the filter control. All the selected fields should be of the same data type. All uncomparable data type fields cannot be bound to filter controls, such as Binary, Blob, Clob, Longvarchar, Longvarbinary, and Varbinary.

The common usage is to select one field, and then based on the field to filter the data of the components created from the same data source as the field.

To filter components using different data sources, choose a common field all the data sources contain and select the field in all the data sources.

Apply To

Specifies the target data components you want to apply the filter to. The drop-down list includes all the data components in the current report that are based on the same data sources the selected fields are in.

OK

Inserts the filter control into the report and closes the dialog.

Cancel

Cancels the operation and closes the dialog.

Help

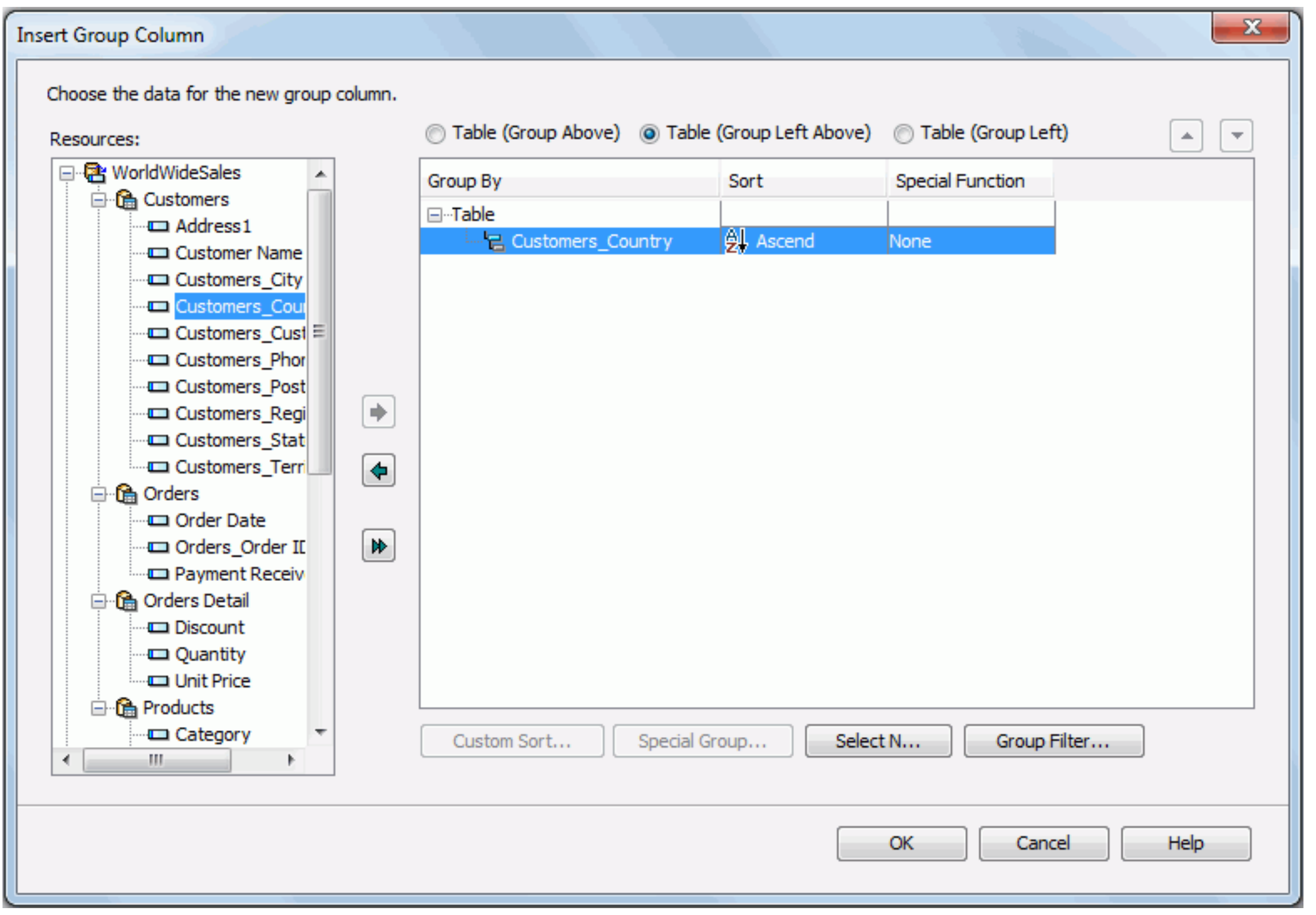
Displays the help document about this feature.

Related topics:

- Using slider or filter control to filter data

Insert Group Column dialog

The dialog appears when you right-click a table or a table column and select Insert > Group Column from the shortcut menu. It helps you to insert new group columns into a table.



The followings are details about options of this dialog:

Resources

Lists all the available data resources for the new group columns.

Adds the selected field as the group by field in the table.

Removes the selected group by field that is not required.

Replaces the selected group by field in the table with the specified field in the Resources box.

Group By

Lists the fields that are used to group data in the table.

Sort

Specifies how groups at the specific group level will be sorted.

- **Ascend**
  Groups will be sorted in an ascending order (A, B, C).
- **Descend**
  Groups will be sorted in a descending order (C, B, A).
- **No Sort**
  Groups will be sorted in the original order in database.
- **Special Group**
  Opens the User Defined Group dialog to define grouping information.

  For example, if you placed a field named Region for grouping, and this field contains all 50 states of the United States; and if you want to see the data between Maryland (MD) and New York (NY), you can define the criteria by selecting the between operator to further define your grouping information.
- **Custom Sort**
  Opens the Custom Sort dialog to set how groups will be sorted.

Special Function

If the group by field is of Numeric/String/Date/Time type, you can select a special function for the field in the Special Function column to further specify to which level the data will be grouped by.

If Customize is selected, the Customized Function dialog will be displayed, in which you can set the function by your own.

Group Above

Specifies to place the selected group by field above the detail panel.

Group Left Above

Specifies to place the selected group by field left above the detail panel.

Group Left

Specifies to place the selected group by field left to the detail panel.

Moves the specified group one step up.

Moves the specified group one step down.

Custom Sort

Specifies how to sort the groups. Activated only when you have clicked Custom Sort from the Sort column to define the sorting manner of groups for the selected group level.

Special Group

Specifies how to group your information. Activated only when you have clicked Special Group from the Sort column to define a special group.

Select N

Opens the Select N dialog to specify the Select N condition.

Group Filter

Opens the Group Filter dialog to specify the group filter condition.

OK

Applies the changes and closes the dialog.

Cancel

Does not retain any changes and closes the dialog.

Help

Displays the help document about this feature.

Related topics:

- Managing columns/rows in a table

Insert Link dialog for page report

The dialog appears when you select a specified object in the design area in a page report, right-click it and select Link from the shortcut menu, or click the button in the value cell of the Link property or Detail Report property in the Report Inspector. It helps you to link the object to a report, a detail report, a website, an e-mail address or a Blob data type field as required.

Conditional Link

Specifies whether it is a conditional link. If checked, the object can be linked to different targets based on different conditions.

- Adds a new condition in the Edit Conditions dialog.
- Edits the selected condition.
- Removes the selected condition.
- Moves the selected condition one step up.
- Moves the selected condition one step down.
- **Others**
  If checked, you can define to which target the object will be linked when none of the conditions you have specified is met.

Link Type

Specifies the type of the link target for the object if it is not a conditional link or for the object under the selected condition if it is a conditional link. It can be one of the following:

- Report
- Master/Detail Report
- URL
- E-mail
- Content

More

Displays more link options. Available only for the Report, Master/Detail Report and Content link types.

OK

Accepts the linking result and closes the dialog.

Cancel

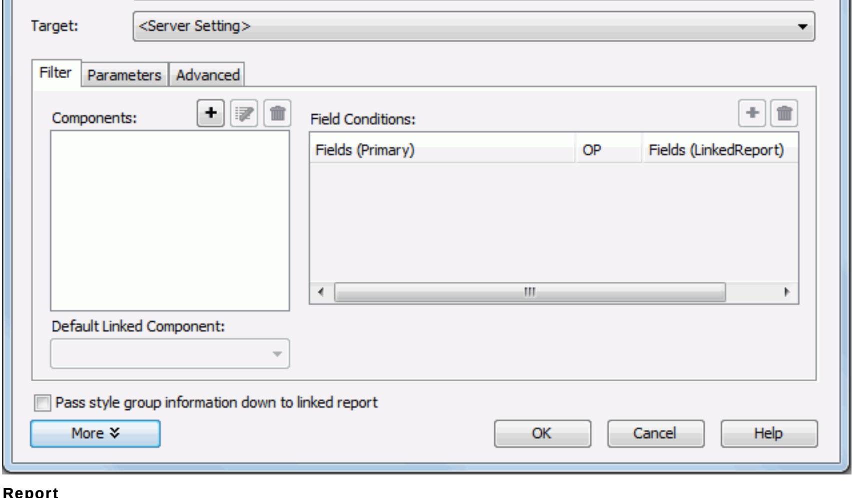
Cancels the linking process and exits the dialog.

Help

Displays the help document about this feature.

Report

It helps you to link the object to a report.



Report

Specifies the report in which the linked report tab is. Click the Browse button to select the desired report in the Select a Report dialog.

Report Tab

Specifies the linked report tab.

Target

Specifies the window or frame in which to load the linked report. Applied only when the link is triggered in Page Report Studio.

- **<Server Setting>**
  Loads the linked report according to setting of the Pop Up New Window for Links option in the Profile > Configure Profile > Page Report Studio > Properties > Default tab on JReport Server.
- **Same Frame**
  Loads the linked report into the same frame as the primary report.
- **Whole Window**
  Loads the linked report into the full browser window.
- **New Window**
  Loads the linked report into a new window. This window is not named.
- **Parent Frame**
  Loads the linked report into the parent frame of the frame in which the primary report is.
- **Other Frame**
  Loads the linked report into some other specified frame. If the frame name does not exist, the linked report will be loaded into a new window.

Filter tab

Specifies the filter conditions based on which the link relationship between the primary report and the linked report is set up.

- **Components**
  Specifies the components in the linked report that will be interlinked with the primary report.
  - Adds a component in the linked report to be interlinked with the primary report.
  - Edits the component in the linked report to be interlinked with the primary report.
  - Removes the selected component from the Component in Report Tab box.
- **Default Linked Component**
  Specifies the default linked component in the linked report.
- **Field Conditions**
  Lists the filter conditions between the primary report and the linked report.
  - **Fields(Primary)**
    Lists the selected fields of the primary report.
  - **OP**
    Specifies the operator to compose the condition. It can be "=", "<>", "<", ">", "<=", ">=" or "IN".
  - **Fields(LinkedReport)**
    Lists the DBFields in the datasets of the linked report which are of the same data type as the selected fields in the primary report.
  - Adds a new condition line.
  - Removes the selected condition.

Parameters tab

This tab is available when the linked report uses parameters. It helps you to assign values to the parameters of the linked report automatically.

- **Parameters**
  Lists all parameters contained in the datasets of the linked report.
- **Value**
  Lists all the DBFields, formulas, summaries and parameters in the datasets of the primary report which are of the same data type as the parameters of the linked report. Select the field the value of which you want to assign to parameter of the linked report from the drop-down list.

Advanced tab

Specifies to pass the values of the filter objects such as filter controls and sliders in the primary report to those in the linked report.

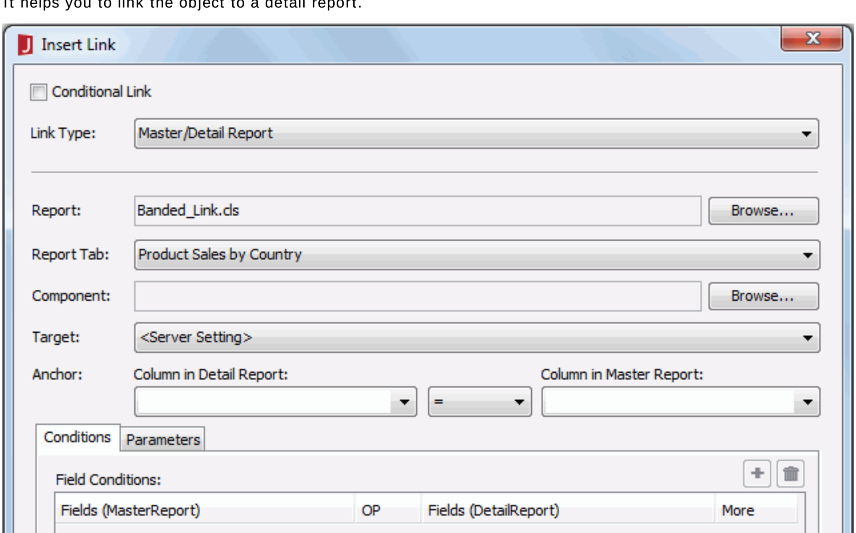
- **Primary Report Property/Object**
  Displays the filter objects in the primary report the values of which will be passed to those in the linked report. Click to select a filter object in the primary report in the Select Value dialog.
- **Linked Report Property/Object**
  Displays the filter objects in the linked report which will adopt the values passed from filter objects in the primary report. Click to select a filter object in the linked report in the Select Value dialog. You need to make sure that values of the filter object in the primary report can be applied to the filter object in the linked report in the same line.
- Adds a new line to pass values between the filter objects in the primary report and the linked report.
- Removes the selected line.

Pass style group information down to linked report

Specifies whether or not to apply the style group of the primary report to the linked report.

Master/Detail Report

It helps you to link the object to a detail report.



Report

Specifies the report in which the detail report tab is. Click the Browse button to select the desired report in the Select a Report dialog.

Report Tab

Specifies the target detail report tab.

Component

Specifies the component in the detail report tab that will be interlinked with the master report. Click the Browse button to add the components. You can add more than one component as long as the components are based on the same dataset.

Target

Specifies the window or frame where to load the detail report.

- **<Server Setting>**
  Loads the detail report according to setting of the Pop up New Window for Links option in the Profile > Configure Profile > Page Report Studio > Properties > Default tab on JReport Server.
- **Same Frame**
  Loads the detail report into the same frame as the master report.
- **Whole Window**
  Loads the detail report into the full browser window.
- **New Window**
  Loads the detail report into a new window, which is not named. If this is selected, the detail report will be opened independently when you click the trigger in Page Report Studio, that is to say the join and filter conditions you set between the master report and detail report will not be applied.
- **Parent Frame**
  Loads the detail report into the parent frame of the frame in which the master report is.
- **Other Frame**
  Loads the detail report into some other specified frame. If the frame name does not exist, the detail report will be loaded into a new window.

Anchor

Specifies the relationships between the master report and the detail report.

- **Column in Detail Report**
  Specifies the field in the detail report which will be used to link the detail report with the master.
- **OP**
  Specifies the operator. It can be "=", "<>", "<", ">", "<="or ">=".
- **Column in Master Report**
  Specifies the field in the master report.

Conditions tab

Specifies the filter conditions between the master report and the detail report.

- **Field Conditions**
  Lists all the filter conditions.
  - **Fields(MasterReport)**
    Lists the selected fields of the master report.
  - **OP**
    Specifies the operator to compose the condition. It can be "=", "<>", "<", ">", "<="or ">=".
  - **Fields(DetailReport)**
    Lists the DBFields in the datasets of the detail report which are of the same data type as the selected fields in the master report.
  - **More**
    Specifies relations between two conditions. Can be AND or OR.
  - Adds a new condition line.
  - Removes the selected condition.

Parameters tab

This tab is only available when there is at least one parameter applied in the filters of the queries that the detail report uses. It helps you to assign values to the parameters of the detail report automatically.

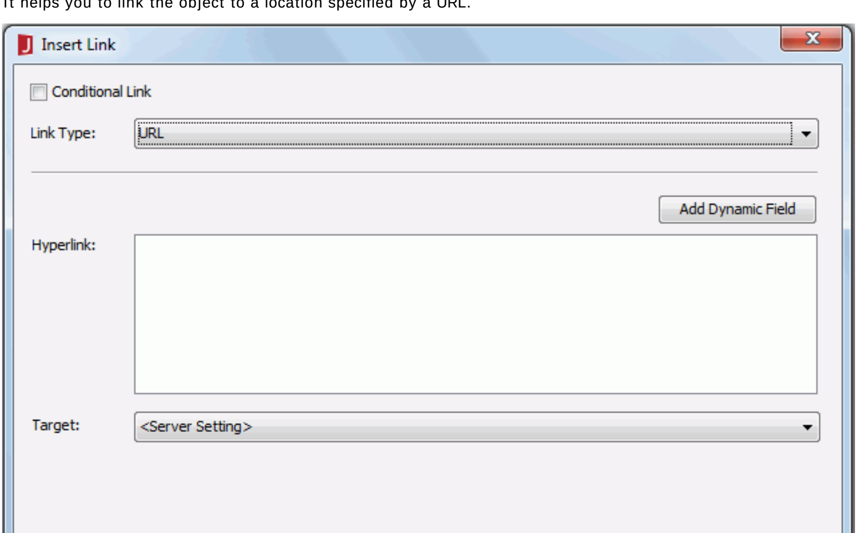
- **Parameters (Detail Report)**
  Lists all parameters contained in the datasets of the detail report.
- **Column (Master Report)**
  Lists all the DBFields, formulas and summaries in the datasets of the master report which are of the same data type as the parameters of the detail report. Select the field the value of which you want to assign to parameter of the detail report from the drop-down list.

Use the same encoding and DB setting for the detail report as that of the master report

Specifies whether the detail report uses the same encoding and DB settings as per the master report.

URL

It helps you to link the object to a location specified by a URL.



Hyperlink

Specifies the URL of the location that the object will be linked to. Fields can work in the hyperlink only when they are inserted via the Add Dynamic Field option.

- **Add Dynamic Field**
  Opens the Select Field dialog to insert a field into the URL.

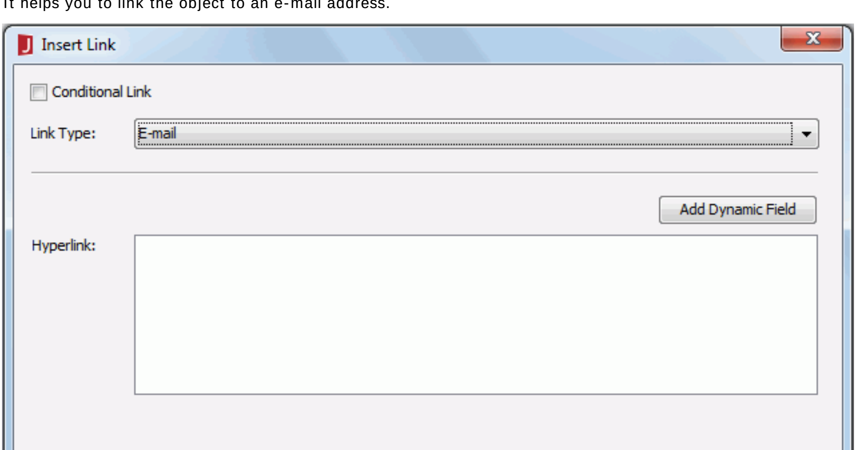
Target

Specifies the window or frame in which to load the location specified by the URL. Applied only when the link is triggered in Page Report Studio.

- **<Server Setting>**
  Loads the linked file according to setting of the Pop Up New Window for Links option in the Profile > Configure Profile > Page Report Studio > Properties > Default tab on JReport Server.
- **Same Frame**
  Loads the linked file into the same frame as the link.
- **Whole Window**
  Loads the linked file into the full browser window.
- **New Window**
  Loads the linked file into a new window. This window is not named.
- **Parent Frame**
  Loads the linked file into the parent frame of the frame that contains the link.
- **Other Frame**
  Loads the linked file into some other specified frame. If the frame name does not exist, the linked file will be loaded into a new window.

E-mail

It helps you to link the object to an e-mail address.



Hyperlink

Specifies the e-mail address that the object will be linked to. Then after you click the trigger object, an e-mail will be popped with the information specified in the Hyperlink box. You can then further customize the e-mail and send it.

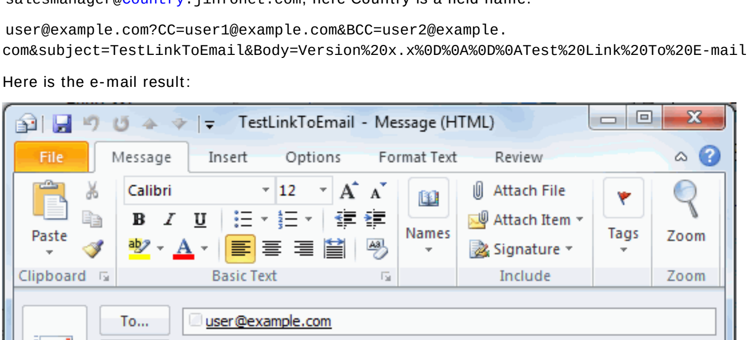
Enter the e-mail address in the Hyperlink box using the syntax `sAddress[sHeaders]`. The following are details about the syntax:

- **sAddress**
  One or more valid e-mail addresses separated by a semicolon. You must use Internet-safe characters, such as %20 for the space character.
- **sHeaders**
  Optional. One or more name-value pairs. The first pair should be prefixed by a "?" and any additional pairs should be prefixed by a "&". The name can be one of the following strings:
  - **CC**
    Addresses to be included in the "cc" (carbon copy) section of the message.
  - **BCC**
    Addresses to be included in the "bcc" (blind carbon copy) section of the message.
  - **subject**
    Text to appear in the subject line of the message.
  - **body**
    Text to appear in the body of the message.

Hyperlink examples:

- `user@example.com`
- `user1@example.com;user2@example.com`
- `Customer Name@jinfonet.com`, here Customer Name is a field name.
- `salesmanager@Country.jinfonet.com`, here Country is a field name.
- `user@example.com?CC=user1@example.com&BCC=user2@example.com&subject=TestLinkToEmail&Body=Version%20x.x%0D%0A%0D%0ATest%20Link%20To%20E-mail`

  Here is the e-mail result:



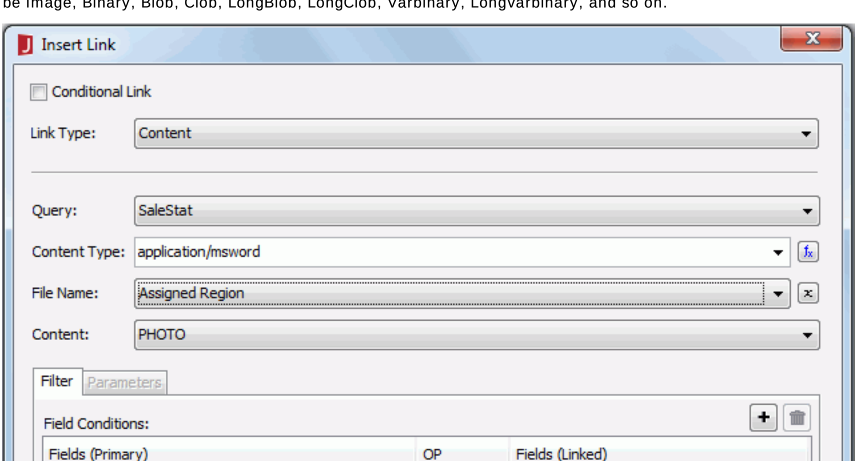
Note: Fields can work in the hyperlink only when they are inserted via the Add Dynamic Field option.

Add Dynamic Field

Opens the Select Field dialog to insert a field into the e-mail address.

Content

It helps you to link the object to a Blob data type field. The Blob data type field will be bound to the object and displayed as a hyperlink. You can download the Blob content by clicking this hyperlink. In JReport, Blob could be image, Binary, Blob, Clob, LongBlob, LongClob, Varbinary, Longvarbinary, and so on.



Query

Specifies the query which contains the required Blob data type field. Only queries from the same data source connection as the current report are listed.

When you set the link type to Content, JReport will check whether there are such kind of queries available. If not, options for this link type will be disabled.

Content Type

Specifies the content type for the Blob type data. You can click the button to bind it with a field column or a record level pass one formula in the selected query.

File Name

Specifies the file name for the linked Blob type data. You can click the button to bind it with a field column or a record level pass one formula in the selected query.

Content

Specifies the Blob data type field in the selected query.

Filter tab

Specifies the link conditions between the query used by the current report and the query that contains the linked Blob content.

- **Field Conditions**
  Lists the filter conditions between the two queries.
  - **Fields(Primary)**
    Lists the selected fields of the query used by the current report.
  - **OP**
    Specifies the operator to compose the condition. It can be "=", "<>", "<", ">", "<=", ">=" or "IN".
  - **Fields(Linked)**
    Lists the DBFields in the datasets of the query that contains the linked Blob content, which are of the same data type as the selected fields in the query used by the current report.
  - Adds a new condition line.
  - Removes the selected condition.

Parameters tab

This tab is available when the query that contains the linked Blob content uses parameters. It helps you to assign values to the parameters automatically.

- **Parameters**
  Lists all the parameters the query that contains the linked Blob content uses.
- **Value**
  Lists all the DBFields, formulas, summaries and parameters in the query used by the current report, which are of the same data type as the parameters the query that contains the linked Blob content uses. Select the field the value of which you want to assign to a parameter from the drop-down list.

Related topic:

- Binding a link to a label

Insert Link dialog for web report/library component

The dialog appears when you select a specified object in the design area in a web report or library component, right-click it and select Link from the shortcut menu, or click the button in the value cell of the Link property in the Report Inspector. It helps you to link the object to a report, a website, an e-mail address or a Blob data type field as required.

Conditional Link

Specifies whether it is a conditional link. If checked, the object can be linked to different targets based on different conditions.

- Adds a new condition in the Edit Conditions dialog.
- Edits the selected condition.
- Removes the selected condition.
- Moves the selected condition one step up.
- Moves the selected condition one step down.
- **Others**
  If checked, you can define to which target the object will be linked when none of the conditions you have specified is met.

Link Type

Specifies the type of the link target for the object if it is not a conditional link or for the object under the selected condition if it is a conditional link. It can be one of the following:

- Report
- URL
- E-mail
- Content

More

Displays more link options. Available only for the Report, Master/Detail Report and Content link types.

OK

Accepts the linking result and closes the dialog.

Cancel

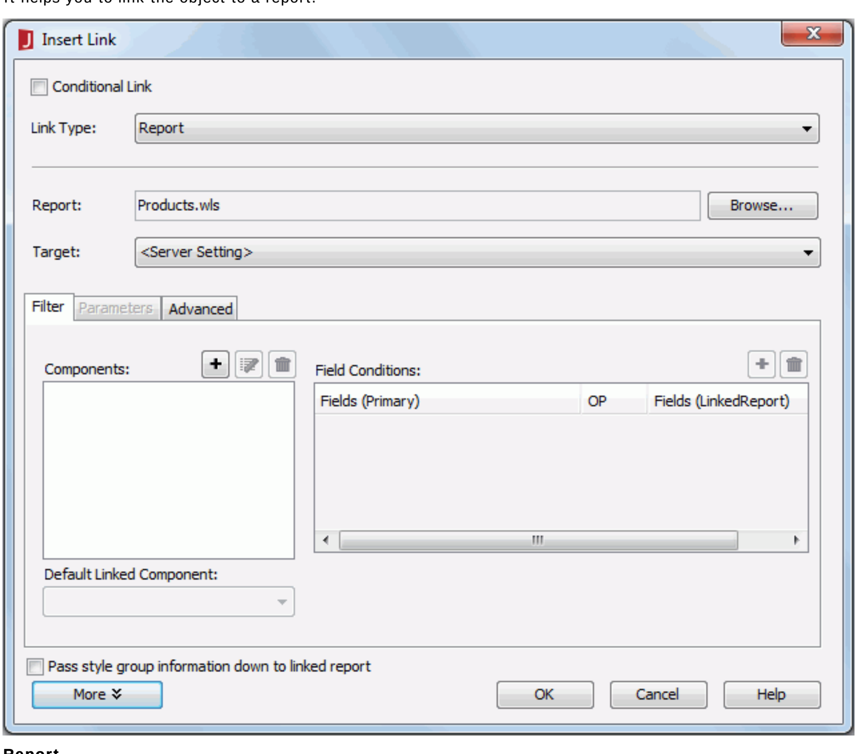
Cancels the linking process and exits the dialog.

Help

Displays the help document about this feature.

Report

It helps you to link the object to a report.



Report

Specifies the target report. Click the Browse button to select the desired report in the Select a Report dialog.

Target

Specifies the window or frame in which to load the linked report. Applied only when the link is triggered Web Report Studio; when triggered in JDashboard, except for Same Frame, all the other targets are treated as New Window.

- **<Server Setting>**
  Loads the linked report according to setting of the Target of Detail Table Report and Links option in the Profile > Configure Profile > Page Report Studio > Properties > Default tab on JReport Server.
- **Same Frame**
  Loads the linked report into the same frame as the primary report.
- **Whole Window**
  Loads the linked report into the full browser window.
- **New Window**
  Loads the linked report into a new window. This window is not named.
- **Parent Frame**
  Loads the linked report into the parent frame of the frame in which the primary report is.
- **Other Frame**
  Loads the linked report into some other specified frame. If the frame name does not exist, the linked report will be loaded into a new window.

Filter tab

Specifies the filter conditions based on which the link relationship between the primary report and the linked report is set up.

- **Components**
  Specifies the components in the linked report that will be interlinked with the primary report.
  - Adds a component in the linked report to be interlinked with the primary report.
  - Edits the component in the linked report to be interlinked with the primary report.
  - Removes the selected component from the Component in Report Tab box.
- **Default Linked Component**
  Specifies the default linked component in the linked report.
- **Field Conditions**
  Lists the filter conditions between the primary report and the linked report.
  - **Fields(Primary)**
    Lists the selected fields of the primary report.
  - **OP**
    Specifies the operator to compose the condition. It can be "=", "<>", "<", ">", "<=", ">=" or "IN".
  - **Fields(LinkedReport)**
    Lists the DBFields in the datasets of the linked report which are of the same data type as the selected fields in the primary report.
  - Adds a new condition line.
  - Removes the selected condition.

Parameters tab

This tab is available when the linked report uses parameters. It helps you to assign values to the parameters of the linked report automatically.

- **Parameters**
  Lists all parameters contained in the datasets of the linked report.
- **Value**
  Lists all the DBFields, formulas, summaries and parameters in the datasets of the primary report which are of the same data type as the parameters of the linked report. Select the field the value of which you want to assign to parameter of the linked report from the drop-down list.

Advanced tab

Specifies to pass the values of the filter objects such as filter controls and sliders in the primary report to those in the linked report.

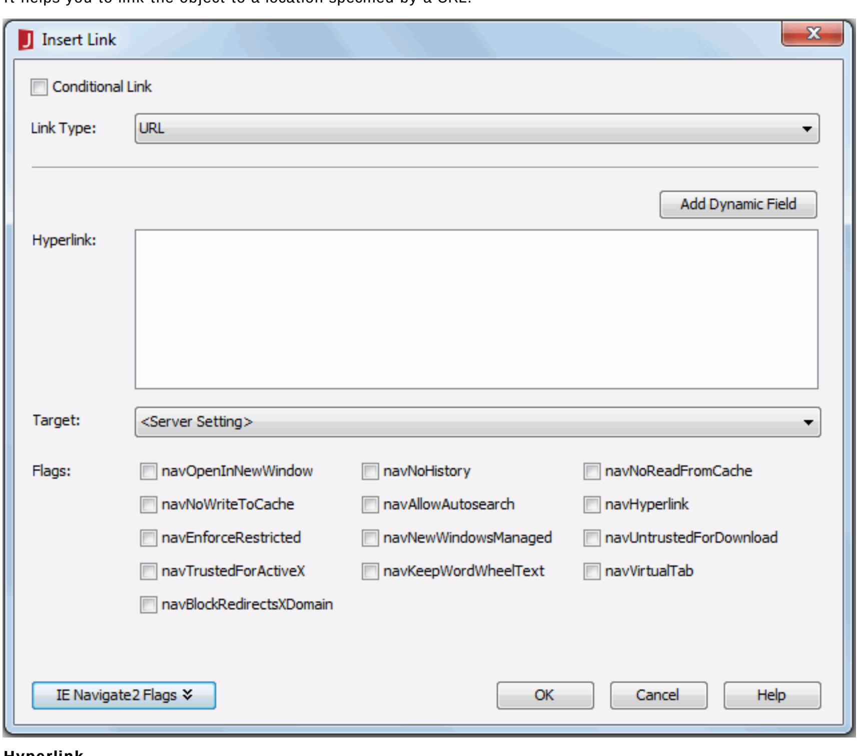
- **Primary Report Property/Object**
  Displays the filter objects in the primary report the values of which will be passed to those in the linked report. Click to select a filter object in the primary report in the Select Value dialog.
- **Linked Report Property/Object**
  Displays the filter objects in the linked report which will adopt the values passed from filter objects in the primary report. Click to select a filter object in the linked report in the Select Value dialog. You need to make sure that values of the filter object in the primary report can be applied to the filter object in the linked report in the same line.
- Adds a new line to pass values between the filter objects in the primary report and the linked report.
- Removes the selected line.

Pass style group information down to linked report

Specifies whether or not to apply the style group of the primary report to the linked report.

URL

It helps you to link the object to a location specified by a URL.



Hyperlink

Specifies the URL of the location that the object will be linked to.

- **Add Dynamic Field**
  Opens the Select Field dialog to insert a field into the URL.

Target

Specifies the window or frame in which to load the location specified by the URL. Applied only when the link is triggered in Web Report Studio; when triggered in JDashboard, except for Same Frame, all the other targets are treated as New Window.

- **<Server Setting>**
  Loads the linked file according to setting of the Target of Detail Table Report and Links option in the Profile > Configure Profile > Page Report Studio > Properties > Default tab on JReport Server.
- **Same Frame**
  Loads the linked file into the same frame as the link.
- **Whole Window**
  Loads the linked file into the full browser window.
- **New Window**
  Loads the linked file into a new window. This window is not named.
- **Parent Frame**
  Loads the linked file into the parent frame of the frame that contains the link.
- **Other Frame**
  Loads the linked file into some other specified frame. If the frame name does not exist, the linked file will be loaded into a new window.

IE Navigate2 Flags

Specifies to use the IE Navigate2 method and its flags to control the behavior of IE when loading the linked file with Internet Explorer. Available only when the link is triggered in a table in Web Report Studio.

Flags

Lists all the flags of the IE Navigate2 method.

To make sure the navigate2 flags work well, uncheck the option **Enable Protected Mode** in Internet Options dialog > Security tab of the Internet Explorer browser, then click the **Custom level** button, in the Security Settings dialog, enable all ActiveX relative options under the ActiveX controls and plug-ins.

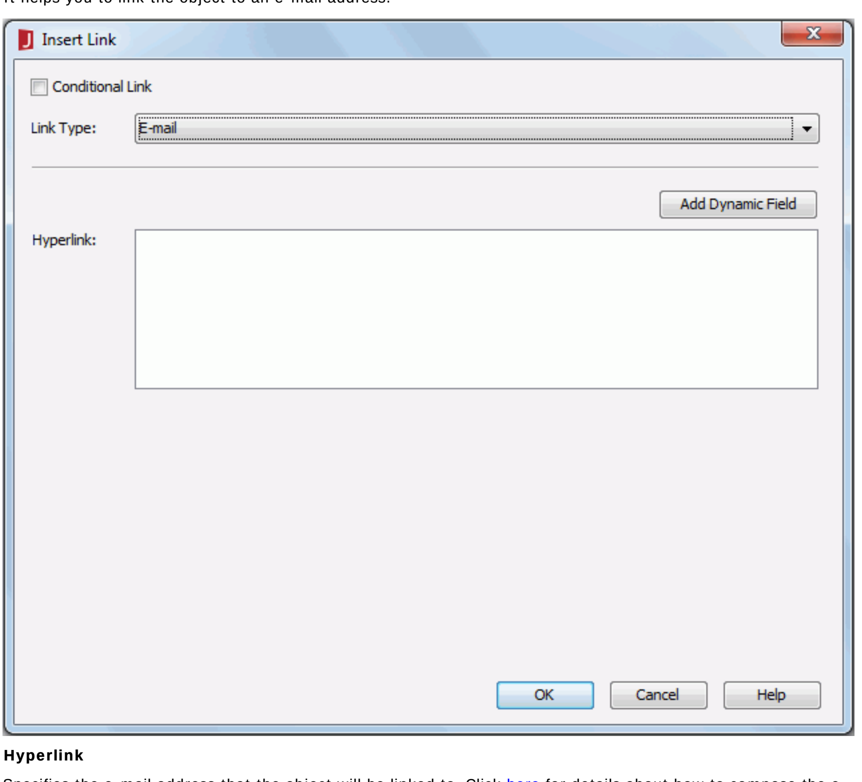
You can see that multiple flags can be selected at a time. Some of them may conflict with each other. The conflict relationship between the flags strictly follows the Microsoft MSDN specification. JReport loyally delivers the user state to the Microsoft interface and does not present or check the conflict.

When you create a new link to a URL, with none of the Navigate2 flags being checked, the linked file will be opened in the specified window if a window name is provided via the "Other Frame or Window" setting in the Target field. If the window name is not given, the linked file will be opened in a new window.

- **navOpenInNewWindow**
  Opens the resource or file in a new window.
- **navNoHistory**
  Does not add the resource or file to the history list. The new page replaces the current page in the list.
- **navNoReadFromCache**
  Not implemented.
- **navNoWriteToCache**
  Not implemented.
- **navAllowAutosearch**
  If the navigation fails, the auto search functionality attempts to navigate common root domains (.com, .edu, and so on). If this also fails, the URL is passed to a search engine.
- **navHyperlink**
  Internet Explorer 6 for Windows XP SP2 and later. If the navigation fails when a hyperlink is being followed, this constant specifies that the resource should then be bound to the moniker using the BINDF_HYPERLINK flag.
- **navEnforceRestricted**
  Internet Explorer 6 for Windows XP SP2 and later. Forces the URL into the restricted zone.
- **navNewWindowsManaged**
  Internet Explorer 6 for Windows XP SP2 and later. Uses the default Popup Manager to block pop-up windows.
- **navUntrustedForDownload**
  Internet Explorer 6 for Windows XP SP2 and later. Blocks files that normally trigger a file download dialog box.
- **navTrustedForActiveX**
  Internet Explorer 6 for Windows XP SP2 and later. Prompts for the installation of ActiveX controls.
- **navKeepWordWheelText**
  For Internet Explorer 7. Maintains state for dynamic navigation based on the filter string entered in the search band text box (wordwheel). Restores the wordwheel text when the navigation completes.
- **navVirtualTab**
  For Internet Explorer 8. Opens the resource as a replacement for the current or target tab. The existing tab is closed while the new tab takes its place in the tab bar and replaces it in the tab group, if any. Browser history is copied forward to the new tab. On Windows Vista, this flag is implied if the navigation would cross integrity levels and navOpenInNewTab, navOpenInBackgroundTab, or navOpenInNewWindow is not specified.
- **navBlockRedirectsXDomain**
  For Internet Explorer 8. Blocks cross-domain redirect requests. The navigation triggers the DWebBrowserEvents2::RedirectXDomainBlocked event if blocked.

E-mail

It helps you to link the object to an e-mail address.



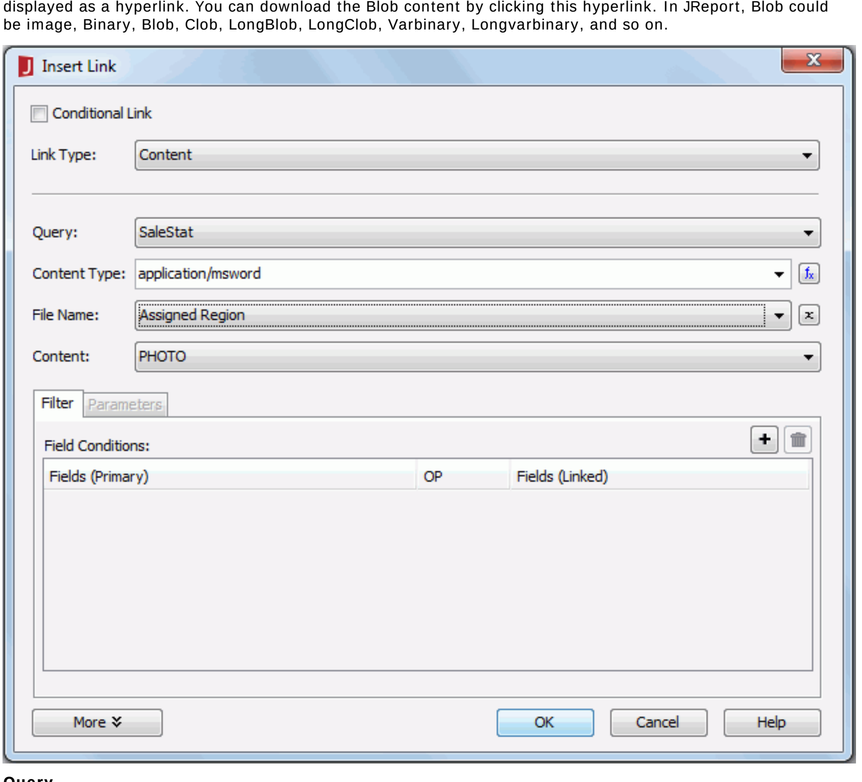
Hyperlink

Specifies the e-mail address that the object will be linked to. Click here for details about how to compose the e-mail address.

- **Add Dynamic Field**
  Opens the Select Field dialog to insert a field into the e-mail address.

Content

It helps you to link the object to a Blob data type field. The Blob data type field will be bound to the object and displayed as a hyperlink. You can download the Blob content by clicking this hyperlink. In JReport, Blob could be image, Binary, Blob, Clob, LongBlob, LongClob, Varbinary, Longvarbinary, and so on.



Query

Specifies the business view which contains the required Blob data type field. Only query based business views from the same data source connection as the current report are listed.

When you set the link type to Content, JReport will check whether there are such kind of business views available. If not, options for this link type will be disabled.

Content Type

Specifies the content type for the Blob type data. You can click the button to bind it with a string type business view element or a record level pass one formula in the selected business view, or with a dynamic formula in the current report.

File Name

Specifies the file name for the linked Blob type data. You can click the button to bind it with a string type business view element or a record level pass one formula in the selected business view.

Content

Specifies the Blob data type field in the selected business view.

Filter tab

Specifies the link conditions between the business view used by the current report and the business view that contains the linked Blob content.

- **Field Conditions**
  Lists the filter conditions between the two business views.
  - **Fields(Primary)**
    Lists the selected fields of the business view used by the current report.
  - **OP**
    Specifies the operator to compose the condition. It can be "=", "<>", "<", ">", "<=", ">=" or "IN".
  - **Fields(Linked)**
    Lists the DBFields in the datasets of the business view that contains the linked Blob content, which are of the same data type as the selected fields in the business view used by the current report.
  - Adds a new condition line.
  - Removes the selected condition

Parameters tab

This tab is available when the business view that contains the linked Blob content uses parameters. It helps you to assign values to the parameters automatically.

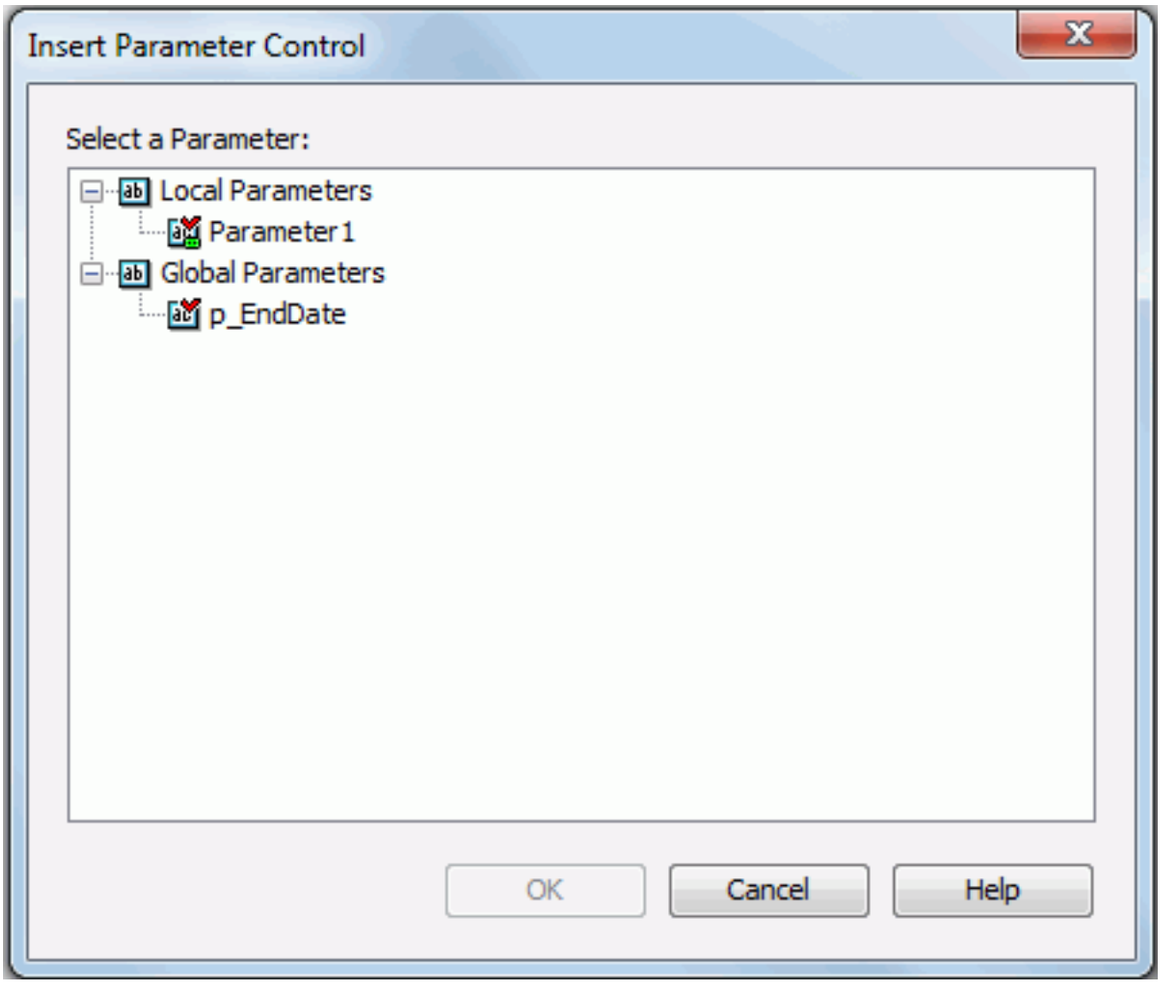
- **Parameters**
  Lists all the parameters the business view that contains the linked Blob content uses.
- **Value**
  Lists all the DBFields, formulas, summaries and parameters in the business view used by the current report, which are of the same data type as the parameters the business view that contains the linked Blob content uses. Select the field the value of which you want to assign to a parameter from the drop-down list.

Related topic:

- Binding a link to a label

Insert Parameter Control dialog

The dialog appears when you click Insert > Web Controls > Parameter Control on the menu bar, drag Parameter Control from the Toolbox panel, or click the down arrow on the Insert Control Objects button and select Parameter Control from the drop-down list. It helps you to insert a parameter control into a report for delivering parameter values.



The followings are details about options of this dialog:

Select a Parameter

Lists all the parameters except cascading parameters used in the current report. Specfiy the one you want to use in the parameter control.

OK

Inserts the parameter control into the report and closes the dialog.

Cancel

Cancels the operation and closes the dialog.

Help

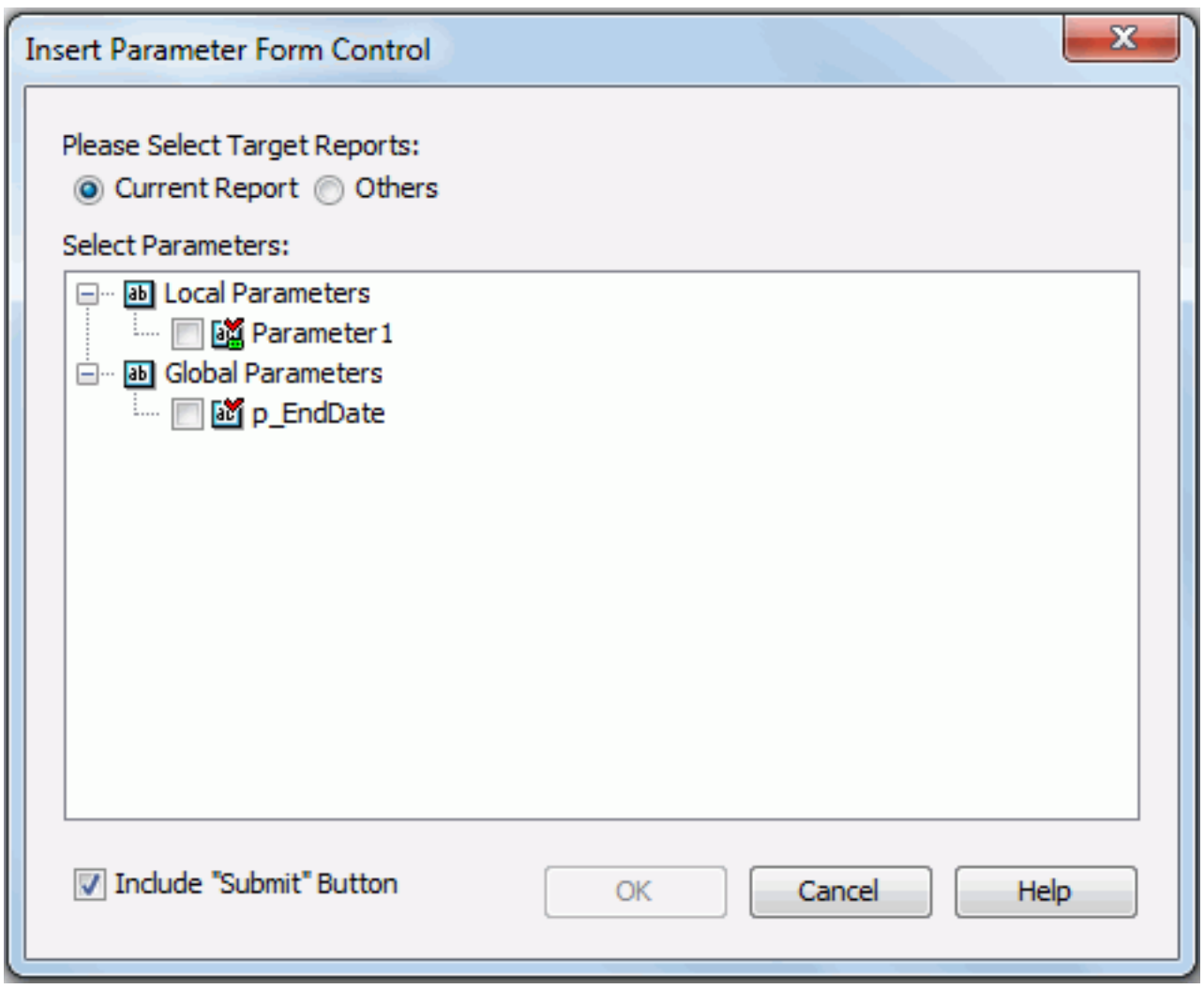
Displays the help document about this feature.

Related topics:

- Using parameter control to specify a parameter to a report

Insert Parameter Form Control dialog

The dialog appears when you click Insert > Web Controls > Parameter Form Control on the menu bar, drag Parameter Form Control from the Toolbox panel, or click the down arrow on the Insert Control Objects button and select Parameter Form Control from the drop-down list. It helps you to insert a parameter form control into a report for running reports from the current report.



The followings are details about options of this dialog:

Please Select Target Reports

Specifies the reports to run.

- **Current**
  Specifies to run the current report with the specified parameters.
  - **Select Parameters**
    Specifies the parameters that are used to run the current report. Once you select one of a group of cascading parameters, all the other parameters in the cascading group will be selected automatically, and vice versa.
- **Others**
  Specifies to run other reports.
  - **Reports**
    Specifies the target reports to run. If all the selected reports contain no parameters, you cannot finish the dialog.
  - **Parameter Information**
    You can click a report and then view its parameters here.

Include "Submit" Button

Specifies whether the inserted parameter form control contains the Submit button. If Submit is included, it is used to submit the parameter values end users specified in the parameter form control at runtime. If Submit is not included, once end users change the values of a parameter in the parameter form control at runtime, the new values will be applied automatically.

OK

Inserts the parameter form control into the report and closes the dialog.

Cancel

Cancels the operation and closes the dialog.

Help

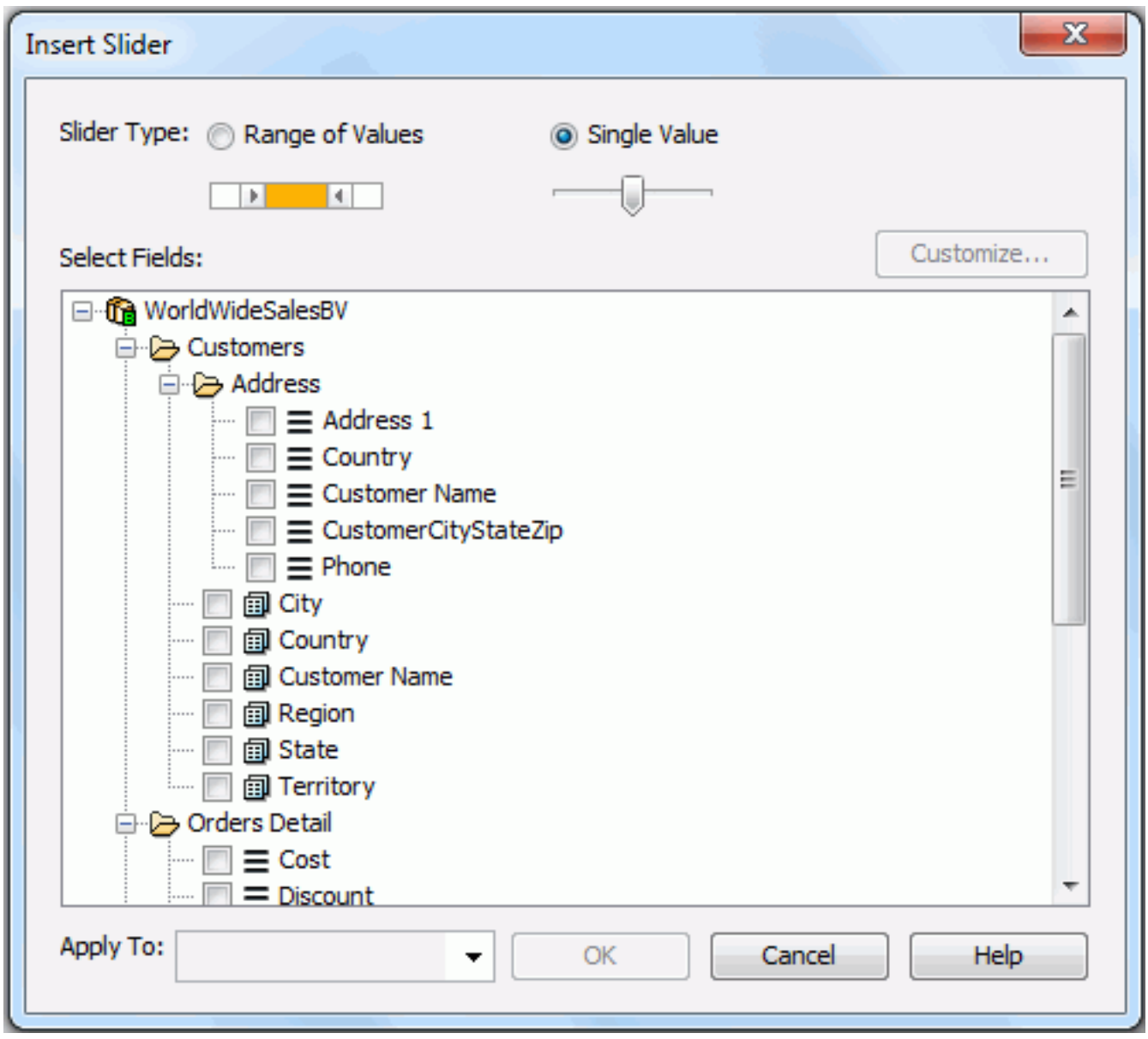
Displays the help document about this feature.

Related topics:

- Using parameter form control to run reports

Insert Slider dialog

The dialog appears when you click Insert > Web Controls > Slider on the menu bar, drag Slider from the Toolbox panel, or click the down arrow on the Insert Control Objects button and select Slider from the drop-down list. It helps you to insert a slider into a library component to filter component data.



The following are details about options in the dialog:

Slider Type

Specifies the type of the slider.

- **Range of Values**
  If selected, you can select a range of values on the slider at runtime.
- **Single Value**
  If selected, you can select only one value on the slider at runtime, not available for numeric data type.

Select Fields

Specifies the fields to bind to the slider. All the selected fields should be of the same data type.

The common usage is to select one field, and then based on the field to filter the data of the components created from the same data source as the field.

To filter components using different data sources, choose a common field all the data sources contain and select the field in all the data sources.

Customize

Opens the Customize Value dialog to customize the values in the selected fields you want to use on the slider.

Apply To

Specifies the target data components you want to apply the slider to.

OK

Inserts the slider into the library component and closes the dialog.

Cancel

Cancels the operation and closes the dialog.

Help

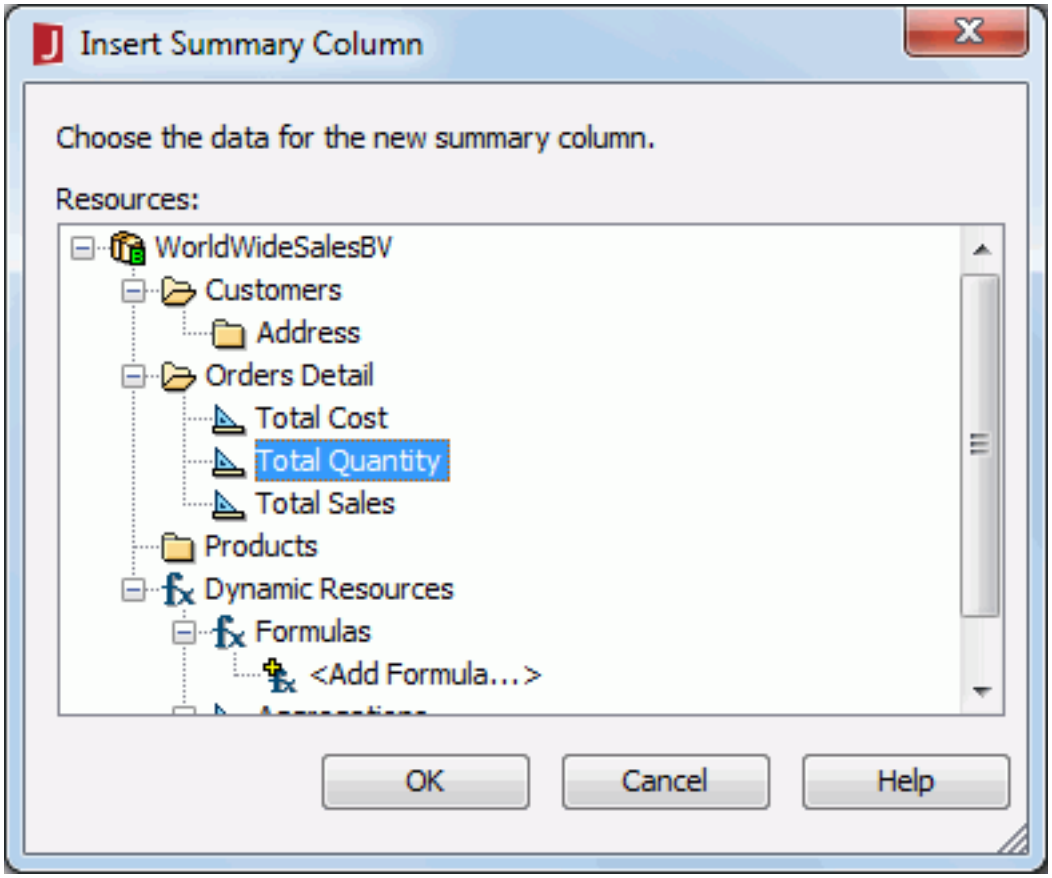
Displays the help document about this feature.

Related topics:

- Using slider or filter control to filter data

Insert Summary Column dialog

The dialog appears when you right-click a table or a table column and select Insert > Summary Column from the shortcut menu. It helps you to insert new summary columns into a table.



The followings are details about options of this dialog:

Resources

Lists all the available data resources for the new summary column. Select the resource you want from the list.

OK

Inserts the column with the specifies resource and closes the dialog.

Cancel

Does not retain any changes and closes the dialog.

Help

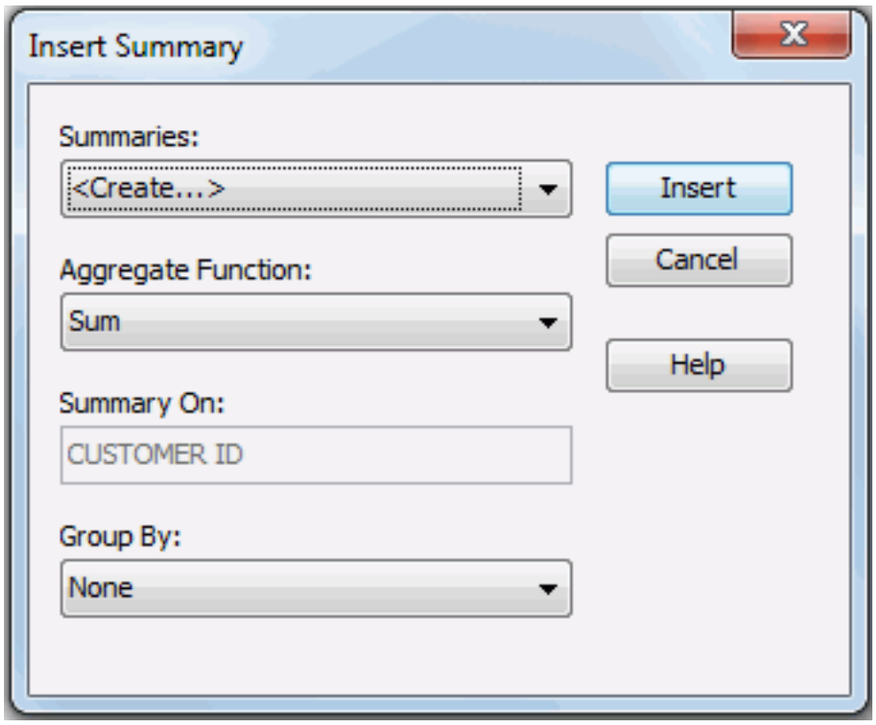
Displays the help document about this feature.

Related topics:

- Managing columns/rows in a table

Insert Summary dialog

This dialog appears when you click Insert > Summary on the menu bar, or the Insert Summary button on the Insert toolbar, or right-click a DBField and select Summary Function from the shortcut menu. It helps you to insert a selected or newly created summary into a report.



The following are details about options in the dialog:

Summaries

To create a new summary, select <Create...> from the drop-down list; to insert an existing summary into the report, select it from the drop-down list.

Function

Lists functions available for the selected field to be summed on.

Summary On

Displays the field on which the summary is to compute.

Group By

Specifies the field used to group the data. If selected, a summary of each group will be calculated.

Insert

Inserts the selected or newly created summary into the report.

Cancel

Does not retain any changes and closes this dialog.

Help

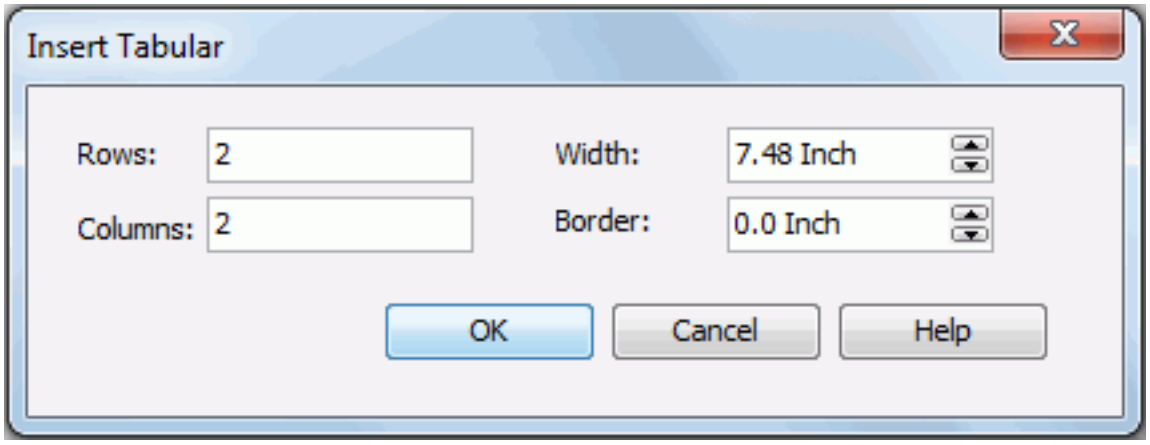
Displays the help document about this feature.

Related topics:

- Inserting a summary field in a report

Insert Tabular dialog

This dialog appears when you click Insert > Tabular on the menu bar, or the Insert Tabular button on the Insert toolbar. It helps you to set options for inserting a tabular into a report.



The following are details about options in the dialog:

Rows

Specifies the number of rows for the tabular.

Columns

Specifies the number of the columns for the tabular.

Width

Specifies the width for the tabular in inches.

Border

Specifies the length of the border for the tabular, in inches.

OK

Accepts all changes and closes the dialog.

Cancel

Does not retain any changes and closes the dialog.

Help

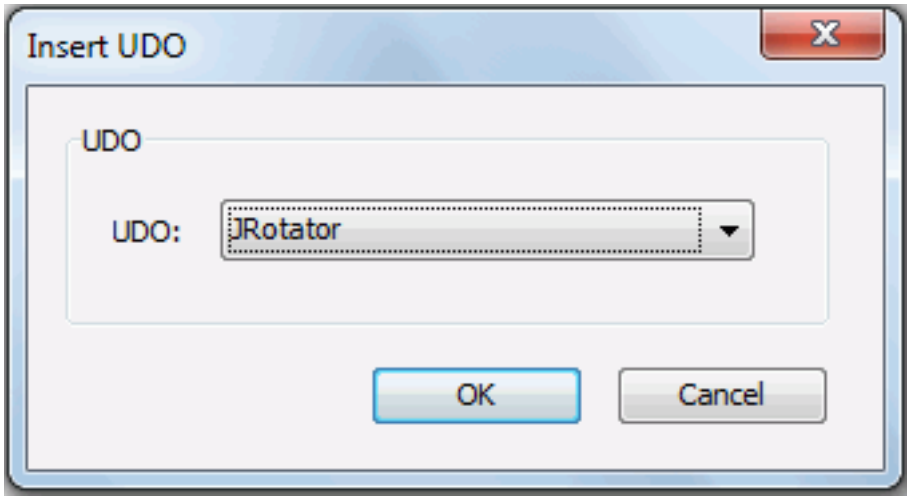
Displays the help document about this feature.

Related topics:

- Inserting a tabular in a report

Insert UDO dialog

This dialog appears when you click Insert > UDO on the menu bar, or the Insert UDO button on the Insert toolbar. It helps you to insert a user defined object into a banded object in a report.



The following are details about options in the dialog:

UDO

Specifies the UDO to insert in the report. The UDOs listed in the drop-down list are predefined.

OK

Accepts all changes and closes the dialog.

Cancel

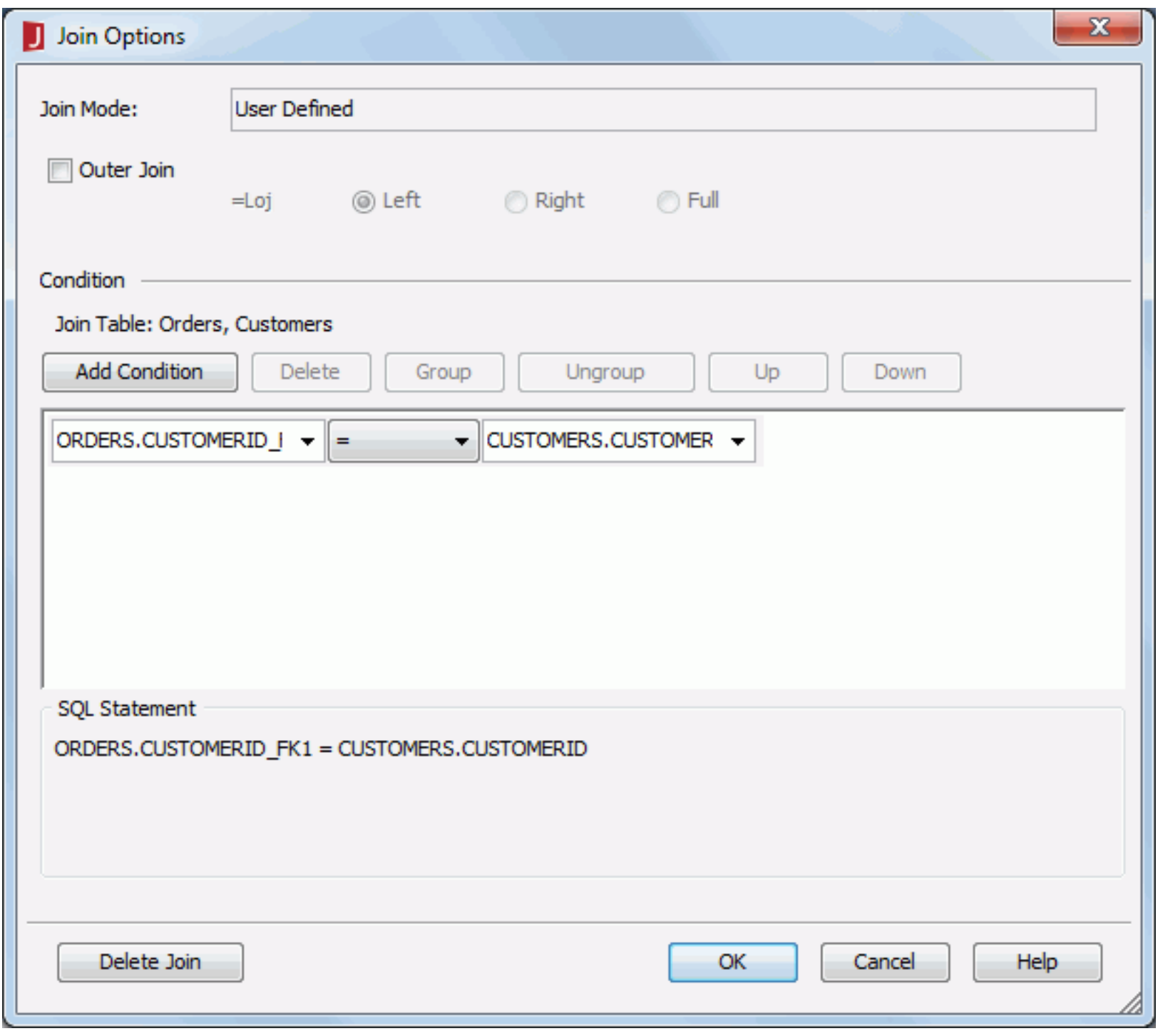
Does not retain any change and closes the dialog.

Related topics:

- Inserting a UDO into a report

Join Options dialog

This dialog appears when you double-click the join button of a data source pre-join, a business view join or a query join. It helps you to edit the conditions of the specified join.



The following are details about options in the dialog:

Join Mode

Displays the mode of the join.

Outer Join

Specifies to make the join an outer join.

Condition

Specifies the join conditions.

- **Join Table**
  Displays the tables on which the join is set up.
- **Add Condition**
  Adds a new condition line.
- **Delete**
  Deletes the selected condition line.
- **Group**
  Makes the selected conditions in one group. Conditions can also be added to an existing group by selecting the conditions and the group while holding the Ctrl button, and then clicking the Group button.
- **Ungroup**
  Makes the selected condition ungrouped.
- **Up**
  Moves the selected condition up to a higher level.
- **Down**
  Moves the selected condition or group down to a lower level.
- **Logic**
  Lists the logic operator.
  - **And**
    Logic operator And which is applied between this and the next line or between more than two lines of a group.
  - **Or**
    Logic operator Or which is applied to this and the next line.
  - **And Not**
    Logic operator And Not which is applied to this and the next line.
  - **Or Not**
    Logic operator Or Not which is applied to this and the next line.
- **Column**
  Specifies the column that will be used in the join condition.
- **Operator**
  Specifies the operator to compose the join condition.
  - **=**
    Displays values that are equal to each other.
  - **>**
    Displays values that are greater on the left side.
  - **<**
    Displays values that are greater on the right side.
  - **> =**
    Displays values that are greater than or equal to the left side.
  - **< =**
    Displays values that are greater than or equal to the right side.
  - **< >**
    Displays values that are not equal to each other.
- **SQL Statement**
  Displays the SQL statement of the join condition.

Delete Join

Deletes the join.

OK

Applies the changes and closes the dialog.

Cancel

Cancels the changes and exits the dialog.

Help

Displays the help document about this feature.

Related topics:

- Defining the join relationships between tables
- Joining tables in a query

JSON Connection Wizard

This wizard appears when you select JSON and click OK in the New Data Source dialog, or in the Catalog Browser right-click a data source and select Add JSON Connection from the shortcut menu or right-click an existing JSON connection and select Edit Connection from the shortcut menu. It helps you to import an XML schema and transform it to a relational schema for future use, and consists of the following screens:

- Extract JSON Schema screen
- Modify Schema Properties screen
- Transformed Relational Schema screen
- Add Table screen

Back
Goes back to the previous screen.

Next
Goes to the next screen.

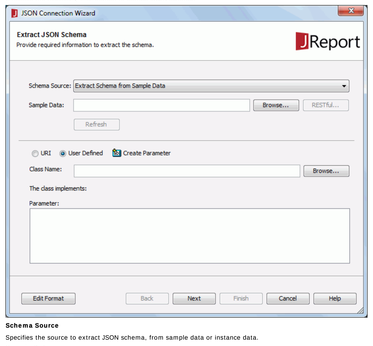
Finish
Finishes confirming to transform a JSON schema to a relational schema.

Cancel
Does not retain changes and closes this wizard.

Help
Displays the help document about this feature.

Extract JSON Schema screen

Specifies the necessary information to extract JSON schema.



Schema Source
Specifies the source to extract JSON schema, from sample data or instance data.

Sample Data
Specifies the sample file by clicking the Browse button or inputting the URI string directly in the text box. It is available only when you select Extract Schema from Sample Data from the Schema Type drop-down list.

- **RESTful**
  Opens the RESTful Data Source Options dialog to specify the RESTful data options for the remote sample data. Available only when the schema is from a remote sample file.

Refresh
Available only when you edit an existing JSON connection. Once you make some change in the wizard and click the Refresh button, the JSON schema information will be reloaded according to what you have changed.

Instance
Specifies the JSON instance. It should match with the selected JSON schema. Not available when you select Extract Schema from Sample Data from the Schema Type drop-down list and select User Defined.

- **RESTful**
  Opens the RESTful Data Source Options dialog to specify the RESTful data options for the remote URI data. Available only when the JSON file is from a remote file.

Edit Format
Opens the Edit Format dialog to specify the format of the referenced values.

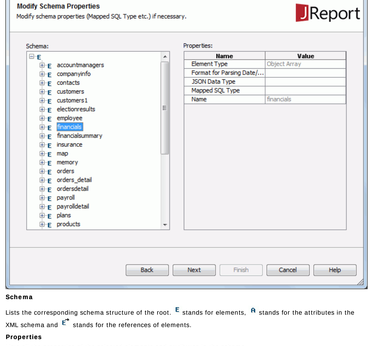
- **Name**
  Displays names of the referenced values.
- **Format**
  Specifies formats for the referenced values.
- **OK**
  Accepts changes and closes the dialog.
- **Cancel**
  Does not retain changes and closes the dialog.

The three operations below are available only when you select Extract Schema from Sample Data from the Schema Type drop-down list.

- **URI**
  If selected, specify a JSON file by clicking the Browse button or input the URI string directly in the text box. You can use parameters or constant level formulas predefined in the current catalog data source or the special field User Name in the string, and the format of the predefined parameters used in the string is "@" + parameter name. For example, if the parameter string of a class is SQL=select * from employee, and you want to use the parameter sql in a catalog to replace the part after "=" in the string, then the parameter string of the class will be SQL=@sql. Moreover, if a parameter string contains characters, such as @, ',' or double quotation marks, or other strings that do not need to be parsed, quote the whole string with double quotation marks.
- **User Defined**
  Specifies to get data from user defined interface.
  - **Class Name**
    Specifies the full name (including package name) of the class by clicking the Browse button or inputting the class name directly in the text box. The class should be appended to the class path in the system environment.
  - **The class implements**
    After filling in the Class Name field, the class name of the interface that the class implements will be displayed here.
  - **Parameter**
    Specifies the parameter string for the user defined data interface. The parameter string must match the format defined in the class. You can use parameters or constant level formulas predefined in the current catalog data source or the special field User Name in the string, and the format of the predefined parameters used in the string is "@" + parameter name. For example, if the parameter string of a class is SQL=select * from employee, and you want to use the parameter sql in a catalog to replace the part after "=" in the string, then the parameter string of the class will be SQL=@sql. Moreover, if a parameter string contains characters, such as @, ',' or double quotation marks, or other strings that do not need to be parsed, quote the whole string with double quotation marks.
- **Create Parameter**
  Opens the Create Parameter dialog to create a parameter, and then you can dynamically specify a JSON instance or user defined interface that matches with the selected JSON schema at runtime.

Modify Schema Properties screen

Specifies properties of the JSON schema.



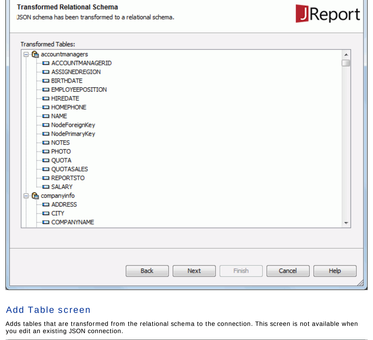
Schema
Lists the corresponding schema structure of the root. stands for elements, stands for the attributes in the XML schema and stands for the references of elements.

Properties
Lists all the properties of the selected elements and attributes in the schema.

- **Name**
  Displays names of properties of the selected elements or attributes.
- **Value**
  Specifies values of properties for the selected elements or attributes.

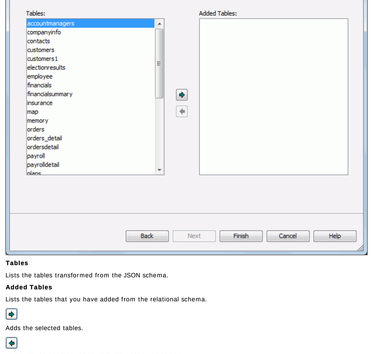
Transformed Relational Schema screen

This screen lists the transformed relational schema structure. It helps you to click Next to confirm the transformation results and go into the Add Table screen.



Add Table screen

Adds tables that are transformed from the relational schema to the connection. This screen is not available when you edit an existing JSON connection.



Tables
Lists the tables transformed from the JSON schema.

Added Tables
Lists the tables that you have added from the relational schema.

Adds the selected tables.

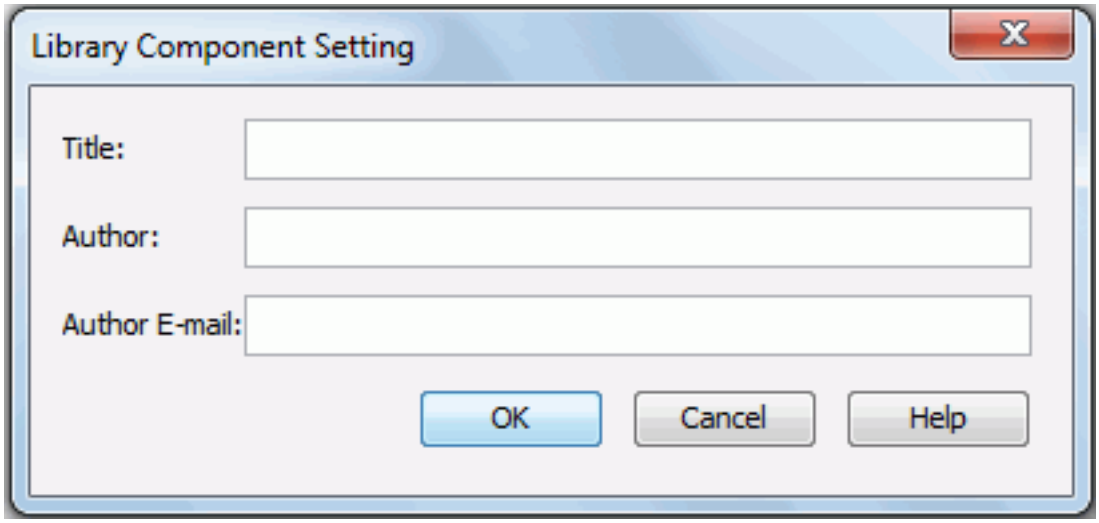
Removes the selected tables from the Added Tables box.

Related topics:

- Setting up a JSON connection

Library Component Setting dialog

This dialog appears when you right-click a table, chart, crosstab, or geographic map in a web report, select Save as Library Component from the shortcut menu, then click Save after specifying the path and name of the library component in the Save As dialog. It helps you to save the selected data component in a web report as a library component.



The following are details about options in the dialog:

Title

Specifies the title of the library component.

Author

Specifies the name of the author.

Author E-mail

Specifies the e-mail address of the author.

OK

Saves the data component in the web report as a library component.

Cancel

Closes the dialog, leaving any changes unsaved.

Help

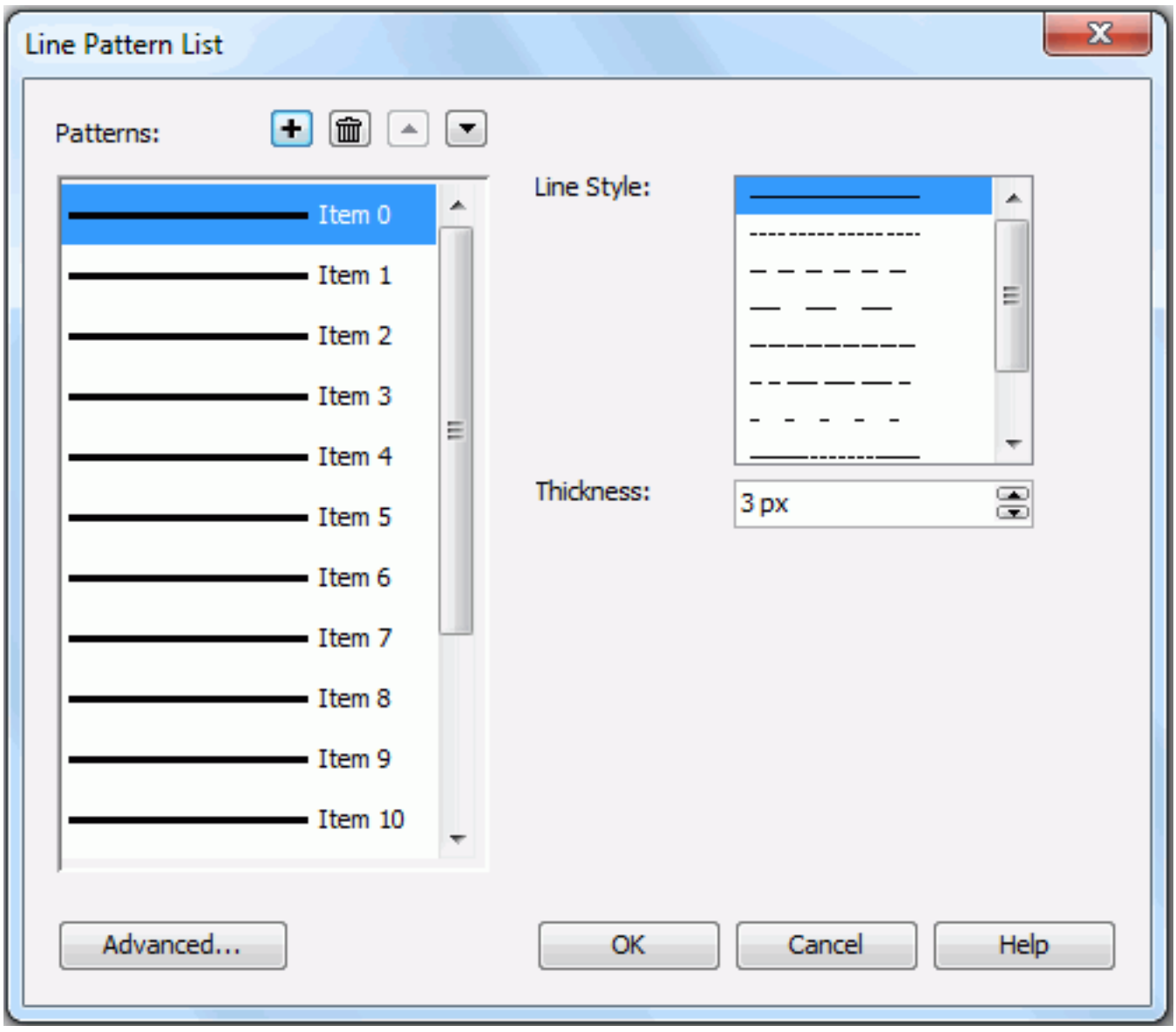
Displays the help document about this feature.

Related topics:

- Creating a library component by saving a data component in a web report

Line Pattern List dialog

This dialog appears when you click the Line Pattern List button in the Format Line dialog for page report, web report, or library component. It helps you to modify the line pattern in a line chart.



The following are details about options in the dialog:

Patterns

Specifies the pattern for the lines. Click the items in the box one by one to set the pattern for each line.

- Adds a new pattern item.
- Removes the selected pattern item.
- Moves the selected pattern item one step up.
- Moves the selected pattern item one step down.

Line Style

Specifies the style of the line.

Thickness

Specifies the thickness of the line, in pixels.

Advanced

Opens the Advanced dialog to specify the line pattern for certain values.

OK

Applies the line patterns and closes the dialog.

Cancel

Does not retain any changes and closes the dialog.

Help

Displays the help document about this feature.

Related topics:

- Formatting the lines

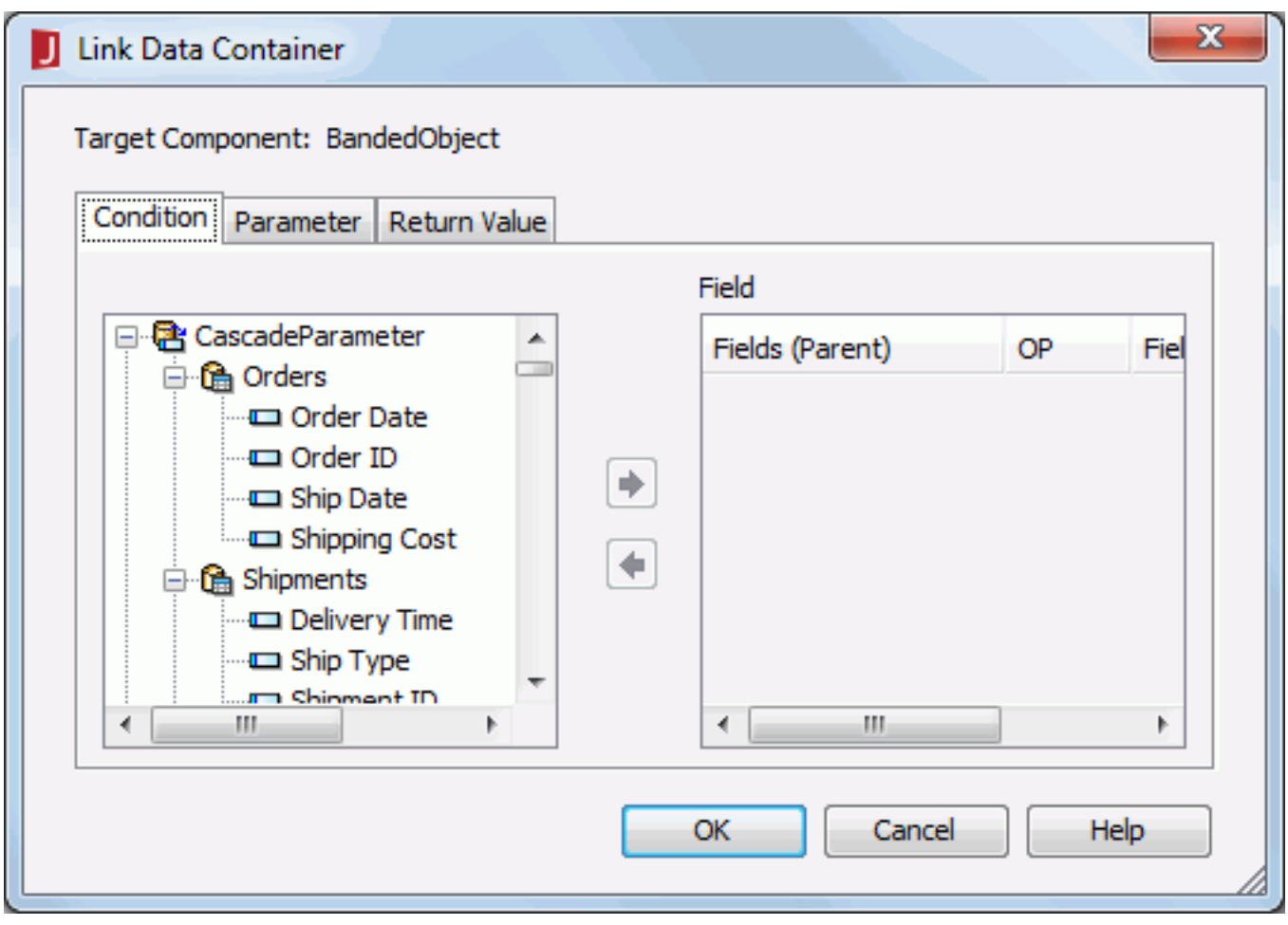
Link Data Container dialog

This dialog appears when you right-click a child data container and select Data Container Link from the shortcut menu. It helps you to set up some link conditions between the child data container and its parent.

The following are details about options in the dialog:

Condition tab

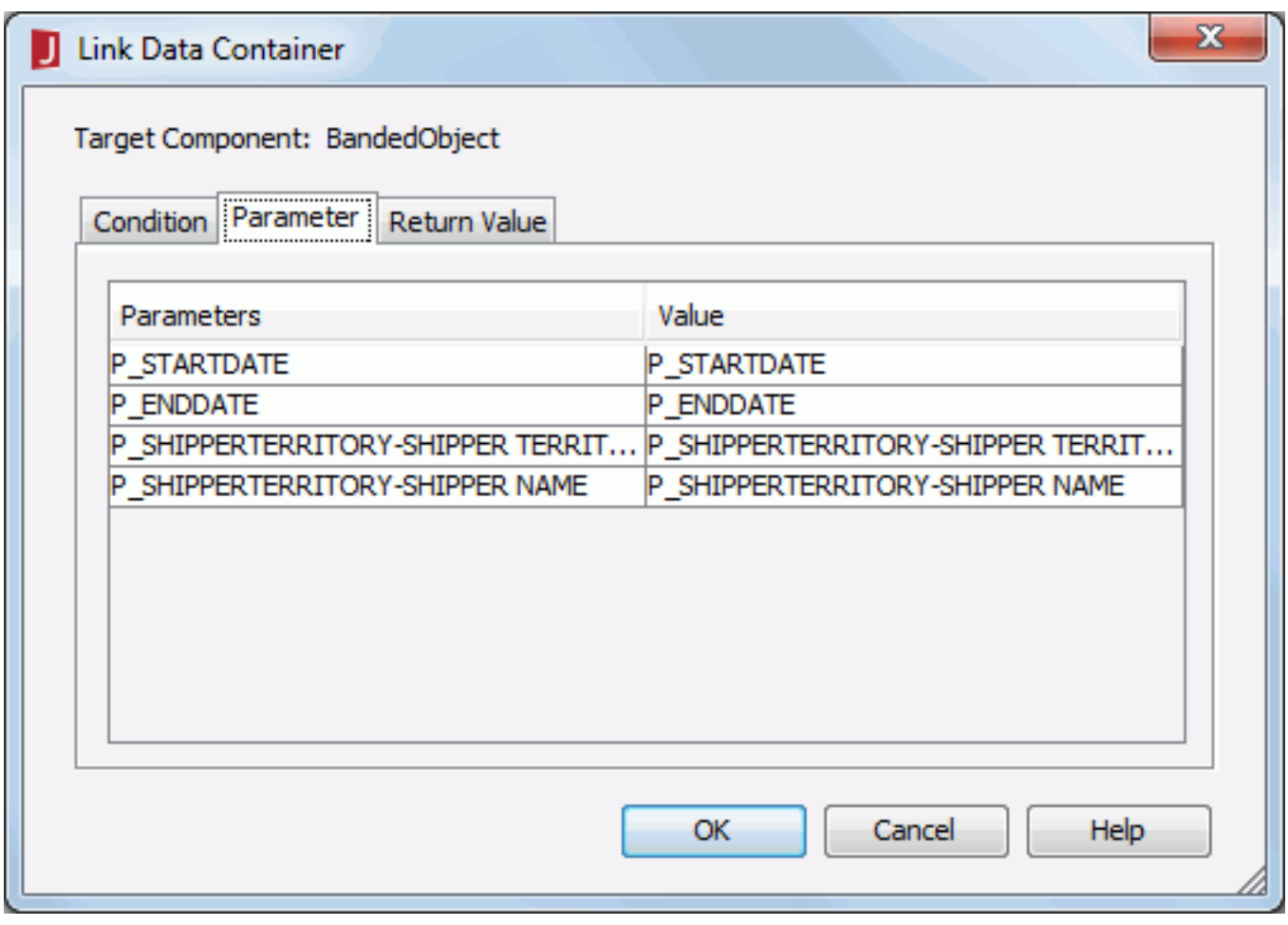
Specifies the link conditions between the parent and child data containers.



- Adds a field of the parent data container to set up links between the parent and child data containers.
- Removes the selected field.
- **Field**
  Lists all the fields that are used in the link conditions.
  - **Fields(Parent)**
    Lists the selected fields of the parent data container.
  - **OP**
    Specifies the operator to setting up links between the parent and child data containers. It can be =, <>, >, >=, <, <=, IN or NOT IN.
  - **Fields(Child)**
    Lists the DBFields in the dataset of the child data container which are of the same data type as the selected fields in the parent data container.

Parameter tab

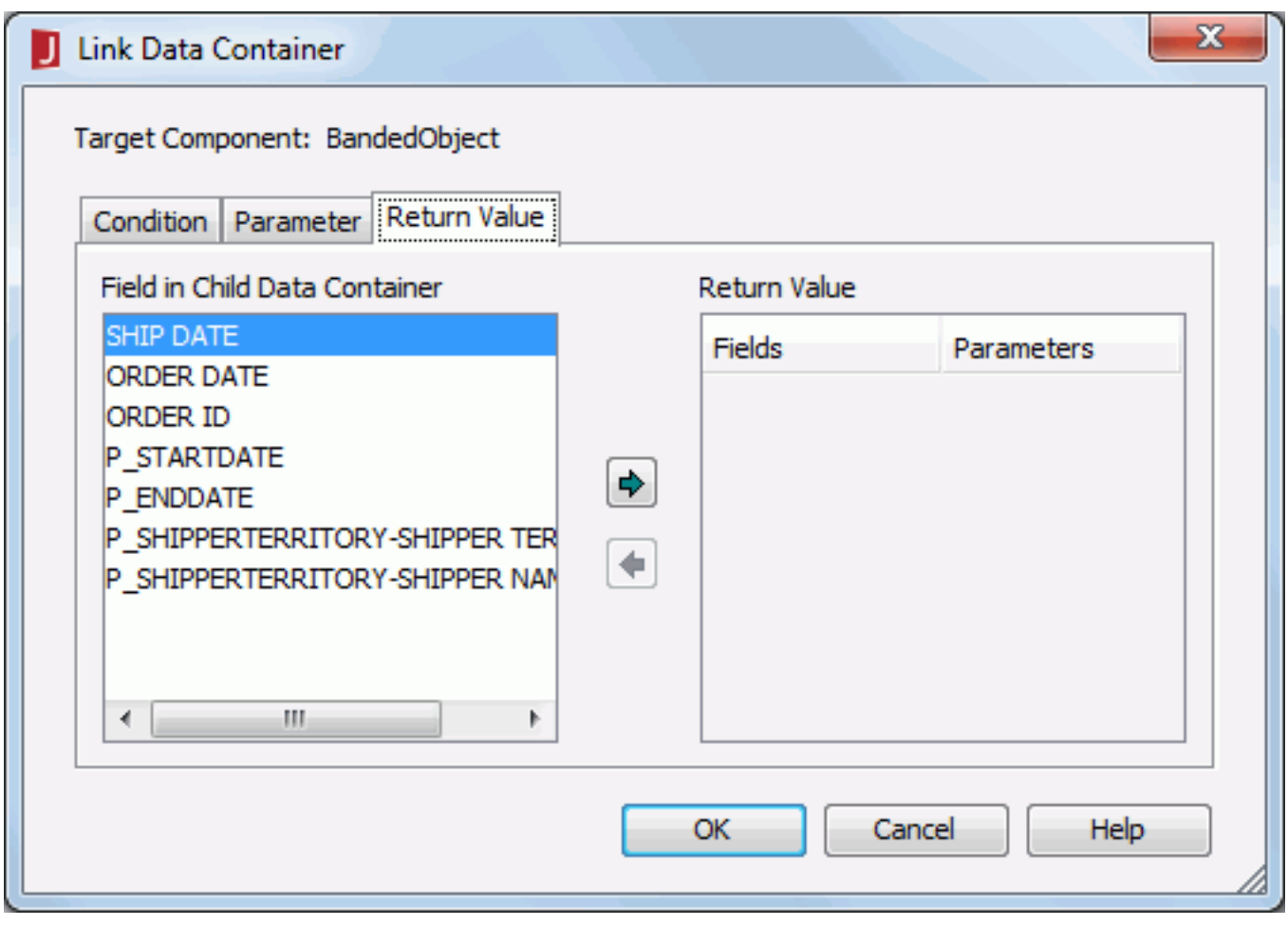
Specifies values for all parameters in the child data container.



- **Parameters**
  Lists all parameters contained in the dataset of the child data container.
- **Value**
  Lists all the DBFields, formulas, summaries and parameters in the dataset of the parent data container which are of the same data type as the parameters of the child data container.

Return Value tab

Specifies values of the child data container fields which will be returned as the parent data container parameter values.



- **Fields in Child Data Container**
  Lists all the fields in the child data container.
- Adds a field in the child data container whose value will be returned to a parameter of the parent data container.
- Removes the selected field.
- **Return Value**
  Lists all the fields in the child data container that will be used to return value to parameters of the parent data container.
  - **Fields**
    Lists the selected fields in the child data container.
  - **Parameters**
    Lists all the parameters in the parent data container which are of the same data type as the fields selected in the child data container.

OK

Applies all changes and closes the dialog.

Cancel

Does not retain any changes and closes the dialog.

Help

Displays the help document about this feature.

Related topics:

- Setting up data container link

Mailing Label Wizard

This wizard appears when you select the Mailing Label layout in the New Page Report Tab dialog or New Page Report dialog and then click OK. It helps you to create a report with a cross banded object in it, which is in the form of a mailing label layout, and consists of the following screens.

- Data screen
- Display screen
- Group screen
- Summary screen
- Chart screen
- Filter screen
- Layout screen
- Style screen

Back

Goes back to the previous screen.

Next

Goes to the next screen.

Finish

Finishes creating the mailing label report and closes this wizard.

Cancel

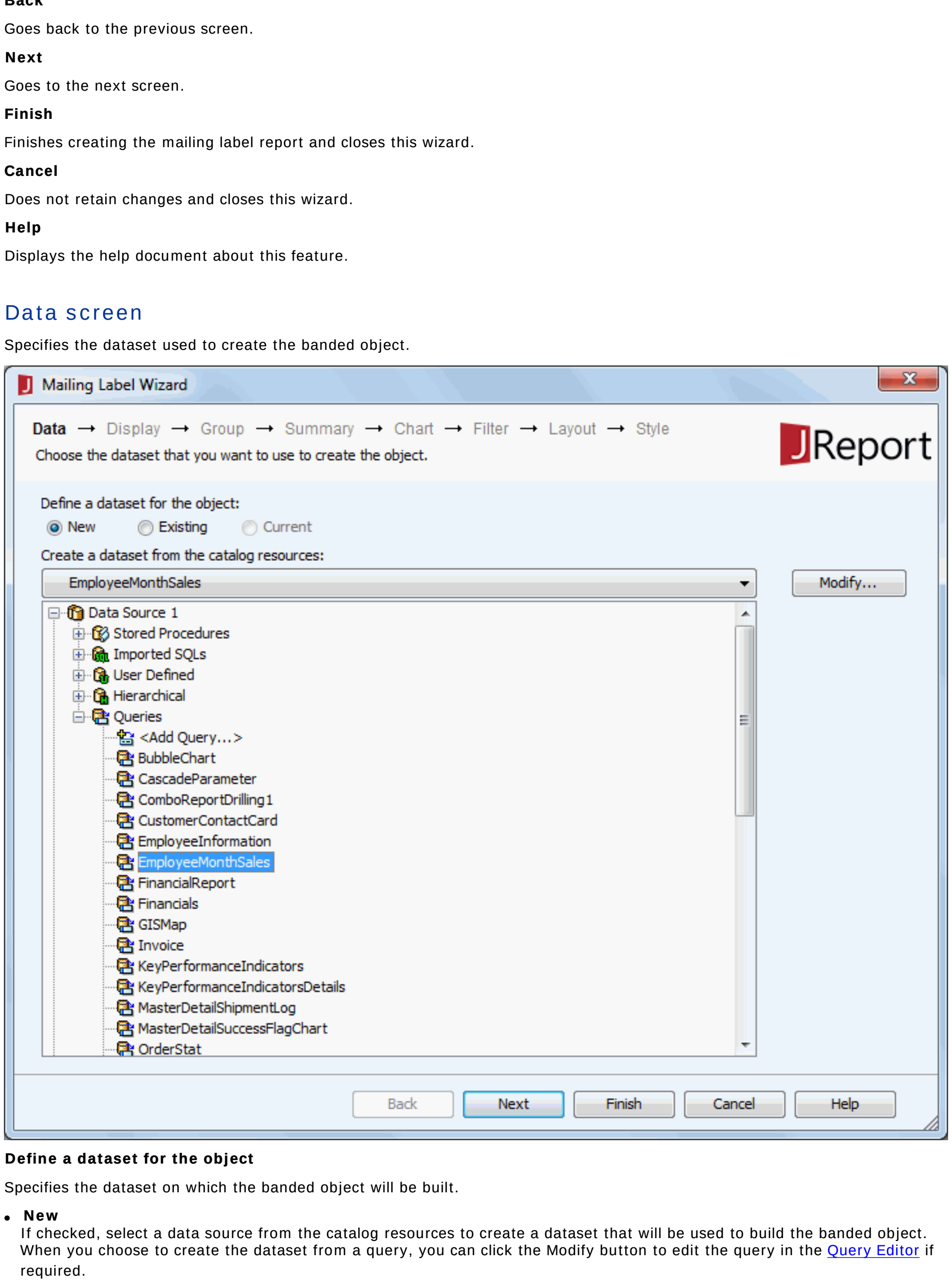
Does not retain changes and closes this wizard.

Help

Displays the help document about this feature.

Data screen

Specifies the dataset used to create the banded object.



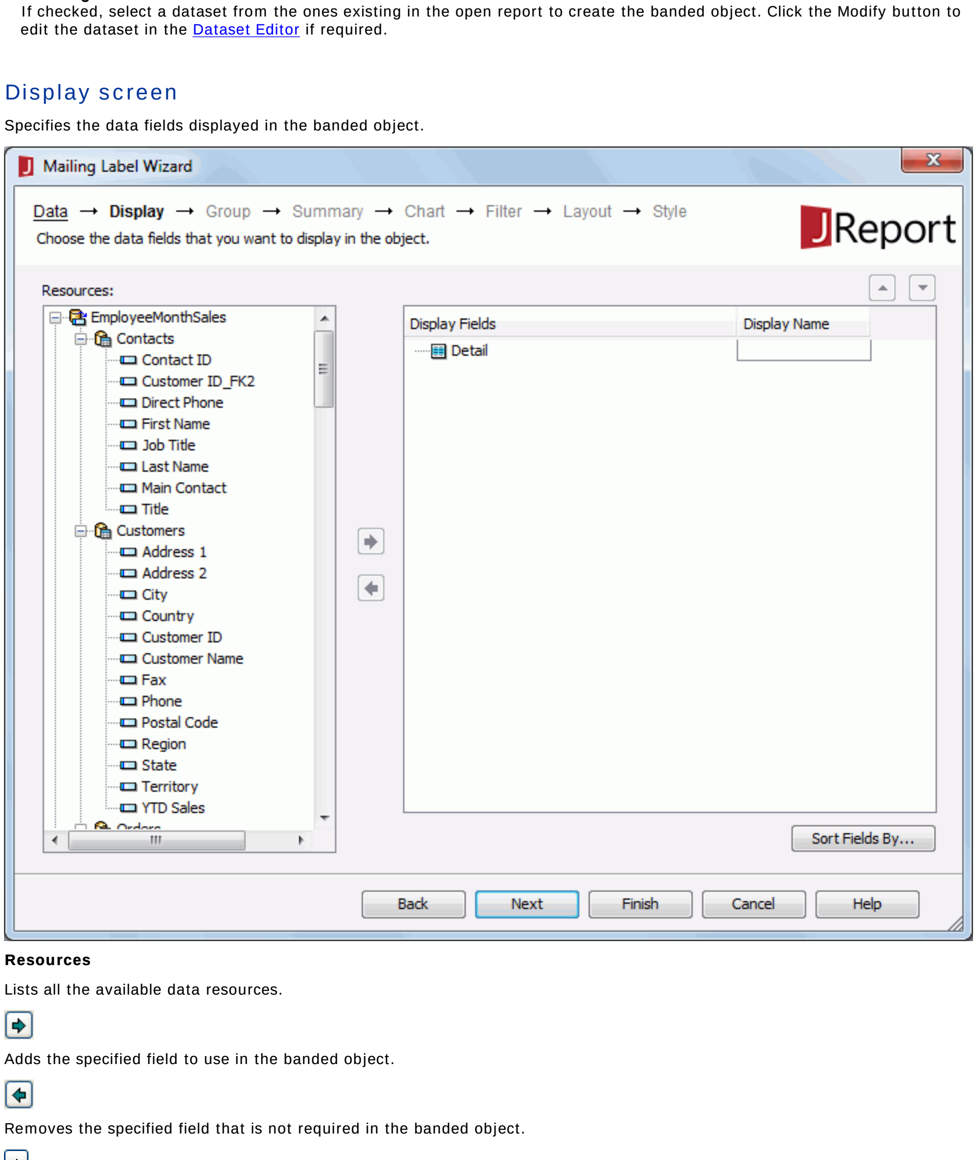
Define a dataset for the object

Specifies the dataset on which the banded object will be built.

- **New**
  If checked, select a data source from the catalog resources to create a dataset that will be used to build the banded object. When you choose to create the dataset from a query, you can click the Modify button to edit the query in the Query Editor if required.
- **Existing**
  If checked, select a dataset from the ones existing in the open report to create the banded object. Click the Modify button to edit the dataset in the Dataset Editor if required.

Display screen

Specifies the data fields displayed in the banded object.



Resources

Lists all the available data resources.

Adds the specified field to use in the banded object.

Removes the specified field that is not required in the banded object.

Moves the specified field one step up.

Moves the specified field one step down.

Display Fields

Lists the fields that have been selected to display in the banded object.

Display Name

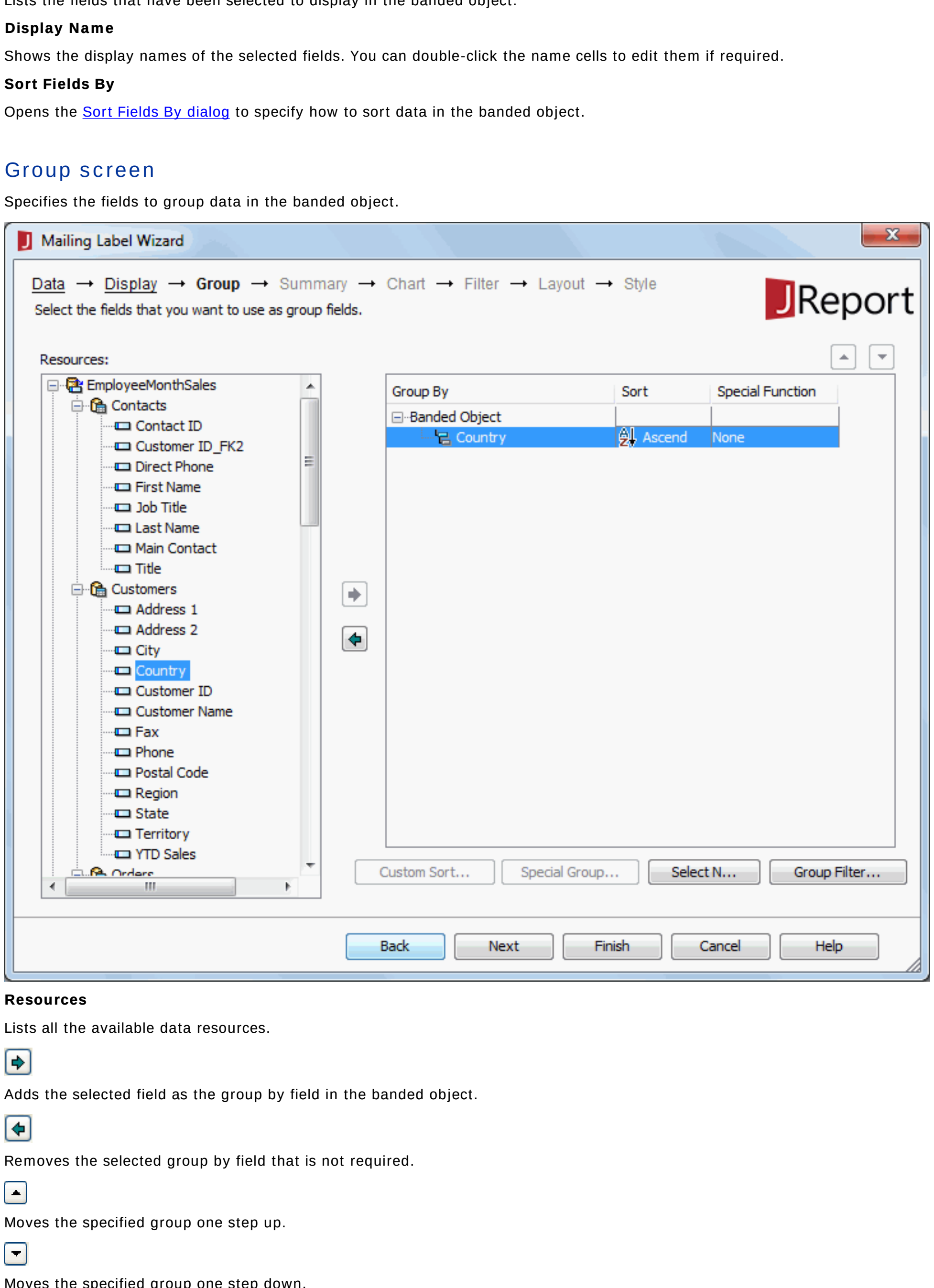
Shows the display names of the selected fields. You can double-click the name cells to edit them if required.

Sort Fields By

Opens the Sort Fields By dialog to specify how to sort data in the banded object.

Group screen

Specifies the fields to group data in the banded object.



Resources

Lists all the available data resources.

Adds the selected field as the group by field in the banded object.

Removes the selected group by field that is not required.

Moves the specified group one step up.

Moves the specified group one step down.

Group By

Lists the fields that are used to group data in the banded object.

Sort

Specifies how groups at the specific group level will be sorted.

- **Ascend**
  Groups will be sorted in an ascending order (A, B, C).
- **Descend**
  Groups will be sorted in a descending order (C, B, A).
- **No Sort**
  Groups will be sorted in the original order in database.
- **Special Group**
  Opens the User Defined Group dialog to define grouping information.

  For example, if you placed a field named Region for grouping, and this field contains all 50 states of the United States; and if you want to see the data between Maryland (MD) and New York (NY), you can define the criteria by selecting the between operator to further define your grouping information.
- **Custom Sort**
  Opens the Custom Sort dialog to set how groups will be sorted.

Special Function

If the group by field is of Numeric/String/Date/Time type, you can select a special function for the field in the Special Function column to further specify to which level the data will be grouped by.

If Customize is selected, the Customized Function dialog will be displayed, in which you can set the function by your own.

Custom Sort

Specifies how to sort the groups. Activated only when you have clicked Custom Sort from the Sort column to define the sorting manner of groups for the selected group level.

Special Group

Specifies how to group your information. Activated only when you have clicked Special Group from the Sort column to define a special group.

Select N

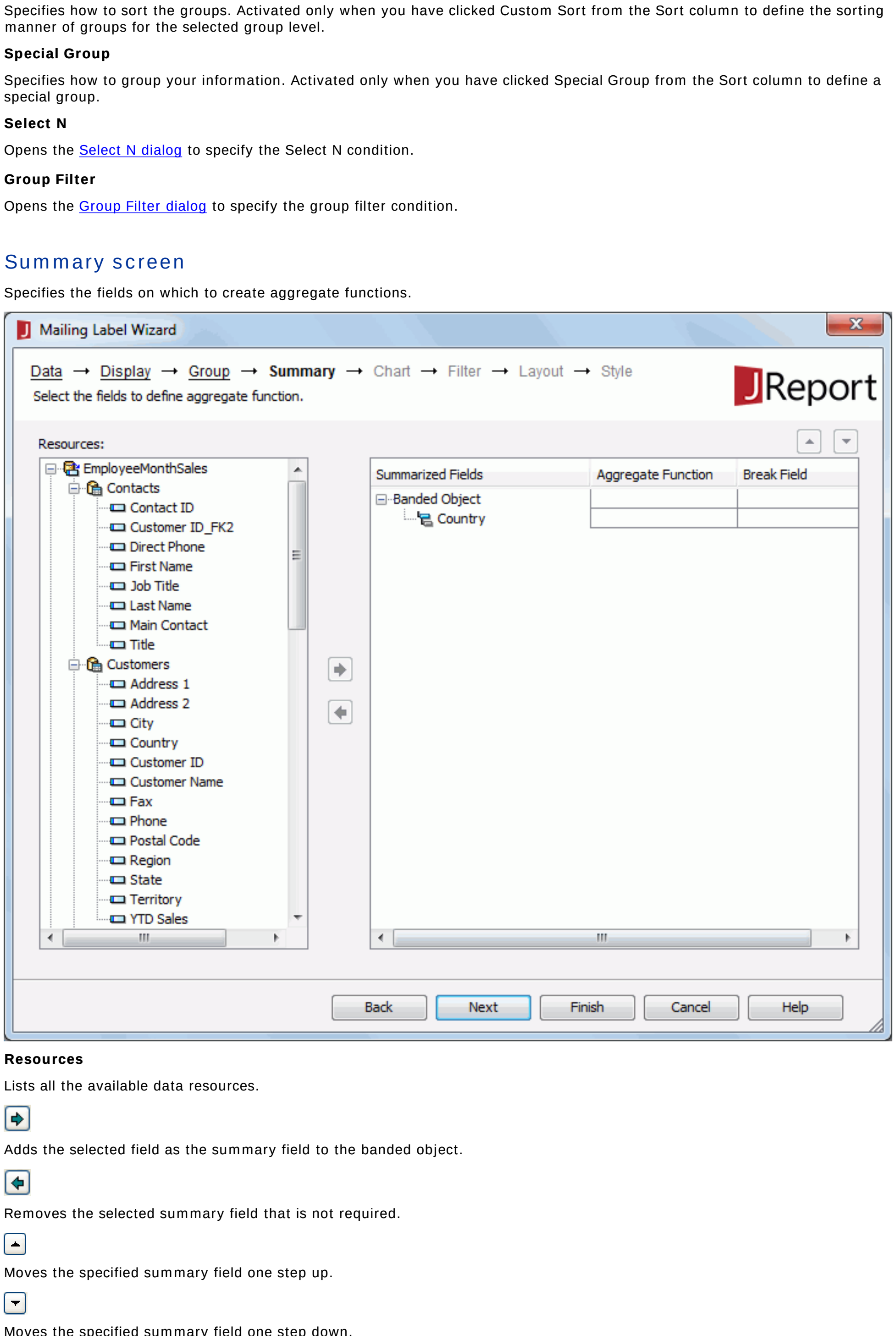
Opens the Select N dialog to specify the Select N condition.

Group Filter

Opens the Group Filter dialog to specify the group filter condition.

Summary screen

Specifies the fields on which to create aggregate functions.



Resources

Lists all the available data resources.

Adds the selected field as the summary field to the banded object.

Removes the selected summary field that is not required.

Moves the specified summary field one step up.

Moves the specified summary field one step down.

Summarized Fields

Lists all the fields that you want to display and to be summarized in the banded object.

Aggregate Function

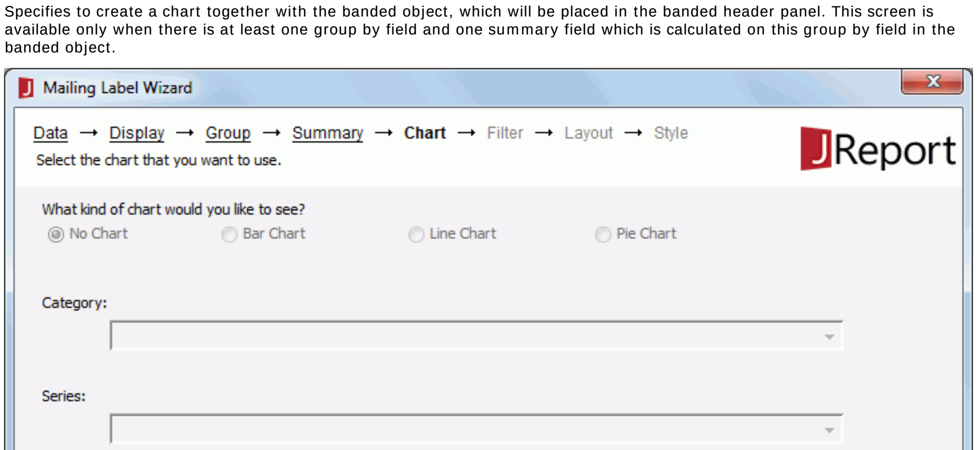
Specifies the function to summarize the selected field.

Break Field

Displays a defined field on which the summary will be calculated. If a summary field is added below the Banded Object node, the break field is null and the summary will be calculated on the whole dataset.

Chart screen

Specifies to create a chart together with the banded object, which will be placed in the banded header panel. This screen is available only when there is at least one group by field and one summary field which is calculated on this group by field in the banded object.



No Chart

Specifies not to create a chart.

Bar Chart

Specifies to create a Clustered Bar 2-D chart together with the banded object.

Line Chart

Specifies to create a Line 2-D chart together with the banded object.

Pie Chart

Specifies to create a Clustered Pie chart together with the banded object.

Category

Lists the group by fields of the banded object on which the summary fields are calculated. Choose the field you want to display on the category (X) axis of the chart from the drop-down list.

Series

Lists the fields that have been added as the group by fields of the banded object. Choose the field you want to display on the series (Z) axis of the chart from the drop-down list.

Show Values

Lists the summary fields which are calculated based on the field you choose to display on the category axis of the chart. Choose the value you want to display in the chart from the drop-down list.

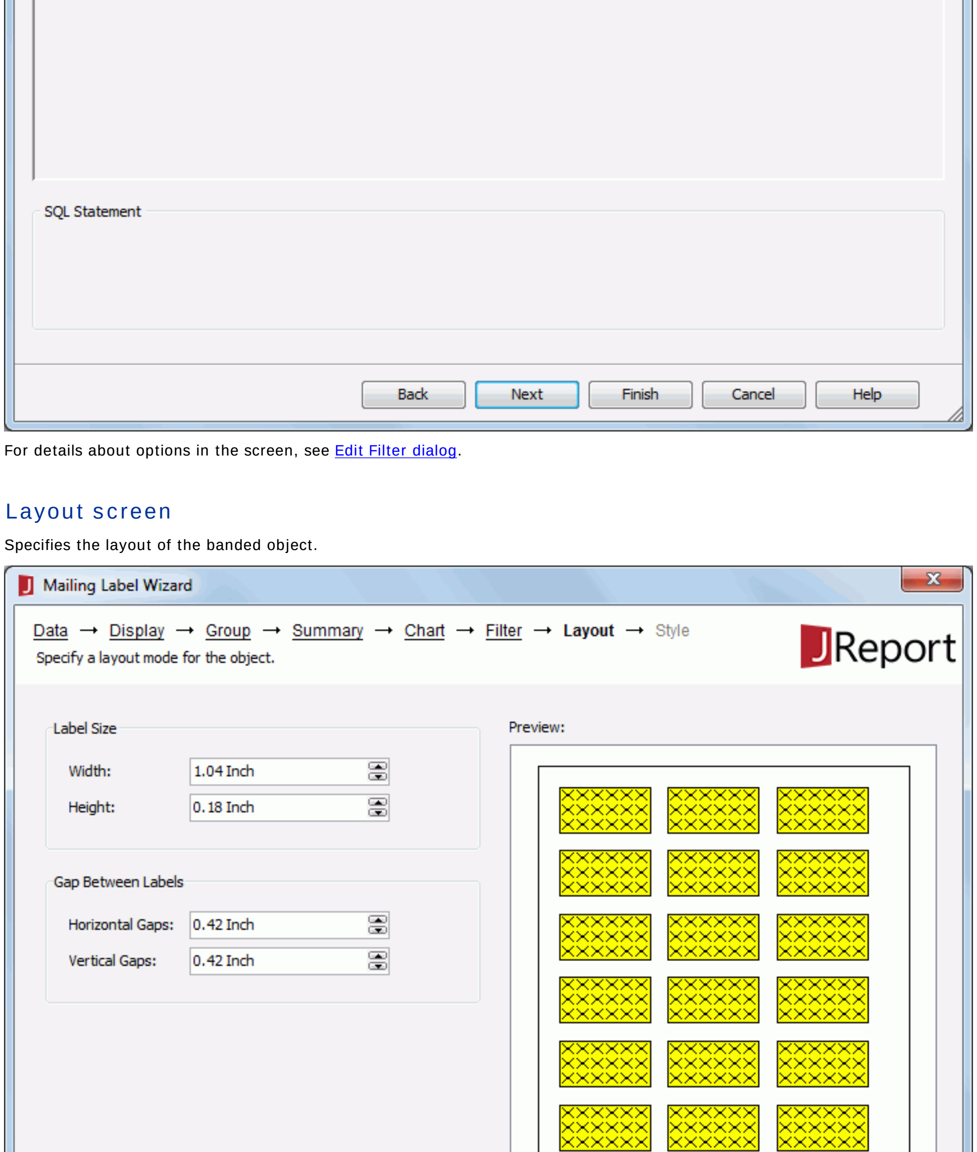
Filter screen

Specifies to filter data displayed in the banded object.

For details about options in the screen, see Edit Filter dialog.

Layout screen

Specifies the layout of the banded object.



Label Size

Specifies the size for the labels in the banded object.

- **Width**
  Specifies the width of the labels, in inches. By default, its value is 1.04.
- **Height**
  Specifies the height of the labels, in inches. By default, its value is 0.18.

Gap Between Labels

Specifies the gap between each two labels.

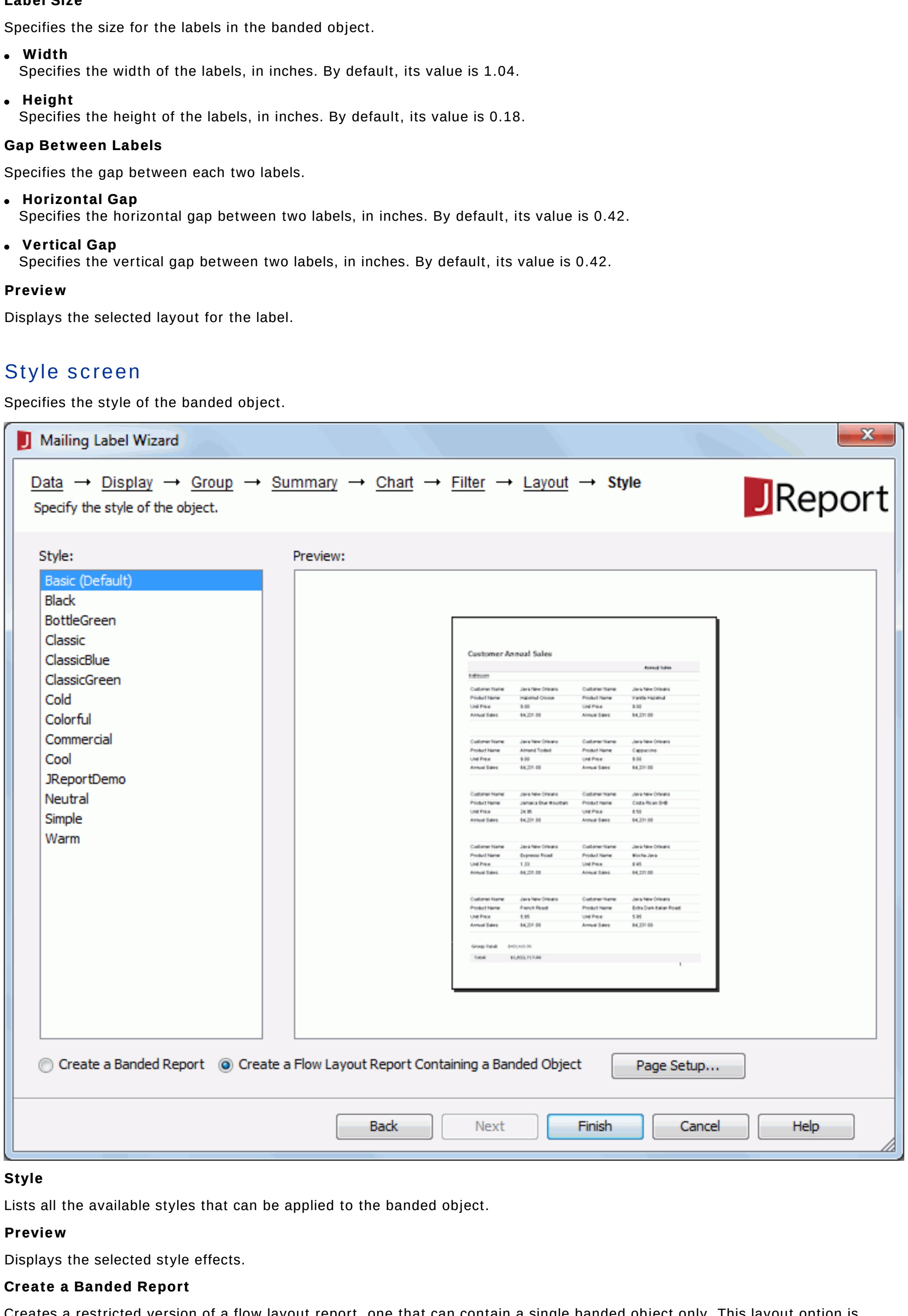
- **Horizontal Gap**
  Specifies the horizontal gap between two labels, in inches. By default, its value is 0.42.
- **Vertical Gap**
  Specifies the vertical gap between two labels, in inches. By default, its value is 0.42.

Preview

Displays the selected layout for the label.

Style screen

Specifies the style of the banded object.



Style

Lists all the available styles that can be applied to the banded object.

Preview

Displays the selected style effects.

Create a Banded Report

Creates a restricted version of a flow layout report, one that can contain a single banded object only. This layout option is intended to provide compatibility with JReport Version 7 report templates.

Create a Flow Layout Report Containing a Banded Object

Creates a report that contains a banded object mapped to the specified dataset. This layout option is specified by default.

Page Setup

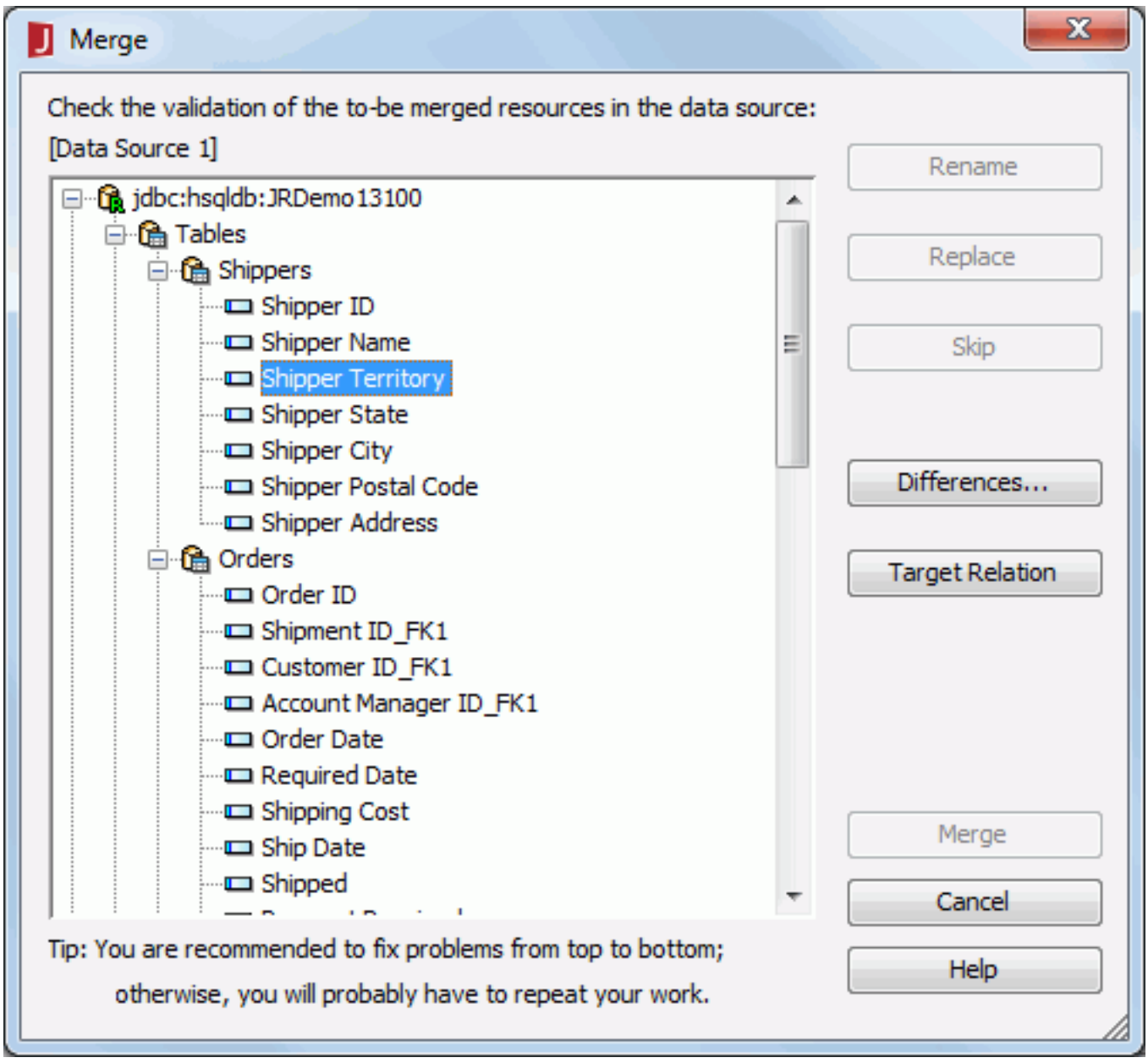
Opens the Page Setup dialog to specify page properties.

Related topics:

- Creating a banded report

Merge dialog

This dialog appears when you click Save in the Save To dialog to save your report in another directory in which a catalog of the same name exists. It helps you to merge the two catalogs.



The following are details about options in the dialog:

Resource to be merged box

Displays all the resources of the report with conflicting resources marked.

Rename

Renames the selected conflicting resource and copy it to the target catalog with a new name.

Replace

Replaces the resource in the target catalog with the selected conflicting resource from the source catalog.

Skip

Leaves the original value in the target catalog.

Differences

Displays the Properties Differences dialog to show the property values of the selected conflicting resource in the two catalogs. The different properties are highlighted.

Note: Sometimes, even though some objects are marked in the Merge dialog, in the Properties Differences dialog, you will not find different properties. The reason for this is that some children of the two objects are different.

Target Relation

Shows the name of the resource and its parent node in the target catalog in the Target Relation dialog.

Merge

Closes the dialog and merges the two catalogs according to the changes that you have made.

Cancel

Closes the dialog, leaving any changes unsaved.

Help

Displays the help document.

Related topics:

- Merging catalogs

MongoDB Connection Wizard

This wizard appears when you select MongoDB and click OK in the New Data Source dialog, or in the Catalog Browser, right-click a data source and select Add MongoDB Connection from the shortcut menu, or right-click an existing MongoDB connection and select Edit Connection from the shortcut menu. It helps you to import MongoDB collection schemas and transform them to relational schemas for future use, and consists of the following screens:

- MongoDB Connection Information screen
- Connection Options screen
- Add Database screen
- Add Schema screen
- Add Table screen

Back

Goes back to the previous screen.

Next

Goes to the next screen.

Finish

Finishes confirming to transform MongoDB schemas to relational schemas.

Cancel

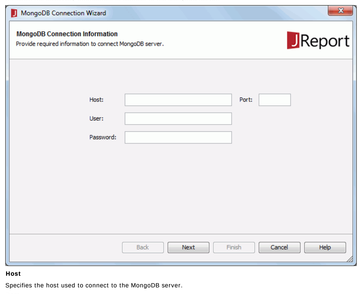
Does not retain any changes and closes this wizard.

Help

Displays the help document about this feature.

MongoDB Connection Information screen

Specifies the necessary information to connect to the MongoDB server.



Host

Specifies the host used to connect to the MongoDB server.

Port

Specifies the port number used to connect to the MongoDB server.

User

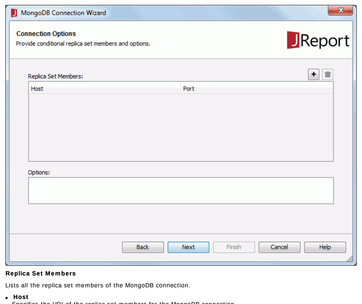
Specifies the user ID used to connect to the MongoDB server.

Password

Specifies the password of the user used to connect to the MongoDB server.

Connection Options screen

Specifies conditional replica set members and options for the MongoDB connection.



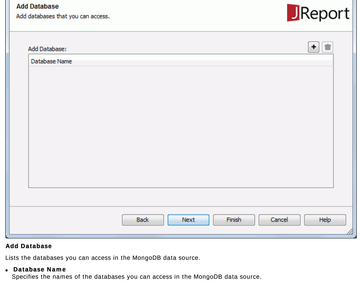
Replica Set Members

Lists all the replica set members of the MongoDB connection.

- **Host**
  Specifies the URI of the replica set members for the MongoDB connection.
- **Port**
  Specifies the port number of the members.
- Adds a new replica set member.
- Removes the selected replica set member.
- **Options**
  Specifies the options of the MongoDB server. The options are name=value pairs and the pairs are separated by "&". What you can specify are listed as follows.
  - **slaveOk=true|false**
    If true, the driver connected to a replica set will send reads to slaves/secondaries, or you can also read a slave/secondary database by adding its URI and port to the Replica Set Members box.
  - **safe=true|false**
    If true, the driver sends a getLastError command after every update to ensure that the update succeed.
  - **w=n**
    The driver adds { w : n } to the getLastError command. Implies safe=true.
  - **wtimeoutMS=ms**
    The driver adds { wtimeout : ms } to the getLastError command. Implies safe=true.
  - **fsync=true|false**
    If true, the driver adds { fsync : true } to the getLastError command. Implies safe=true.
  - **journal=true|false**
    If true, sync to journal. Implies safe=true.
  - **connectTimeoutMS=ms**
    How long a connection can take to be opened before timing out.
  - **socketTimeoutMS=ms**
    How long a send or receive on a socket can take before timing out.

Add Database screen

Adds the databases you can access to the MongoDB connection. This screen will be displayed only when you have no right to access all the databases in the MongoDB data source.



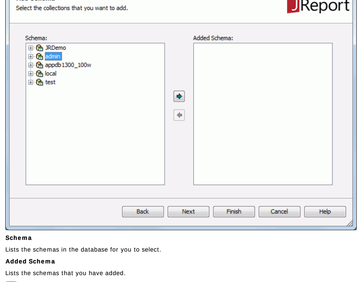
Add Database

Lists the databases you can access in the MongoDB data source.

- **Database Name**
  Specifies the names of the databases you can access in the MongoDB data source.
- Adds a new database you can access.
- Removes the selected database.

Add Schema screen

Adds the collection schemas that will be transformed to relational schemas. This screen is not available when you edit an existing MongoDB connection.



Schema

Lists the schemas in the database for you to select.

Added Schema

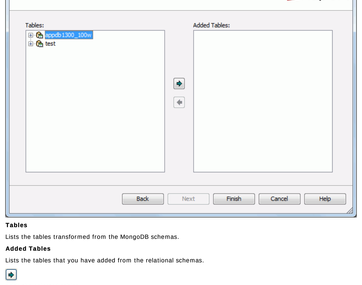
Lists the schemas that you have added.

Adds the selected schemas.

Removes the selected schemas from the Added Schema box.

Add Table screen

Adds tables that are transformed from the relational schemas to the connection. This screen is not available when you edit an existing MongoDB connection.



Tables

Lists the tables transformed from the MongoDB schemas.

Added Tables

Lists the tables that you have added from the relational schemas.

Adds the selected tables.

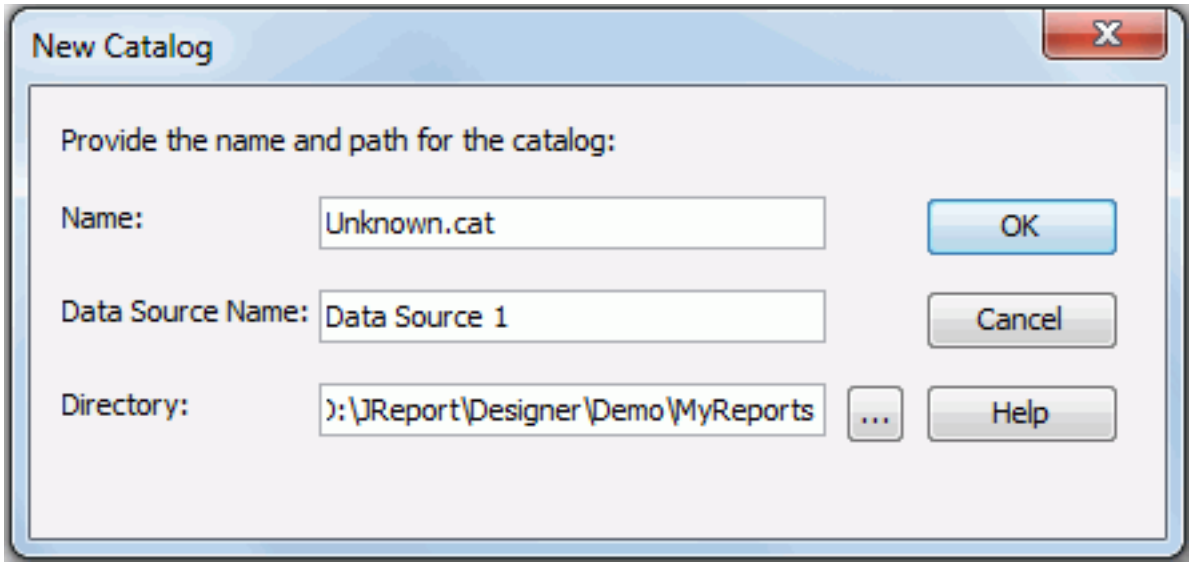
Removes the selected tables from the Added Tables box.

Related topics:

- MongoDB connections

New Catalog dialog

This dialog appears when you click File > Catalog Management > New Catalog on the menu bar. It helps you to create a catalog in JReport Designer.



The following are details about options in the dialog:

Name

Specifies the name of the catalog file. Input the name directly in the text box, and do not forget to add the extension .cat or .xml.

Data Source Name

Specifies the name of the data source in the catalog. Input the name directly in the text box, and make sure the name you give follows the Java class naming rule.

Directory

Specifies the location to save the new catalog. Type the directory of the catalog in the text box or click the button ... to find a directory path.

OK

Accepts all changes and closes the dialog.

Cancel

Closes the dialog, leaving any changes unsaved.

Help

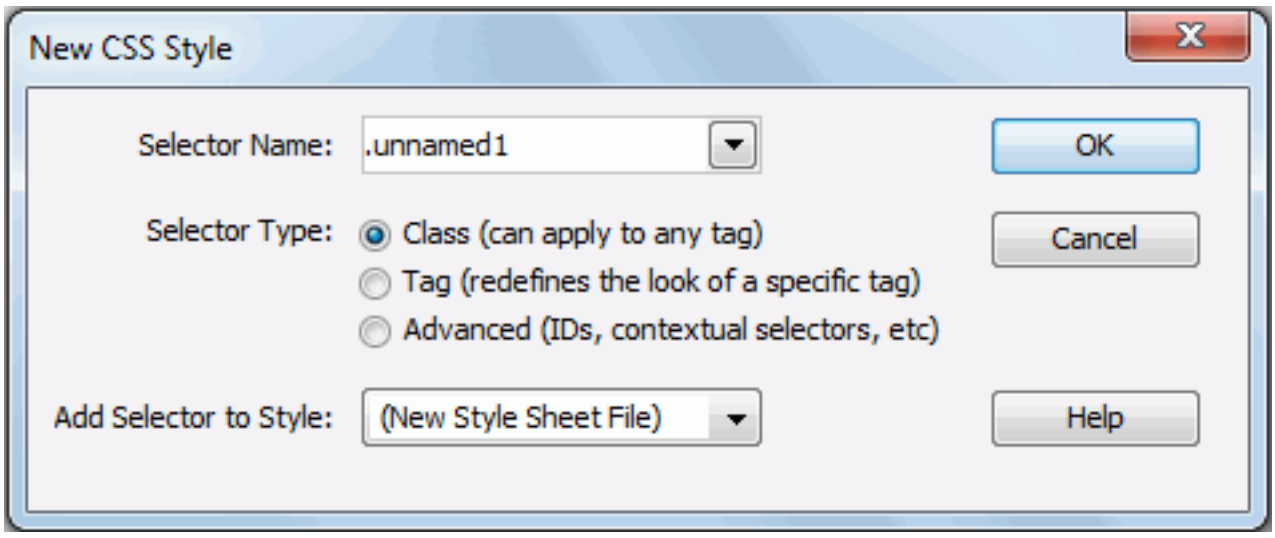
Displays the help document about this feature.

Related topics:

- Creating a catalog

New CSS Style dialog

This dialog appears when you right-click an object in the design area and select Save Style from the shortcut menu, or click the New button in the CSS Editor dialog. It helps you to create a CSS style for a specified object in a report.



The following are details about options in the dialog:

Selector Name

Specifies the name of the selector.

Selector Type

Specifies the type of the CSS style you want to create.

- **Class (can apply to any tag)**
  Specifies to create a custom style that can be applied as a class attribute to a range or block of text.

  Note: Class names must begin with a period and can contain any combination of letters and numbers (for example, .myhead1). If you don't enter a beginning period, JReport Designer will automatically enter it for you.

- **Tag (redefines the look of a specific tag)**
  Specifies to redefine the default formatting of a specific tag.

- **Advanced (IDs, contextual selector, and so on)**
  Specifies to define the formatting of a particular combination of tags or for all tags that contain a specific ID attribute.

Add Selector to Style

Specifies the location into which the selector will be added.

- **New Style Sheet File**
  Specifies to save the new style into a new CSS file.

- **Other existing CSS files**
  Specifies to save the new style into the specified CSS file.

OK

- If New Style Sheet File is selected, clicking this button will bring out the Save CSS As dialog.

- If any of the existing CSS files is selected, clicking this button will bring out the CSS Style Definition dialog.

Cancel

Closes the dialog, leaving any changes unsaved.

Help

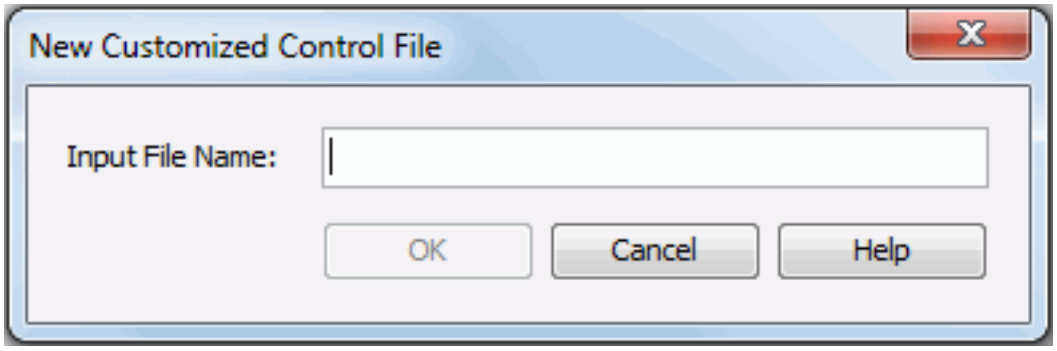
Displays the help document about this feature.

Related topics:

- Creating a CSS style

New Customized Control File dialog

This dialog appears when you click Save As in the Customized Control - Web Action Builder dialog, or click + in the Customized Control Manager dialog. It helps you to create a file in the customized control library to save the customized control definition into.



The following are details about options in the dialog:

Input File Name

Input the name of the file, with or without the appendix .wctrl, to save the customized control definition into.

OK

- Saves the specified file to the customized control library and exits the current dialog if the dialog is opened after you click Save As in the Customized Control - Web Action Builder dialog.
- Exits the current dialog and displays the Customized Control - Web Action Builder dialog for defining a customized control in the specified file if the dialog is opened after you click + in the Customized Control Manager dialog.

Cancel

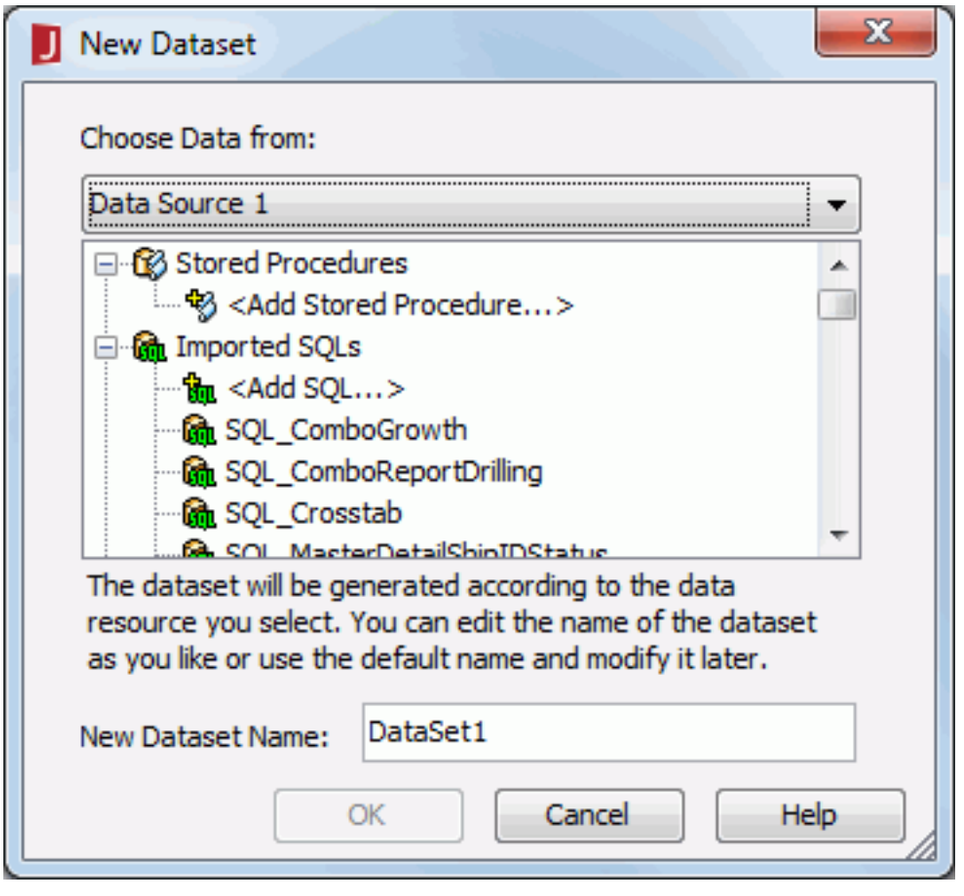
Cancels the settings and exits this dialog.

Help

Displays the help document about this feature.

New Dataset dialog

This dialog appears when you click the <Add Dataset...> item in the Data tab of the report wizard, click the <Choose Data From...> item from the dataset drop-down list in the Resource View panel for a page report, or click the Add button in the Dataset Management dialog. It helps you to create a dataset for a report.



The following are details about options in the dialog:

Choose Data from

Specifies the data source for the dataset. Data sources that exist in the current catalog will be listed here.

New Dataset Name

Specifies the name of the dataset.

OK

Accepts changes and closes this dialog.

Cancel

Does not retain changes and closes this dialog.

Help

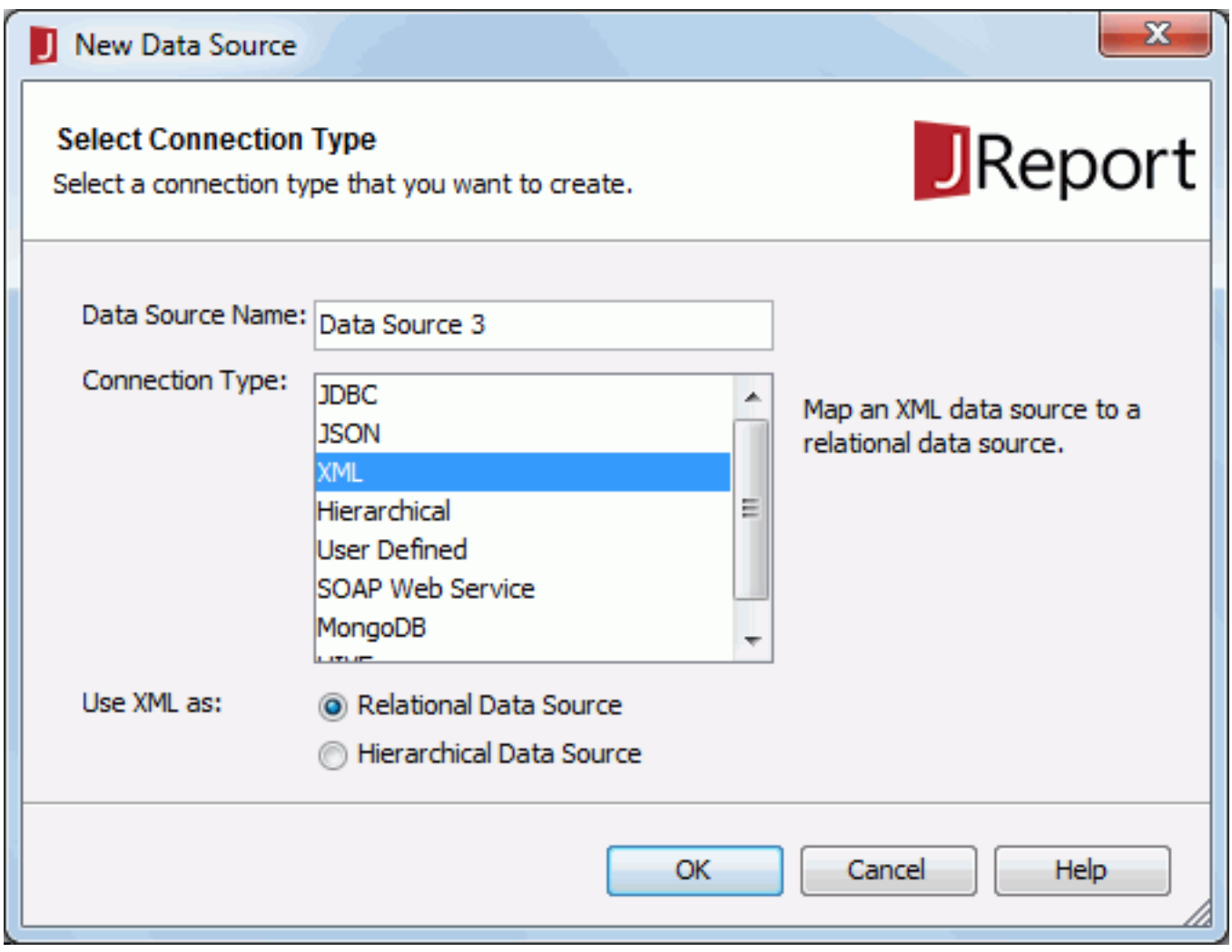
Displays the help document.

Related topics:

- Creating a dataset

New Data Source dialog

This dialog appears when you click File > New > Data Source on the menu bar, or select a data source and click the New Data Source button on the Catalog Browser toolbar. It helps you to create different data sources by selecting a connection type: JDBC, XML, Hierarchical, User Defined, Web Service, MongoDB and HIVE.



The following are details about options in the dialog:

Data Source Name

Specifies the name of the new data source. Input the name directly in the text box, and make sure to follow the Java class naming rule.

Connection Type

Specifies the connection type for the new data source:

- **JDBC**
  If selected, a JDBC data source will be created and data stored in RDBMS can be accessed via JDBC connection or JDBC-ODBC bridge.
- **JSON**
  If selected, a JSON data source will be created.
- **XML**
  If selected, an XML data source will be created and data stored in XML instance can be accessed via XML data transformation.
  - **Use XML as**
    Specifies to create an XML data source as a relational one or a hierarchical one.
    - **Relational Data Source**
      Specifies to create an XML data source by transforming an XML instance.
    - **Hierarchical Data Source**
      Specifies to import an XML hierarchical data source.
- **Hierarchical**
  If selected, a new hierarchical data source will be created.
- **User Defined**
  If selected, a new user defined data source will be created.
- **SOAP Web Service**
  If selected, a new SOAP web service data source will be created and the data can be retrieved through a SOAP web service.
- **MongoDB**
  If selected, a new MongoDB data source will be created and data stored in the data source can be accessed via data transformation.
- **HIVE**
  If selected, a new HIVE data source will be created and data stored in the data source can be accessed via a JDBC connection.

OK

Applies the changes and closes the dialog.

Cancel

Cancels the changes and exits the dialog.

Help

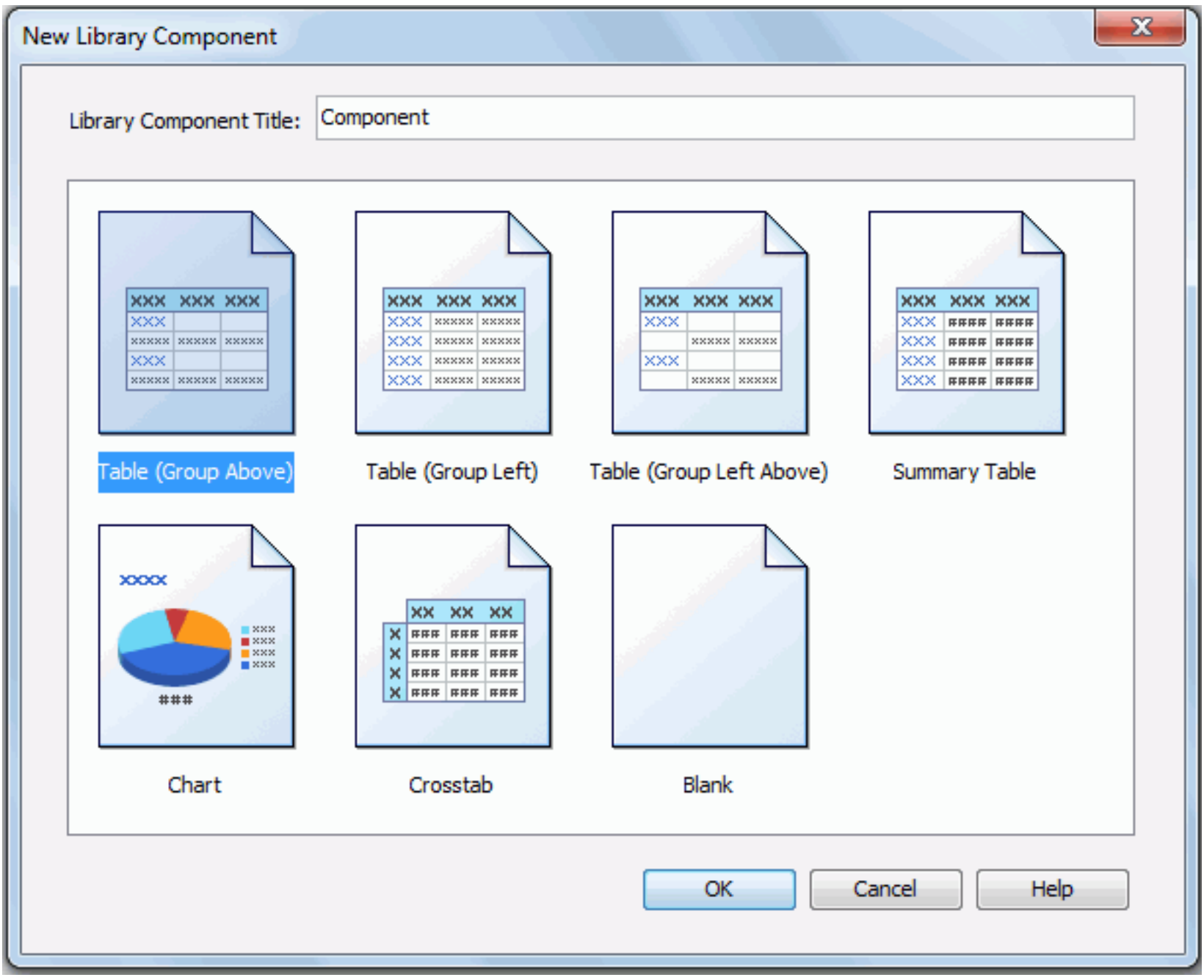
Displays the help document about this feature.

Related topics:

- Data Source Connections

New Library Component dialog

This dialog appears when you click File > New > Library Component on the menu bar. It helps you to create a library component.



The following are details about options in the dialog:

Library Component Title

Specifies the name of the library component.

Component layout

Specifies the layout with which to create the library component.

- **Table (Group Above)**
  Creates a library component containing a table with group information above the detail panel.
- **Table (Group Left)**
  Creates a library component containing a table with group information left to the detail panel.
- **Table (Group Left Above)**
  Creates a library component containing a table with group information left above the detail panel.
- **Summary Table**
  Creates a library component containing a table with only group and summary information.
- **Chart**
  Creates a library component containing a chart.
- **Crosstab**
  Creates a library component containing a crosstab.
- **Blank**
  Creates a library component with nothing in it.

OK

Creates a library component with the selected layout.

Cancel

Closes the dialog and deletes any unsaved changes.

Help

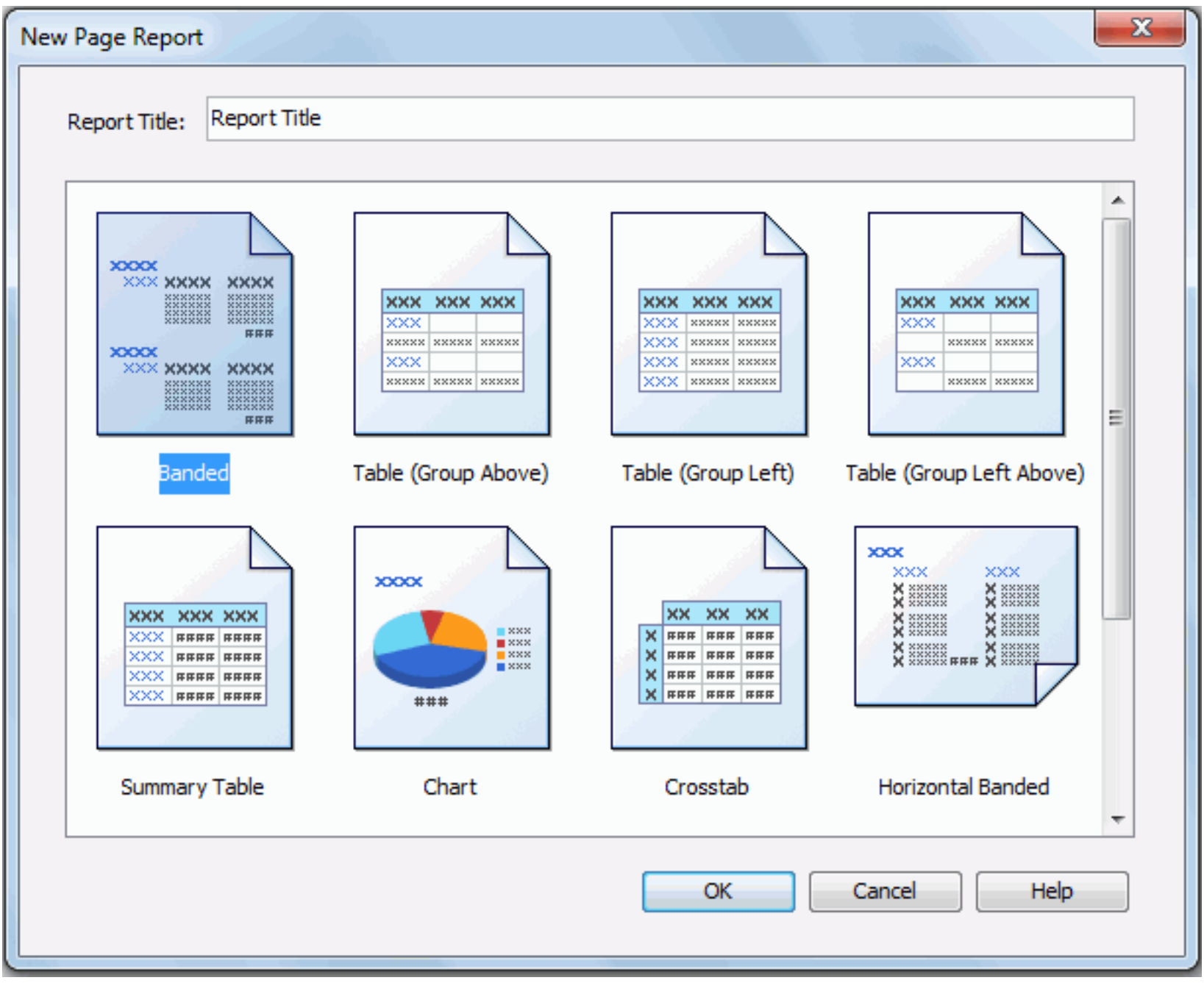
Displays the help document about this feature.

Related topics:

- Creating a library component

New Page Report dialog

This dialog appears when you click File > New > Page Report on the menu bar, or click the New Page Report button on the Standard toolbar. You use this dialog to create a report. Reports cannot be empty; therefore this dialog also creates the first report tab in the new report.



The following are details about options in the dialog:

Report Title

Specifies the name of the new report.

Report layout

Specifies the layout with which to create the report.

- **Banded**
  Creates a report containing a vertical banded object.
- **Table (Group Above)**
  Creates a report containing a table with group information above the detail panel.
- **Table (Group Left)**
  Creates a report containing a table with group information left to the detail panel.
- **Table (Group Left Above)**
  Creates a report containing a table with group information left above the detail panel.
- **Summary Table**
  Creates a report containing a table with only group and summary information.
- **Chart**
  Creates a report containing a chart.
- **Crosstab**
  Creates a report containing a crosstab.
- **Horizontal Banded**
  Creates a report containing a horizontal banded object.
- **Mailing Label**
  Creates a report containing a banded object in the form of a mailing label layout.
- **Tabular**
  Creates a report containing a tabular component.
- **Blank**
  Creates a report with nothing in it.

OK

Creates a report containing its first report tab with the selected layout.

Cancel

Closes the dialog and deletes any unsaved changes.

Help

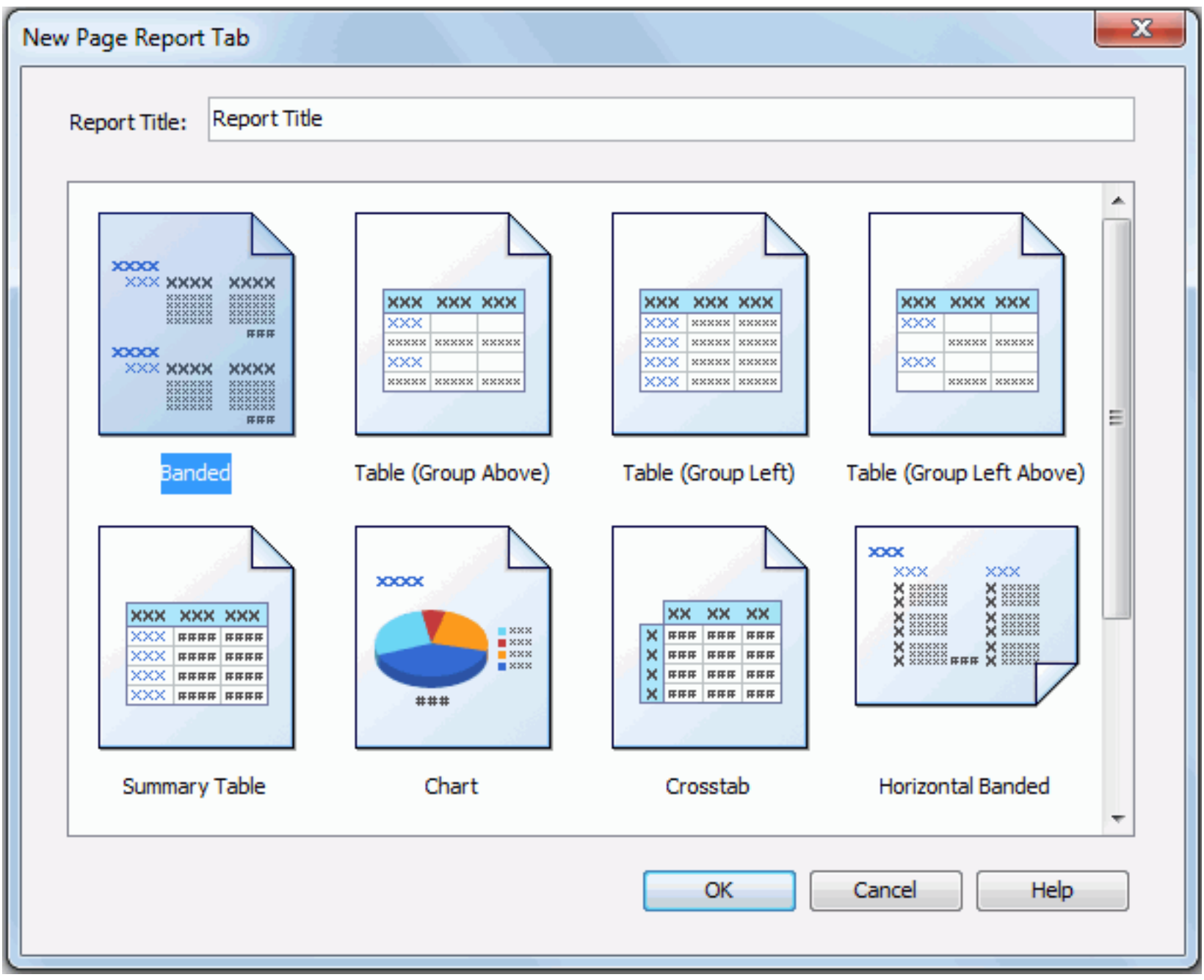
Displays the help document about this feature.

Related topics:

- Creating a page report

New Page Report Tab dialog

This dialog appears when you click File > New > Page Report Tab on the menu bar, or on the report tab bar, right-click an existing report tab and click Insert on the shortcut menu. It helps you to create a report tab in an existing report.



The following are details about options in the dialog:

Report Title

Specifies the name of the new report tab.

Report layout

Specifies the layout with which to create the report tab.

- **Banded**
  Creates a report containing a vertical banded object.
- **Table (Group Above)**
  Creates a report containing a table with group information above the detail panel.
- **Table (Group Left)**
  Creates a report containing a table with group information left to the detail panel.
- **Table (Group Left Above)**
  Creates a report containing a table with group information left above the detail panel.
- **Summary Table**
  Creates a report containing a table with only group and summary information.
- **Chart**
  Creates a report containing a chart.
- **Crosstab**
  Creates a report containing a crosstab.
- **Horizontal Banded**
  Creates a report containing a horizontal banded object.
- **Mailing Label**
  Creates a report containing a banded object in the form of a mailing label layout.
- **Tabular**
  Creates a report containing a tabular component.
- **Blank**
  Creates a report with nothing in it.

OK

Creates a report tab with the selected layout.

Cancel

Closes the dialog and deletes any unsaved changes.

Help

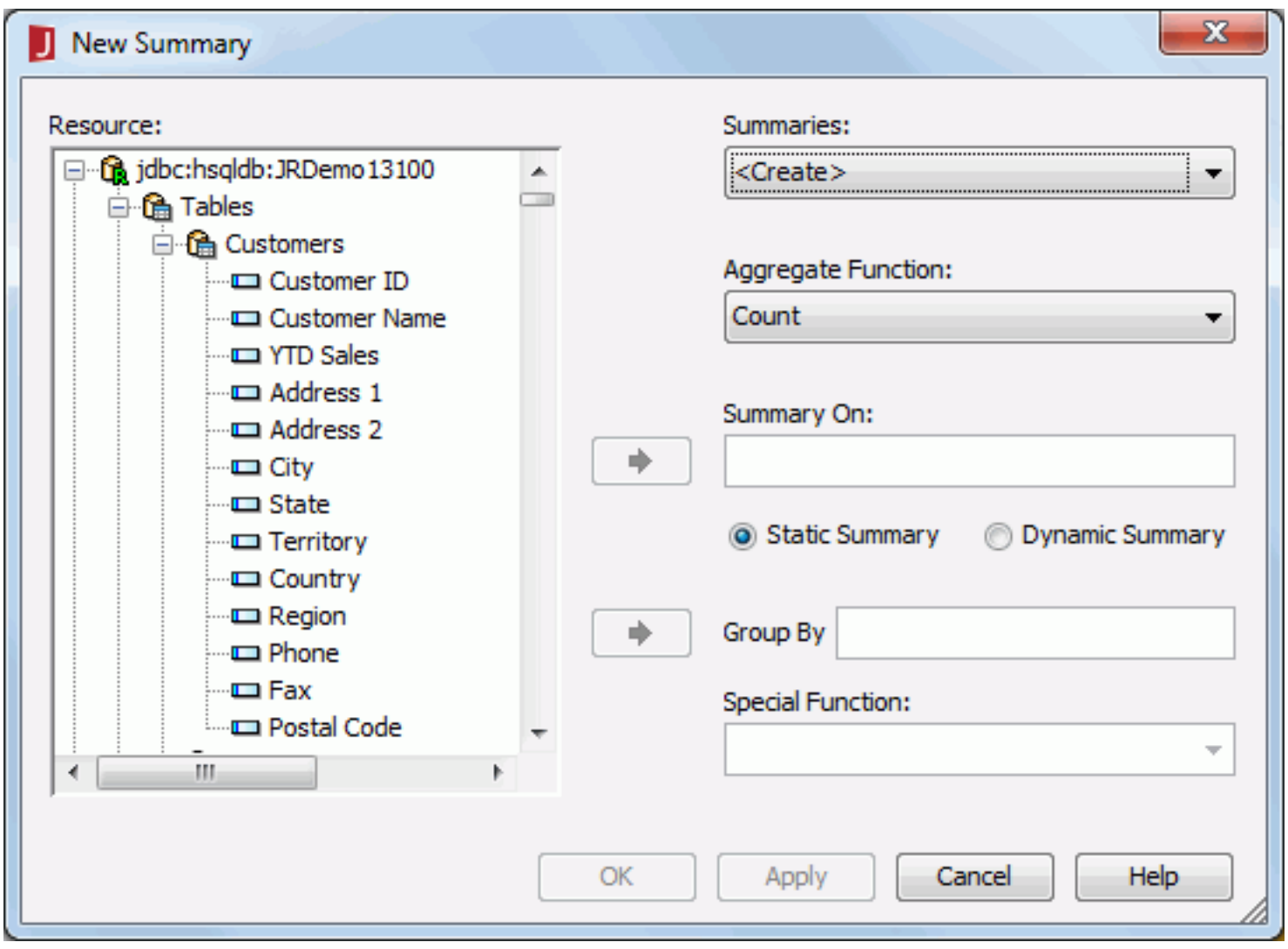
Displays the help document about this feature.

Related topics:

- Creating report tabs in a page report

New Summary dialog

This dialog appears when you right-click the Summaries node in the Data tab of the Catalog Browser and select Add Summary from the shortcut menu. It helps you to create a summary in a catalog.



The following are details about options in the dialog:

Resource

Lists all the available resources that can be used to make a summary.

Summaries

By default, <Create...> is selected in the drop-down list. You can also select one of the existing summaries from the drop-down list for modifying it.

Aggregate Function

Lists the functions which are available to the selected field. For details about each function, see Math functions.

Summary On

Specifies the field on which the summary is to compute.

To specify the sum on field, highlight it in the Resource box and then click next to Summary On to add it to the field box.

Static Summary

If selected, the summary will take effect on a specific group that you define in the Group By field and it can only be inserted into the group.

- **Group By**
  Specifies the field used to group the data.

  To specify the group by field, highlight it in the Resource box and then click next to Group By to add it to the field box.

Dynamic Summary

If selected, the summary will take effect on the group according to the number, which is specified in the Group By field and it can be inserted into any groups of a component.

- **Group By**
  Specifies on which group the summary will take effect.

  To specify the group level, first select Up or Down from the drop-down list, then in the combo box, select a value or input an integer, which should be between 0 to 127, to specify the group on which the summary will take effect. For example, if you select Up or Down and input 0 in the combo box and insert this summary into any group, the summary will take effect on this group; if you select Up and input 1 and insert this summary to a group, the summary will take effect on the group higher than the group where the summary is inserted, and if you specify Down and 1, it will take effect on the lower group; and the rest may be deduced by analogy.

  Notes:

  - If the dynamic summary is to be used in a chart, when defining the summary, you must choose Down from the Group By drop-down list; if it is to be inserted to other data containers, the summary must be defined on Up level.
  - When you insert a dynamic summary into a report and there is no corresponding group level to match settings of the dynamic summary, an error message will be generated.
  - If you want to insert a summary with a special function into a table or banded object, there must be a group with the same special function in the table or banded object.

Special Function

If the group by field is of Numeric/String/Date/Time type, you can select a special function for the field in the Special Function drop-down list to further specify to which level the data will be grouped by. If Customize is selected, the Customized Function dialog will be displayed, in which you can set the function by your own.

OK

Accepts all changes and closes the dialog.

Apply

Applies all changes to the existing summary or adds the new summary to the catalog.

Cancel

Does not retain any changes and closes this dialog.

Help

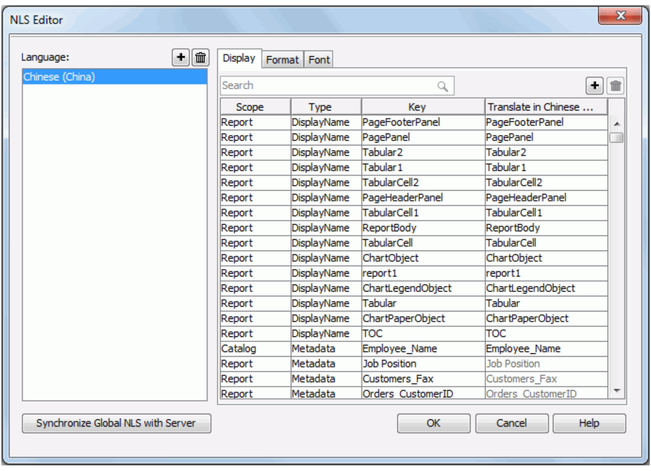
Displays the help document about this feature.

Related topics:

- Creating a summary

NLS Editor dialog

This dialog appears when you click Edit > NLS Editor on the menu bar, or the NLS Editor button on the NLS Language toolbar. It helps you to edit NLS for the current report and catalog.



The following are details about options in the dialog:

Language

Displays the languages into which the display text in the current report and catalog will be translated.

Opens the Add Language dialog to add languages for editing here.

Removes the selected language from the Language box.

Display tab

Specifies the translation of all the display text. You can click the column header in the display table to sort the items by the corresponding column in an ascending or descending order.

- **Quick search toolbar**
  Searches for the items that contain the specified text in the Key and Translate in <language> columns. After typing the text, click the search button on the toolbar or press Enter on the keyboard to start searching.
- Opens the Add Display dialog to add available display text keys for the target language to edit NLS.
- Removes the selected rows of display text for the target language.
- **Scope**
  Lists the scope of the keys.
  - **Global**
    Indicates the key is from global NLS resources.
  - **Catalog**
    Indicates the key is from catalog NLS resouces.
  - **Report**
    Indicates the key is from report NLS resouces.
- **Type**
  Lists types of display text for different objects.
  - **Column**
    This type is only for reports running in Page Report Studio. It is the type of display text of columns.
  - **DisplayName**
    Type of display text of object display name.
  - **Label**
    Type of display text of label, some web controls and UDOs.
  - **Metadata**
    Type of display text of metadata. Metadata mainly refers to catalog resources, such as table/view columns, business views, formulas, summaries, parameters, and so on.
  - **Prompt**
    Type of display text of parameter prompt value.
  - **Title**
    Type of display text of filter control and library component.
  - **TOC**
    Type of display text on the TOC tree.
- **Key**
  Lists keys to indicate the objects in the original language.
- **Translate in <language>**
  Specifies to which the display text will be translated in the target language.

Format tab

Specifies the formats for all fields. You can click the column header in the format table to sort the items by the corresponding column in an ascending or descending order.

- **Quick search toolbar**
  Searches for the items that contain the specified text in the Key and Format in <language> columns. After typing the text, click the search button on the toolbar or press Enter on the keyboard to start searching.
- Opens the Add Format dialog to add available format keys for the target language to edit NLS.
- Removes the selected rows of format for the target language.
- **Scope**
  Lists the scope of the keys.
  - **Global**
    Indicates the key is from global NLS resources.
  - **Catalog**
    Indicates the key is from catalog NLS resouces.
  - **Report**
    Indicates the key is from report NLS resouces.
- **Key**
  Displays formats of all fields in the current report and catalog in the original language.
- **Format in <language>**
  Specifies the format for every field in the current report and catalog in the target language.

Font tab

Specifies font properties for all fields. You can click the column header in the font table to sort the items by the corresponding column in an ascending or descending order.

- **Quick search toolbar**
  Searches for the items that contain the specified text in the Font Face and Font Size columns. After typing the text, click the search button on the toolbar or press Enter on the keyboard to start searching.
- Opens the Add Font dialog to add available font keys for the target language to edit NLS.
- Removes the selected rows of font for the target language.
- **Scope**
  Lists the scope of the keys.
  - **Global**
    Indicates the key is from global NLS resources.
  - **Catalog**
    Indicates the key is from catalog NLS resouces.
  - **Report**
    Indicates the key is from report NLS resouces.
- **Key**
  Lists font face and font size of all fields in the current report and catalog in the original language.
- **Font Face**
  Specifies font face for every field in the current report and catalog in the target language.
- **Font Size**
  Specifies font size for every field in the current report and catalog in the target language.

Synchronize Global NLS with Server

Synchronizes the global NLS resources for the target language with that on JReport Server.

OK

Applies the changes and closes the dialog.

Cancel

Cancels the changes and closes the dialog.

Help

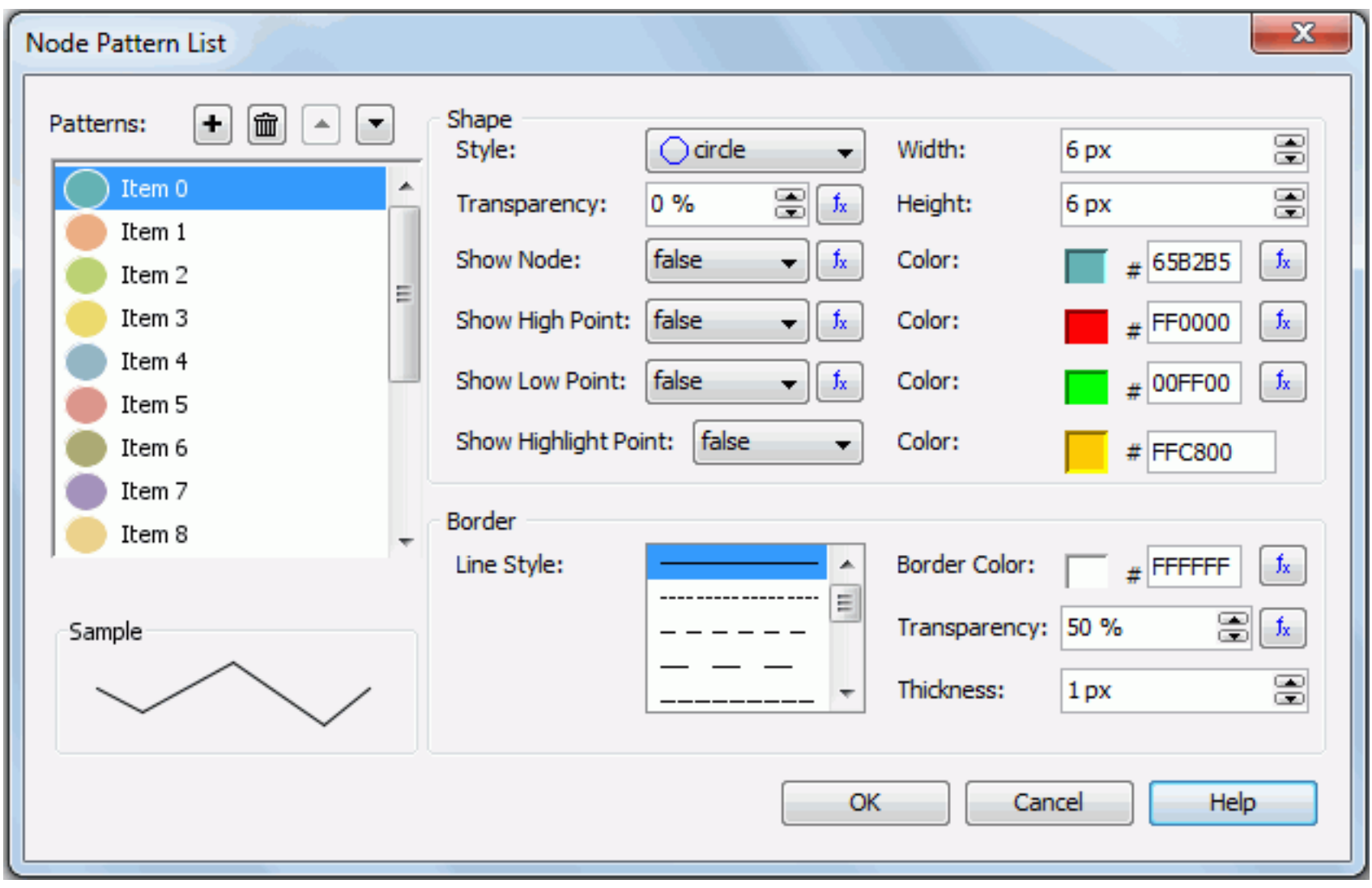
Displays the help document about this feature.

Related topics:

- Creating NLS property files

Node Pattern List dialog

The dialog appears when you click the Node Pattern List button in the Format Line dialog for page report, web report or library component. It helps you to modify the node pattern for each chart line.



The followings are details about options in the dialog:

Patterns

Specifies the pattern for the nodes in each chart line. Click the item in the box one by one to set the node pattern for each line.

- Adds a new pattern item.
- Removes the selected pattern item.
- Moves the selected pattern item one step up.
- Moves the selected pattern item one step down.

Shape

Specifies the shape properties of the nodes in a chart line.

- **Style**
  Specifies the style of the nodes.
- **Transparency**
  Specifies the transparency of the nodes.
- **Width**
  Specifies the width of the nodes.
- **Height**
  Specifies the height of the nodes.
- **Show Node**
  Specifies whether to show nodes in the chart line. Only when it is set to true, the node related properties for the line can take effect. However, it does not control the property Show High/Low Point and the Color property for each one.
  - **Color**
    Specifies the color of the nodes.
- **Show High Point**
  Specifies whether to show the line node in the highest point in the chart line.
  - **Color**
    Specifies the color of the node in the highest point.
- **Show Low Point**
  Specifies whether to show the line node in the lowest point in the chart line.
  - **Color**
    Specifies the color of the node in the lowest point.
- **Show Highlight Point**
  Specifies whether to show the highlight point when you hover the mouse over a node in the selected chart line.
  - **Color**
    Specifies the color of the highlight point in the selected chart line.

Border

Specifies the border properties of the nodes in a chart line. Available when the node is not of the plus, multiplication, star1 or star 2 style.

- **Border Color**
  Specifies the color for the border of the nodes.
- **Transparency**
  Specifies the transparency for the border of the nodes.
- **Line Style**
  Specifies the line style for the border of the nodes.
- **Thickness**
  Specifies the thickness for the border of the nodes.

Line

Specifies the line properties of the nodes in a chart line. Available when the node is of plus, multiplication, star1 or star2 style.

- **Line Style**
  Specifies the line style of the nodes.
- **Thickness**
  Specifies the line thickness of the nodes.

Sample

Displays a preview sample of your selection.

OK

Applies the node patterns and closes the dialog.

Cancel

Does not retain any changes and closes the dialog.

Help

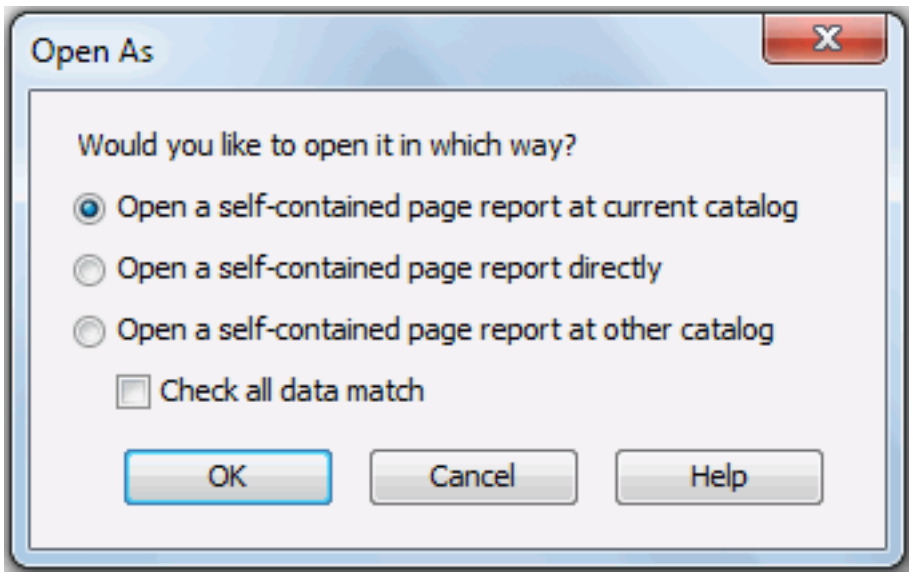
Displays the help document about this feature.

Related topics:

- Formatting the lines

Open As dialog

This dialog appears when you open a self-contained page report from the Open Report dialog (a self-contained page report is a binary report file, which has a .clx suffix and contains not only the report's layout but also the catalog with its own resources). It helps you to specify the way in which the self-contained page report will be opened.



The following are details about options in the dialog:

Open a self-contained page report at current catalog

Specifies to open the selected self-contained page report in the current catalog and make it work under the current catalog.

JReport Designer will compare the resources in the catalog of the self-contained page report with the current catalog according to the checking level (whether the Check all data match is selected). If conflicts exist between the two catalogs, the page report cannot be opened.

Open a self-contained page report directly

Specifies to open the selected self-contained page report directly to make it work under a new catalog.

Open a self-contained page report at other catalog

Specifies to open the selected self-contained page report at another catalog and make it work under that catalog.

If this option is checked, you will be prompted to open an existing catalog for the self-contained page report. JReport Designer will then compare the resources in the catalog of the self-contained page report with the newly selected catalog according to the checking level (whether the Check all data match is selected). If conflicts exist between the two catalogs, the page report cannot be opened.

Check all data match

This checkbox is enabled when Open a self-contained page report at current catalog or Open a self-contained page report at other catalog is checked. It is used to specify the checking level between two catalogs.

- If checked, all the differences between the two catalogs will be checked by JReport Designer.
- If unchecked, only the differences that can cause the report engine to fail when running the page report will be checked by JReport Designer.

OK

Opens the page report in the specified way and closes the dialog.

Cancel

Closes the dialog, leaving any changes unsaved.

Help

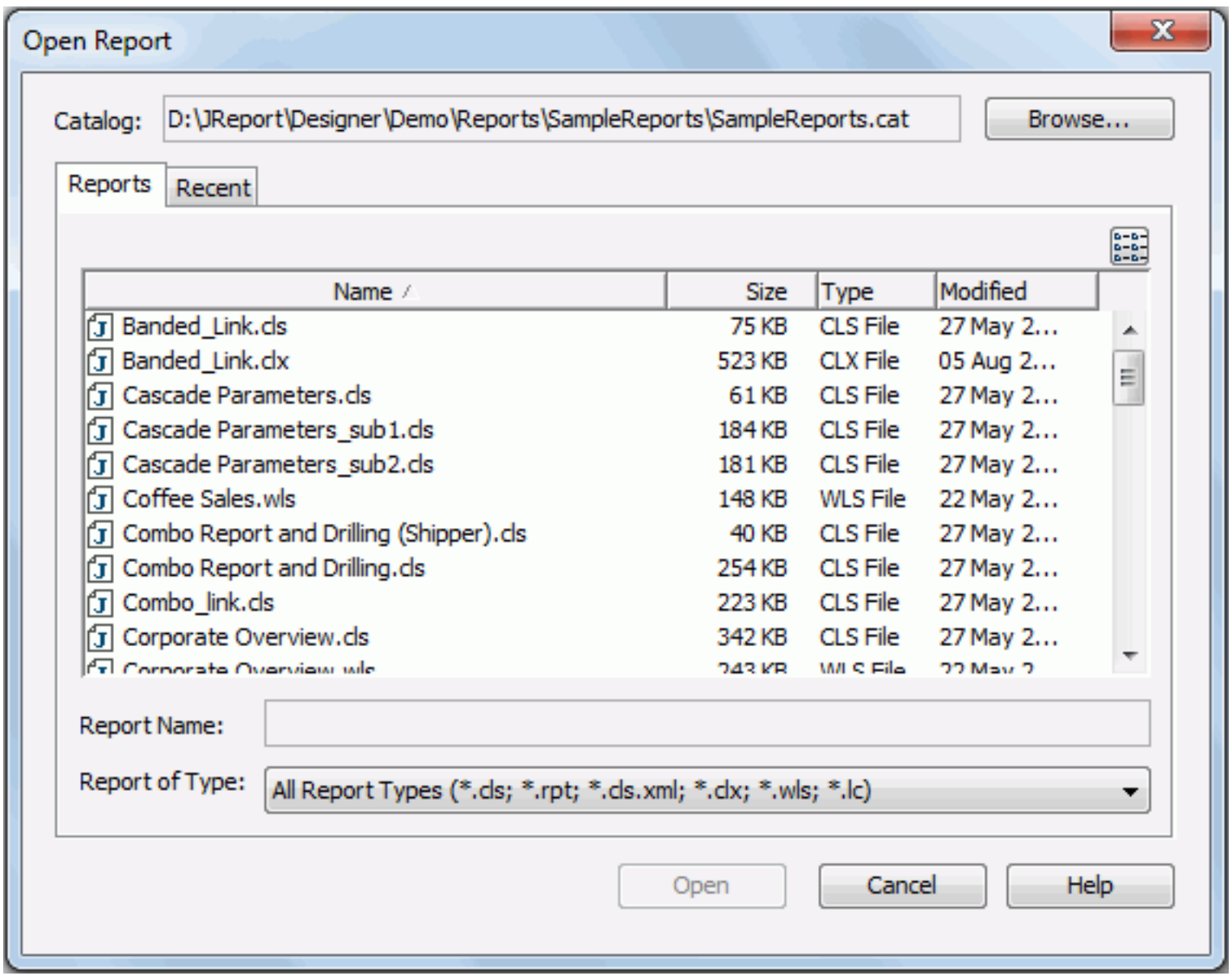
Displays the help document about this feature.

Related topics:

- Opening a page report

Open Report dialog

This dialog appears when you click File > Open or File > Import From > JReport Page Report on the menu bar, or click the Open button on the standard toolbar. It helps you to open a report (page report or web report) or a library component, or import report tabs from existing page reports into current page report.



The following are details about options in the dialog:

Catalog

Shows the directory of the default catalog that is being used now. You can also click the Browse button to specify a new one.

Note: If you are going to import report tabs into current page report, you can only choose from page reports of the same catalog. In this case, the Browse button is disabled.

Reports tab

Specifies one or more reports to open.

- **List**
  Clicks to go to the List mode where only the names of the files are displayed.
- **Details**
  Clicks to go to the Details mode where the files with detailed information such as name, size, type and last modified time are displayed.
- **Name**
  Shows the name of all the reports included in this catalog, based on the type you selected from the Report of Type drop-down list.
- **Size**
  Shows the size of the reports.
- **Type**
  Shows the type of the reports.
- **Modified**
  Shows the time when the reports were modified at the last time.
- **Report Name**
  Shows the names of the selected reports.
- **Report of Type**
  Specifies the type of the reports.

Recent tab

Lists all the reports that have been opened recently.

- **Report Name**
  Shows the names of all the recently opened reports.
- **Catalog Name**
  Shows the name of the catalog that the reports are located in.
- **Report Location**
  Shows the location of all the reports.

Open

Opens the selected reports and closes the dialog. If a self-contained page report is selected, the Open As dialog will appear, asking you to specify the way in which the report will be opened.

- To open consecutive reports, click the first report, press and hold down **SHIFT**, and then click the last report. Then click the **Open** button. The last report will be the active report.
- To open reports that are not consecutive, press and hold down **CTRL**, click each report, and then click the **Open** button. The last selected report will be the active report.

Cancel

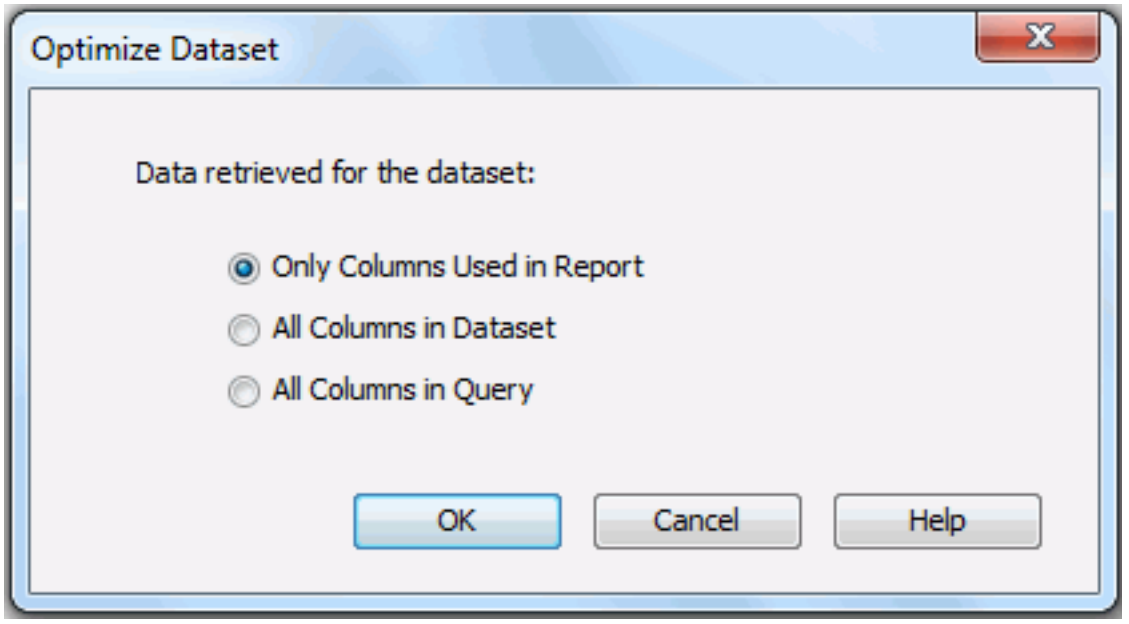
Closes the dialog, leaving any changes unsaved.

Help

Displays the help document about this feature.

Optimize Dataset dialog

This dialog appears when you click the Optimize Dataset button in the Dataset Management dialog. It helps you to optimize the selected dataset.



The following are details about options in this dialog:

Data retrieved for the dataset

Specifies the method used to retrieve data at runtime.

- **Only Columns Used in Report**
  If checked, only data columns used in the current report are retrieved at runtime. This way ensures the best performance since the least data is involved in.
- **All Columns in Dataset**
  If checked, all data columns defined in the dataset are retrieved at runtime.
- **All Columns in Query**
  If checked, all data columns in the query that the dataset is based on are retrieved at runtime. The performance of this way is not as good as the other two.

OK

Applies the settings and closes the dialog.

Cancel

Does not retain the changes and closes the dialog.

Help

Displays the help document about this feature.

Related topics:

- Optimizing a dataset

Options dialog

This dialog appears when you click File > Options on the menu bar, or click the Options button on the Catalog Browser toolbar. It helps you to configure JReport to your preferences and requirements.

The following are details about the options in the dialog:

- General
- Editor
- Panel
- Format
- Dataset
- Component
- Preview
- Catalog
- Export to
- Print
- Query Editor
- Security
- Cache

OK

Applies the changes and closes the dialog.

Cancel

Does not retain the changes and closes the dialog.

Apply

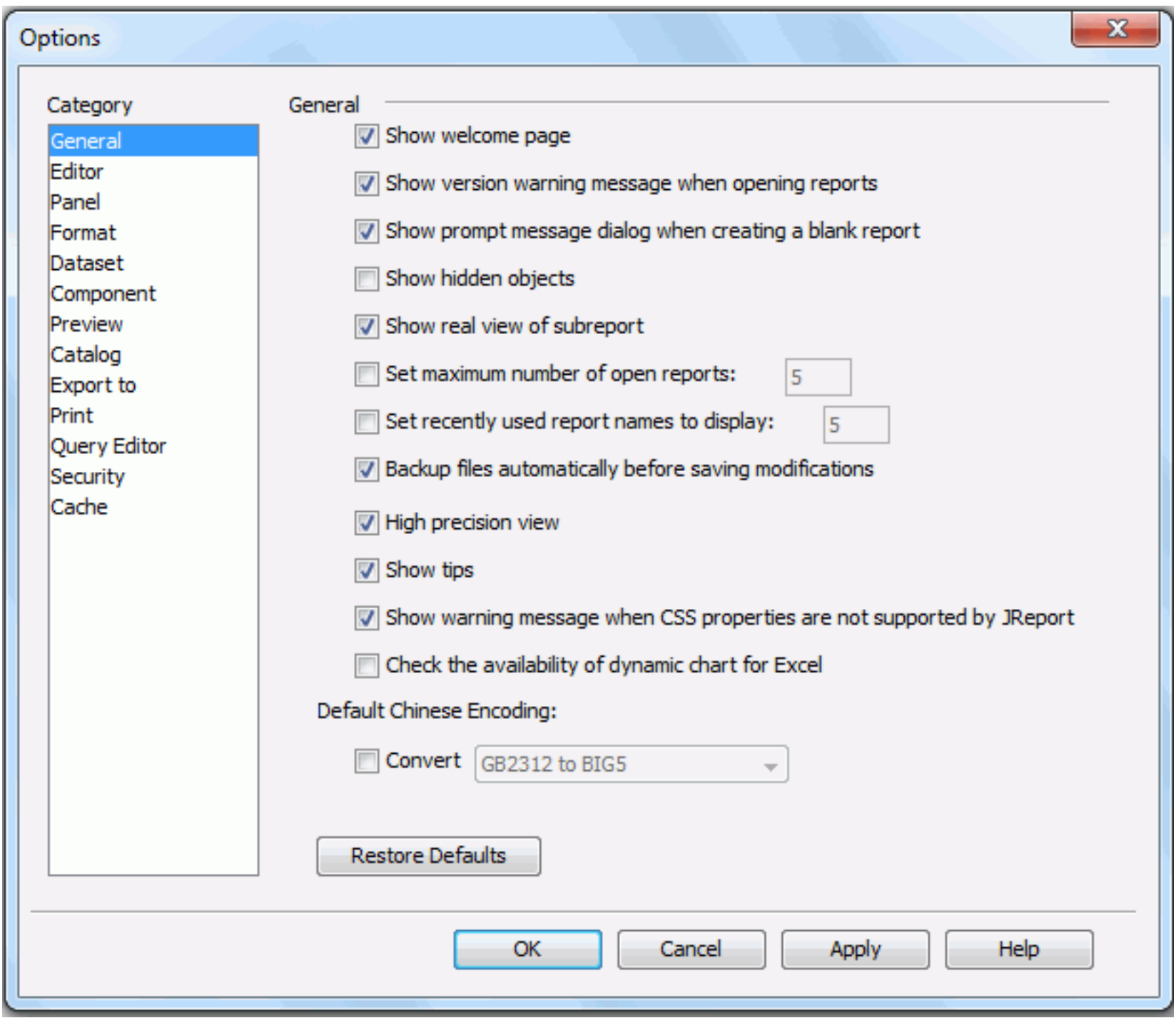
Applies the changes and leaves the dialog open.

Help

Displays the help document about this feature.

General

The following are details about options of this category.



Show welcome page

Specifies whether or not to show the welcome page when you start JReport Designer.

Show version warning message when opening reports

Specifies whether or not to show the version warning message when a report of an earlier version is opened using a higher version. The message warns you that you are about to open a report which belongs to an earlier version, and if you save the report using the current version, you may not be able to open it with the earlier version of JReport Designer again.

Show prompt message dialog when creating a blank report

Specifies whether or not to show the prompt message when a blank report is created. The message prompts you to drag and drop data components from the Toolbox panel and data fields from the Resource View panel onto the blank report.

Show hidden objects

Specifies whether or not to show objects that are hidden in your report.

Show real view of subreport

Specifies whether or not to show the detailed structure of a subreport in its primary report in design mode.

Set maximum number of open reports

Specifies whether or not to limit the number of the reports that can be opened in JReport Designer. If check, you can set the the maximum number of the reports that can be opened in JReport Designer.

Set recently used report names to display

Specifies whether or not to display names of the recently used reports in the welcome page. If checked, you can set the number of report names that are recently used to display.

Backup files automatically before saving modifications

Specifies whether or not to generate the backup files (.cat.bak and .cls.bak) of the catalog and report when they are saved.

High Precision View

Specifies the default value for the High Precision View option on the View menu.

Show tips

Specifies whether or not to show the tooltip when you hover the mouse over a property in the Report Inspector or Catalog Browser, or some dialog option.

Show warning message when CSS properties are not supported by JReport

Specifies whether or not to show the warning message when the applied CSS style contains properties that JReport does not support.

Check the availability of dynamic chart for Excel

Specifies to check if a chart can be correctly mapped to a banded object/table when you save the report or export it to Excel.

Default Chinese Encoding

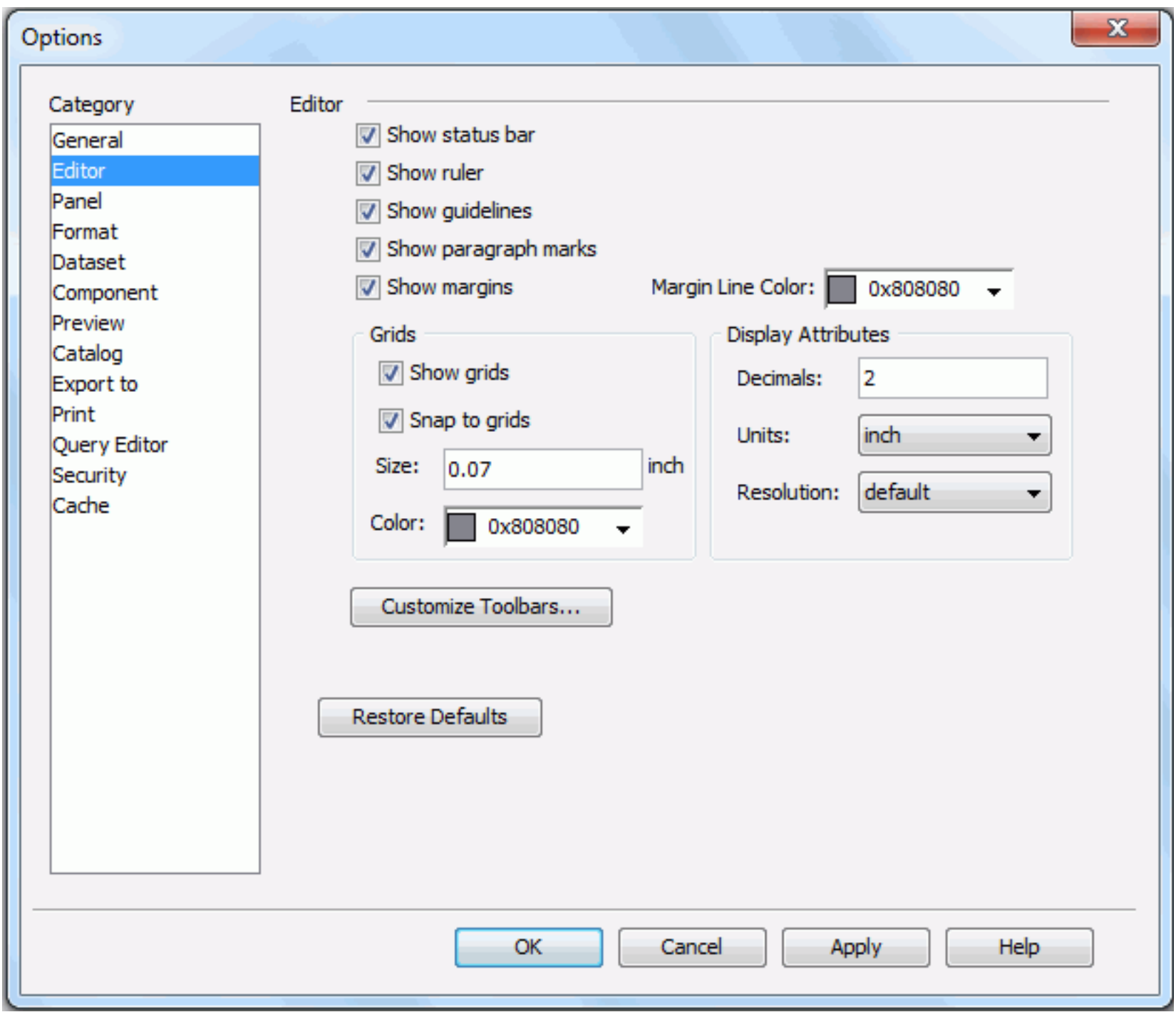
- **Convert**
  Enables the converting encoding function. This function is applied to data which is stored using the Unicode UTF8 in the database. When this function is enabled, the changes will only be apparent when you view the report or export it to certain formats.

Restore Defaults

Restores values of all options to be the default ones.

Editor

The following are details about options of this category.



Show status bar

Specifies whether to show the status bar in JReport Designer.

Show ruler

Specifies whether or not to show the horizontal and vertical rulers in the main window of JReport Designer.

Show guidelines

Specifies whether or not to show the guidelines in the main window of the reports.

Show paragraph marks

Specifies whether or not to show the paragraph marks in the design mode of Designer.

Show margins

Specifies whether to show the margins in the design area.

Margin Line Color

Specifies the color of the margin line that is to be displayed in reports. Available only when Show margins is checked.

Grids

Specifies the grid properties.

- **Show grids**
  Specifies whether to show grids in reports.
- **Snap to grids**
  Specifies whether to snap and lock objects with the grids when moving and placing objects around.
- **Size**
  Specifies the size of the grids in the design area.
- **Color**
  Specifies the grid color.

Display Attributes

Specifies the display attributes.

- **Decimals**
  Specifies the precision of the fractional data in reports. For instance, if you set Decimal to 3, then all the property values of the objects in the Report Inspector will be rounded to the thousandth.
- **Units**
  Specifies the unit of the measurement in reports. It can be either centimeter or inch.
- **Resolution**
  Specifies the resolution. By default, it will be the resolution of your monitor.

Customize Toolbars

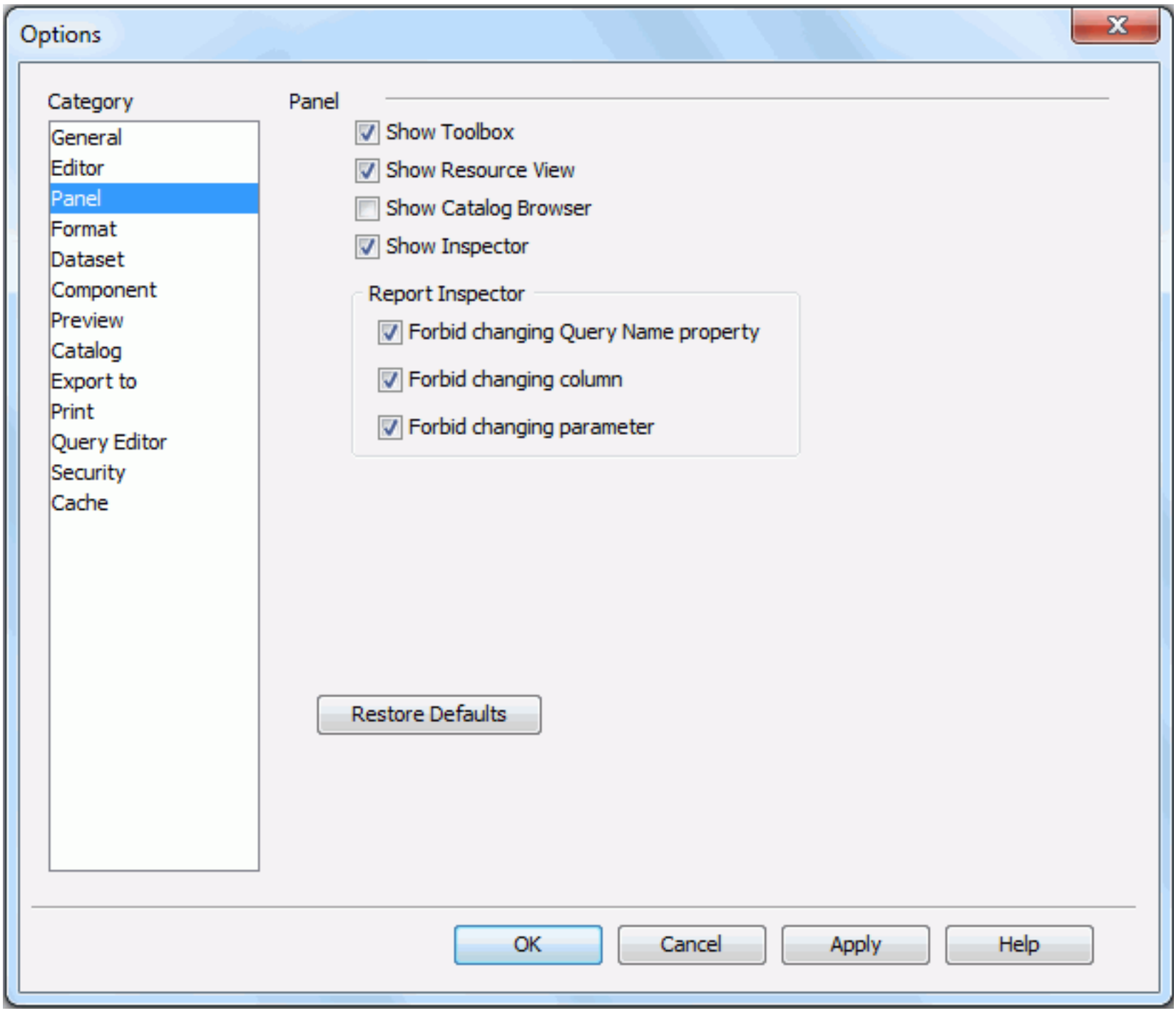
Opens the Customize Toolbars dialog to customize the toolbars.

Restore Defaults

Restores values of all options to be the default ones.

Panel

The following are details about options of this category.



Show Toolbox

Specifies whether to show the Toolbox panel in Designer.

Show Resource View

Specifies whether to show the Resource View panel in Designer.

Show Catalog Browser

Specifies whether to show the Catalog Browser panel in Designer.

Show Inspector

Specifies whether to show the Inspector panel in Designer.

Report Inspector

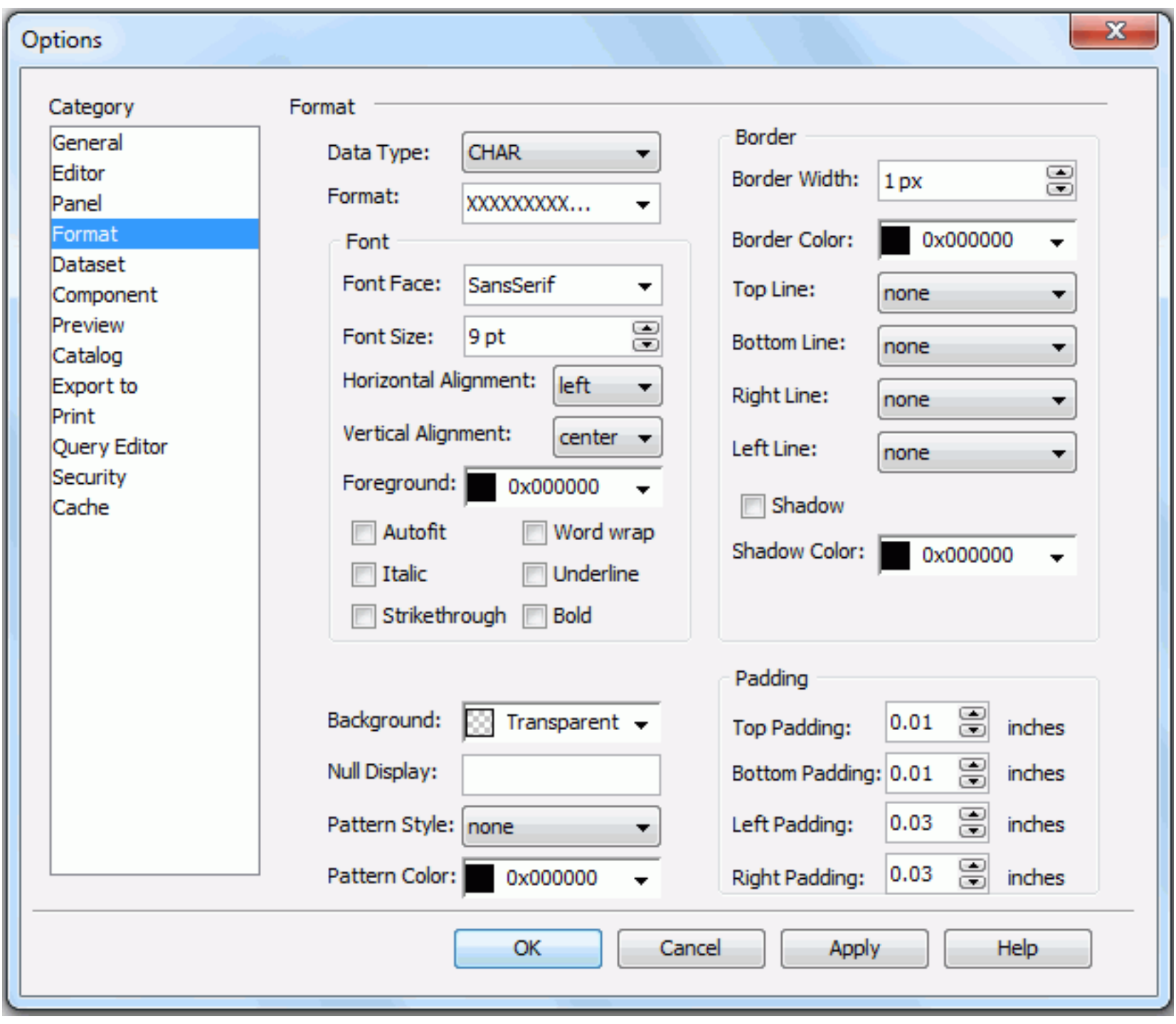
- **Forbid changing Query Name property**
  Specifies whether to allow the changing of the property Query Name or Data Name in the Report Inspector.
- **Forbid changing column**
  Specifies whether to allow the changing of the property Column Name in the Report Inspector.
- **Forbid changing parameter**
  Specifies whether to allow the changing of the property Parameter in the Report Inspector.

Restore Defaults

Restores values of all options to be the default ones.

Format

The following are details about options of this category. The format properties set here will be applied to all columns in a catalog.



Data Type

Specifies the data type of the columns.

Format

Specifies the format of the columns.

Font

Specifies the font properties of the column contents.

- **Font Face**
  Specifies the font face of the column contents.
- **Font Size**
  Specifies the size of the font.
- **Horizontal Alignment**
  Specifies the horizontal alignment of the column contents.
- **Vertical Alignment**
  Specifies vertical alignment of the column contents.
- **Foreground**
  Specifies the color of the font foreground.
- **Autofit**
  Specifies whether to adjust the width according to the contents.
- **Word wrap**
  If checked, the contents will be wrapped.
- **Italic**
  If checked, italic fontface will be used.
- **Underline**
  If checked, the contents will be underlined.
- **Strikethrough**
  Specifies whether or not to draw a line through the column contents.
- **Bold**
  If checked, fold fontface will be used.

Border

Specifies properties of the column border.

- **Border Width**
  Specifies the width of the border. Can be from 1 to 3 points.
- **Border Color**
  Specifies the color of the border. This works only if you specify lines for the borders.
- **Top Line**
  Specifies the top border line style.
- **Bottom Line**
  Specifies the bottom border line style.
- **Right Line**
  Specifies the right border line style.
- **Left Line**
  Specifies the left border line style.
- **Shadow**
  Specifies to draw a shadow to the border.
  - **Shadow Color**
    Specifies the color for the shadow.

Background

Specifies the color for the background of the columns.

Null Display

If there is no value in a column, JReport will display NULL as default. You can also set what will be displayed when there is no value in the column by inserting the mouse in the Report Inspector and inputting a value.

Pattern Style

Specifies the pattern for the columns.

Pattern Color

Specifies the color to draw the pattern.

Padding

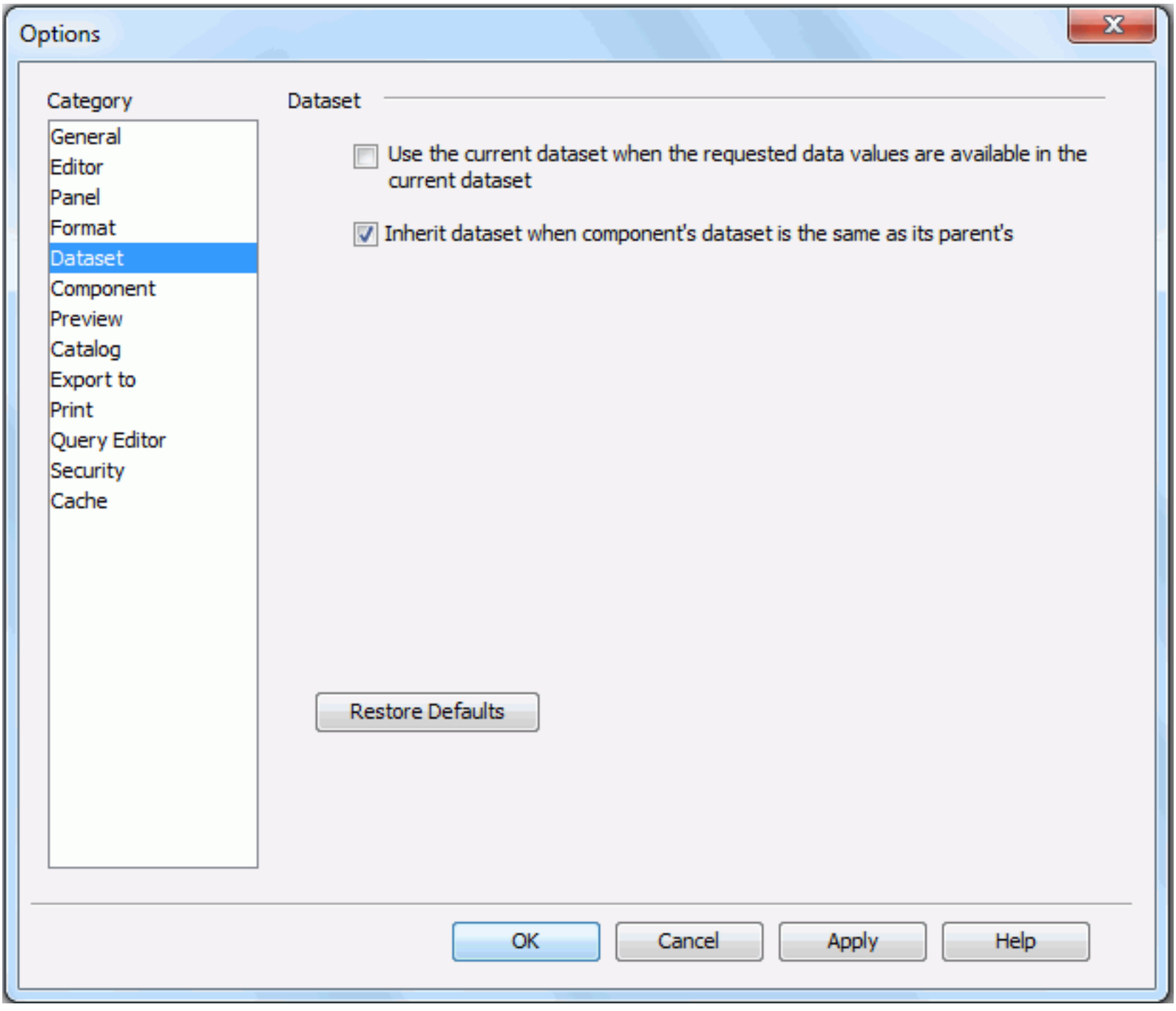
Specifies the spaces between the contents of a column and its borders.

- **Top Padding**
  Specifies the space between the contents of a column and its top borders.
- **Bottom Padding**
  Specifies the space between the contents of a column and its bottom borders.
- **Left Padding**
  Specifies the space between the contents of a column and its left borders.
- **Right Padding**
  Specifies the space between the contents of a column and its right borders.

Note: Values specified here will be applied when data columns are added to a catalog. You can adjust the values for each column in the Catalog Browser, or after the columns are added to reports in the Report Inspector.

Dataset

The following are details about option of this category.



Use the current dataset when the requested data values are available in the current dataset

Specifies whether or not to use the current dataset when the requested data values are available in the current dataset.

Inherit dataset when component's dataset is the same as its parent's

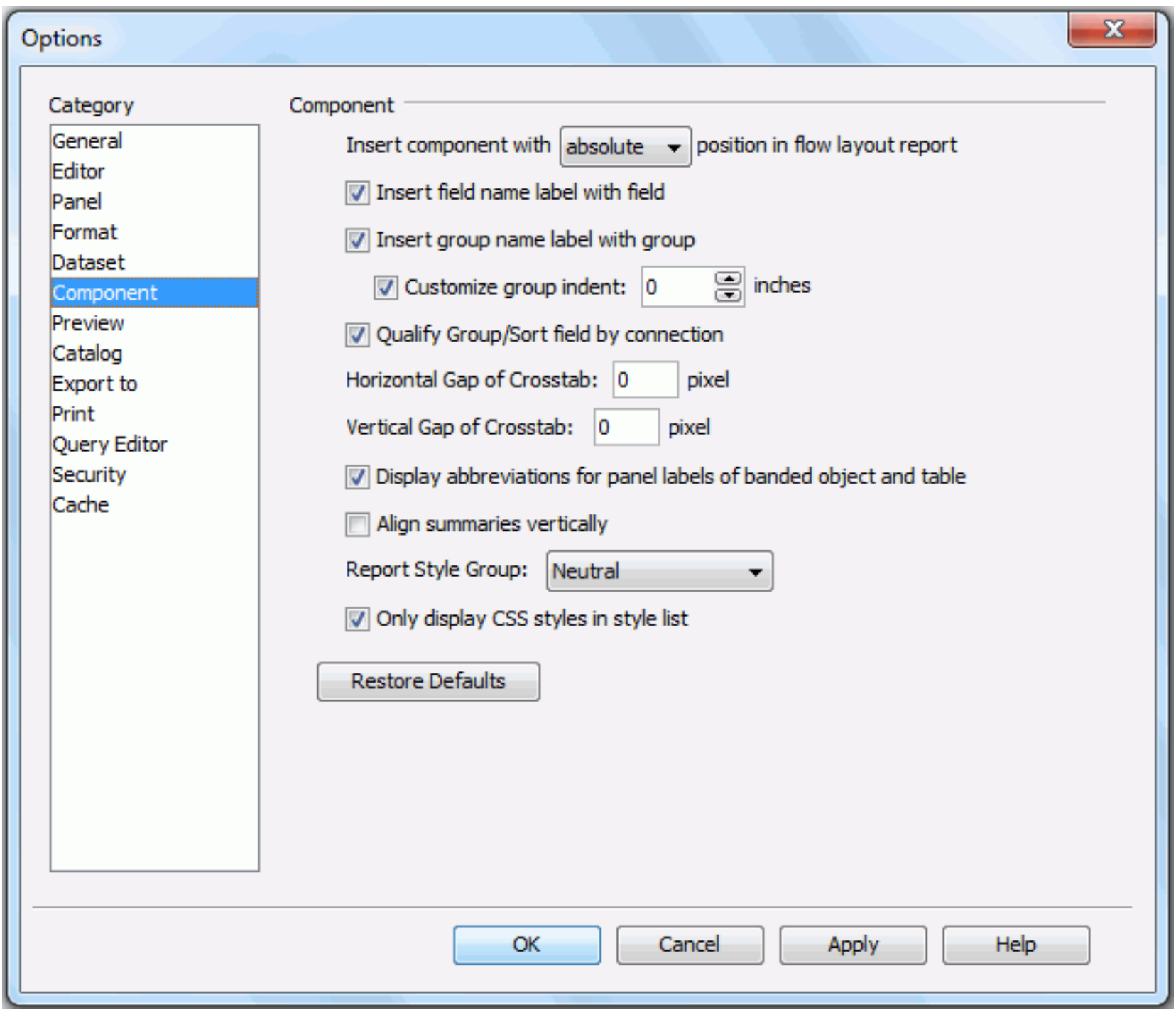
Specifies whether or not to inherit dataset when component's dataset is the same as its parent's dataset.

Restore Defaults

Restores values of all options to be the default ones.

Component

The following are details about options of this category.



Insert component with static/relative/absolute position in flow layout report

Specifies the default method with which to place components:

- **static**
  If selected, the component will be positioned at the location in which it is inserted.
- **relative**
  If selected, the component will be positioned at an offset to the position in which it is inserted. The offset is determined by the X and Y coordinate property values.
- **absolute**
  If selected, the component will be located at the position specified by dragging and dropping or by setting its X and Y coordinate property values.

See the Position property for more information.

Notes:

- When you insert a table, crosstab, chart, tabular, banded object, subreport or map component into a flow layout container, this option does not take effect.
- Since the position for all the components that are inserted into a banded panel is just absolute, so when you insert component into the banded panel, this option does not take effect either.

Insert field name label with field

Specifies whether or not the fields will be inserted into reports together with their default name labels.

Insert group name label with group

Specifies whether or not the groups will be inserted into reports together with their default name labels.

Customize group indent

Specifies whether to customize the indent of groups created via the report wizard. By default, the indent of the groups is 0 inch. You can also input a value between 0 and 4 to customize the indent.

If this option is not checked, when you create groups in a report in the report wizard, the groups will be indented according to the width of the group by fields.

Qualify Group/Sort field by connection

By default, any field added to the catalog in JReport can be used to group and sort. However, you can specify here whether or not a field can be used to group or sort. The qualification information will be stored in the database connection.

Horizontal Gap of Crosstab

Specifies the horizontal gap between two parts of a crosstab, which is split according to the page settings.

Vertical Gap of Crosstab

Specifies the vertical gap between two parts of a crosstab, which is split according to the page settings.

Display abbreviations for panel labels of banded object and table

Specifies whether to display abbreviations for the panel labels of banded object and table in the design area.

Align summaries vertically

Specifies whether to make summaries aligned vertically, which are created via the report wizard. For a table report, this option takes effect only when the summary field of the table is located in the detail column.

Report Style Group

Specifies the default selected style for the four types of components when creating/inserting them via the report/component creation wizard: table, crosstab, chart, and banded object. However, when inserting them into a table or banded object which can bind style, the default selected style type will be Inherit Style instead of the value specified by this option.

All CSS styles in the `<install_root>\style` directory are available in the drop-down list.

Only display CSS styles in style list

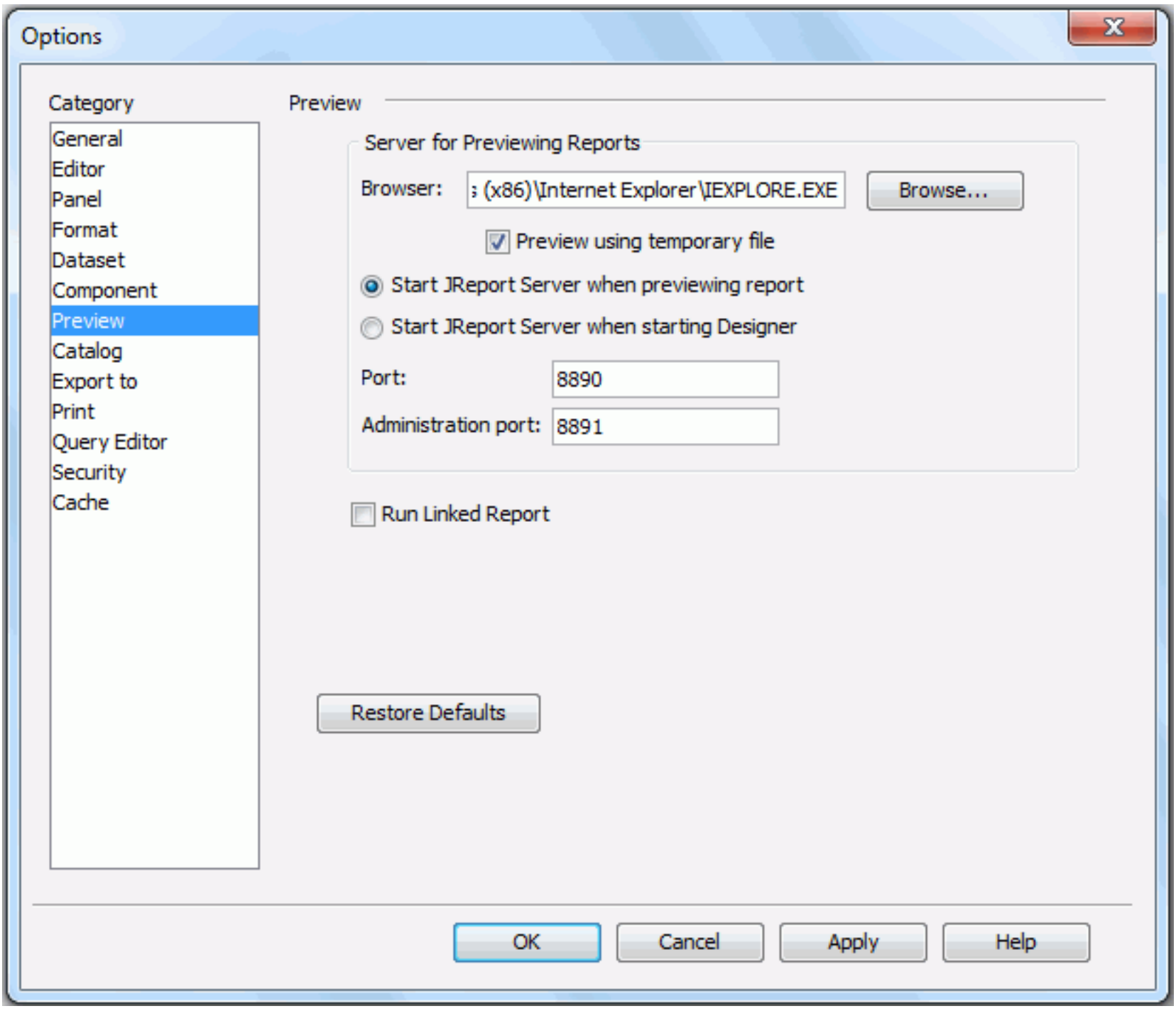
Specifies whether to display only the CSS styles in the style list.

Restore Defaults

Restores values of all options to be the default ones.

Preview

The following are details about options of this category.



Browser

Specifies the browser for previewing reports.

Preview using temporary file

Specifies whether to preview reports that you design in JReport Designer using a temporary file.

Start JReport Server when previewing report

Specifies whether to start JReport Server at the same time when you preview a report.

Start JReport Server when starting Designer

Specifies whether to start JReport Server when Designer is started.

Port

Specifies the port as the TCP port to which JReport Server listens.

Administration Port

Specifies the port used for remote administration.

Run Linked Report

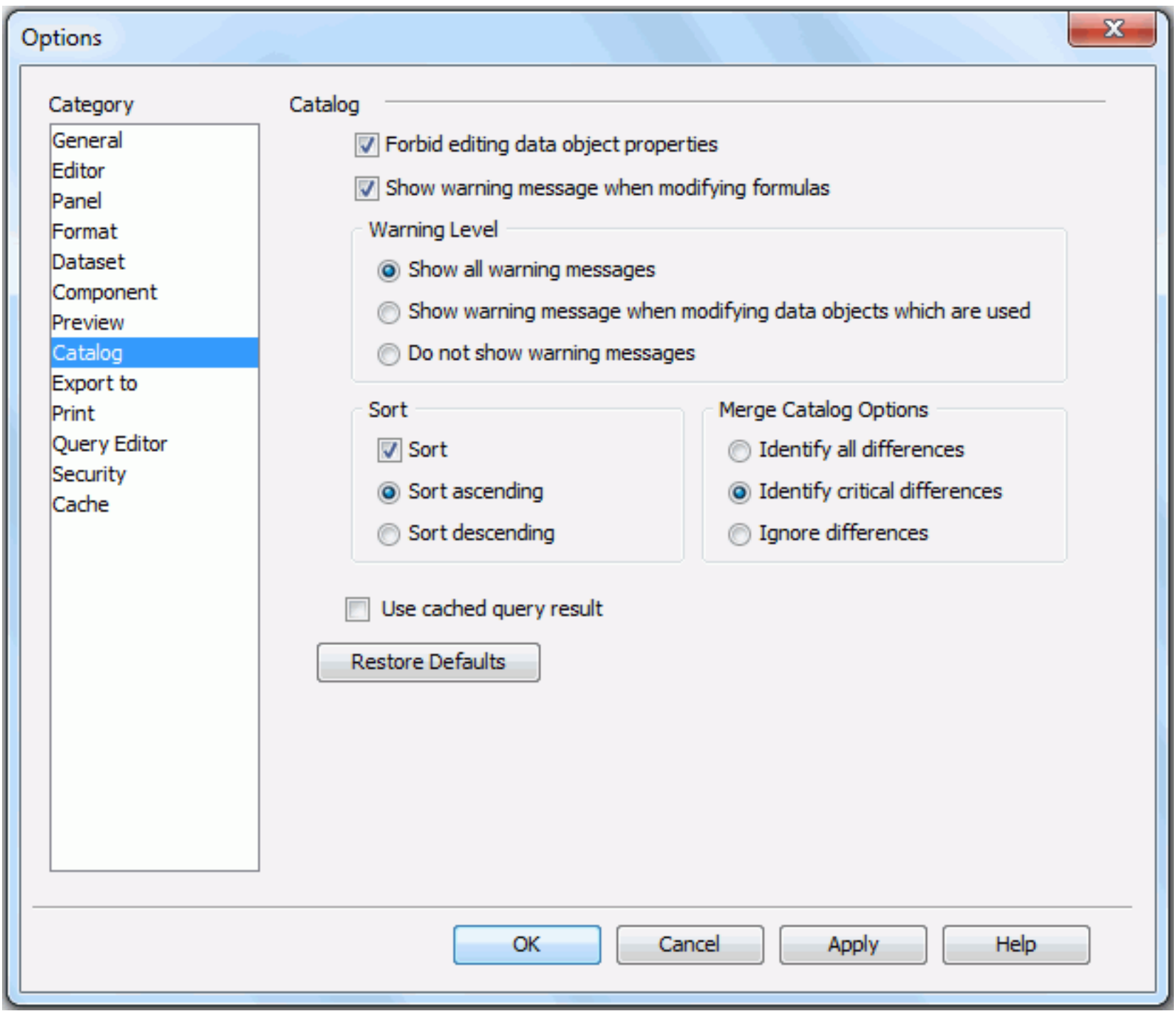
Specifies whether or not to generate the linked reports when previewing the primary reports in all formats except for Web Report and Page Report. Generating linked reports at the same time, especially when the linked reports contain a large amount of data, will cause performance issue.

Restore Defaults

Restores values of all options to be the default ones.

Catalog

The following are details about options of this category.



Forbid editing data object properties

By default, you are not permitted to change the data property value in the Catalog Browser. Here you can specify whether to forbid editing object property values of data.

Show warning message when modifying formulas

Specifies whether or not to show warning message when the formulas are modified. The message warns you that modifying a certain formula may impact other reports using the same formula.

Warning Level

Specifies the warning level from the following three options:

- **Show all warning messages**
  If selected, show all the warning messages.
- **Show warning message when modifying data objects which are used**
  If selected, show warning massage when you modify data objects that are used.
- **Do not show warning messages**
  If selected, do not show the warning messages at any time.

Sort

Specifies whether or not, and how to sort items in the Catalog Browser and report wizard resource tree.

- **Sort ascending**
  If selected, the items will be sorted in an ascending order.
- **Sort descending**
  If selected, the items will be sorted in a descending order.

Note: The sort function only applies to resources belong to the same parent node and of the same type. And real objects in the catalog instead of logic folder nodes can be sorted.

Merge Catalog Options

Specifies the checking level of merging catalogs when you save a report to a directory where a catalog already with the same name exists.

- **Identify all differences**
  If selected, JReport Designer will check for all the differences between the two catalogs. All the resources of the report will be displayed in the Merge dialog with all the conflicting resources marked.
- **Identify critical differences**
  If selected, JReport Designer will check for differences which can cause the engine to fail in running a report. All the resources of the report will be displayed in the Merge dialog with any critical conflicting resources marked. As a result, when this option is selected, certain different values may be returned when you run the saved report later.
- **Ignore differences**
  If selected, the differences between the two catalogs will be shown. If any conflicts exist, they will remain in the resources of the target catalog. As a result, when this option is selected, the saved report might not run later.

Use cached query result

To use cached query result, in the Catalog Browser, right-click on the query (or imported SQL files, stored procedures and UDSs which function as queries) of your report, and choose **Create Cached Query Result**, then specify the file name with or without an extension and the folder where you want to save the result file. After which, when you view the report, a dialog will be displayed for you to choose the data file with which to view the report instead of using ODBC to view the report.

Restore Defaults

Restores values of all options to be the default ones.

Export to

The options of this category are divided into three sub tabs as listed below:

- HTML tab
- E-mail tab
- Fax tab

Restore Defaults

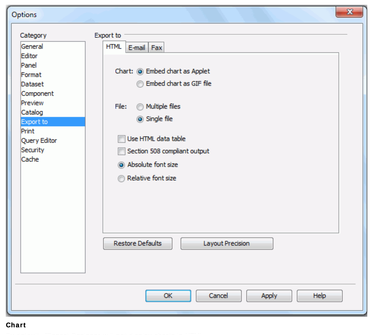
Restores values of all options to be the default ones.

Layout Precision

Opens the Advanced Export Settings dialog to set the precision level for each export format.

HTML tab

Specifies the preferences for exporting the report results to HTML.



Chart

By default, JReport Designer will not display charts in HTML.

- **Embed chart as Applet**
  Specifies whether the chart will be embedded as an Applet for displaying in JReport Designer.
- **Embed chart as GIF file**
  Specifies whether the chart will be embedded as a GIF file for displaying in JReport Designer.

File

Specifies whether to convert the report into a single or multiple files.

- **Multiple files**
  If selected, JReport Designer will convert the report into multiple files and designates a serial number for each HTML page. For example, if a 3-page report is named Sales, JReport Designer will create three files called Sales_1.html, Sales_2.html, and Sales_3.html.
- **Single file**
  If selected, the report will be exported into one file.

Use HTML data table

Specifies whether the table and crosstab components will be output as table objects in the HTML format result.

Section 508 compliant output

If checked, the accessibility attributes defined for the report elements via the Report Inspector will be exported to the HTML format report result which is Section 508 compliant. See Accessibility for more information.

When Section 508 Compliant Output is checked, the Use HTML Data Table and Relative Font Size options will be checked and disabled. The output will be Section 508 compliant including HTML data table, accessible attributes, and relative font feature.

Absolute font size

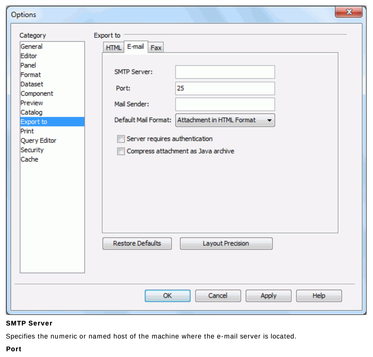
Specifies whether to generate the report result using an absolute font size, which means the font size is fixed and cannot be adjusted according to the font size settings in the web browser.

Relative font size

Specifies whether to generate the report result using a relative font size, which means the font size can be adjusted according to the font size settings in the web browser.

E-mail tab

Specifies the preference for exporting the report results to E-mail.



SMTP Server

Specifies the numeric or named host of the machine where the e-mail server is located.

Port

Specifies the port where the e-mail server runs.

Mail Sender

Specifies the address of the e-mail sender. You must specify an address and make sure that the format of the specified address is valid.

Default Mail Format

Specifies the format with which to send the report result.

- **E-mail Result in HTML E-mail Format**
  Sends the report result via e-mail to the specified address in HTML format. The report result will be shown in HTML format in the mail body.
- **E-mail Result in Plain Text E-mail Format**
  Sends the report result via e-mail to the specified address in plain text format. The report result will be shown in plain text format in the mail body with no other information such as color, images and so on.
- **Attachment in JReport Result Format**
  Sends the report result via e-mail to the specified address with a JReport result file as attachment.
- **Attachment in HTML Format**
  Sends the report result via e-mail to the specified address with an HTML file as attachment.
- **Attachment in PDF Format**
  Sends the report result via e-mail to the specified address with a PDF file as attachment.
- **Attachment in Excel Format**
  Sends the report result via e-mail to the specified address with an Excel file as attachment.
- **Attachment in Text Format**
  Sends the report result via e-mail to the specified address with a Text file as attachment.
- **Attachment in RTF Format**
  Sends the report result via e-mail to the specified address with a RTF file as attachment.
- **Attachment in XML Format**
  Sends the report result via e-mail to the specified address with an XML file as attachment.
- **Attachment in PostScript Format**
  Sends the report result via e-mail to the specified address with a PostScript file as attachment.

Server requires authentication

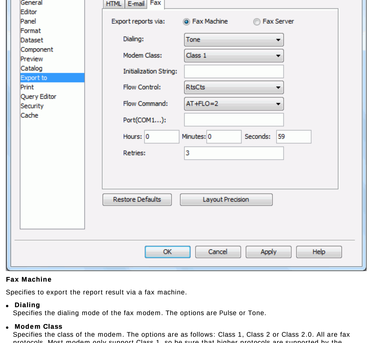
This option must be checked if an SMTP Server requires authentication.

Compress Attachment as Java Archive

Specifies whether or not to compress the mail attachment as Java Archive.

Fax tab

Specifies the preference for exporting the report result to fax. You can specify to export the report result either via a fax machine or a fax server.



Fax Machine

Specifies to export the report result via a fax machine.

- **Dialing**
  Specifies the dialing mode of the fax modem. The options are Pulse or Tone.
- **Modem Class**
  Specifies the class of the modem. The options are as follows: Class 1, Class 2 or Class 2.0. All are fax protocols. Most modem only support Class 1, so be sure that higher protocols are supported by the modem if Class 2 or Class 2.0 is selected.
- **Initialization String**
  Initializes the modem. The string should be obtained from the modem manual.
- **Flow Control**
  Specifies the flow control mode between DTE (Data Terminal Equipment) and DCE (Data Circuit-terminating Equipment). Options are as follows:
  - **RtsCts**
    Flow control of the hardware.
  - **Xon/Xoff**
    Flow control of the software.
  - **None**
    No flow control specified.

    Specifying flow control can help the compressing data function of the modem function better. Using the RtsCts mode is recommended.
- **Flow Command**
  Specifies the flow command according to the flow control mode. The command should be obtained from the modem manual.
- **Port**
  Specifies the port number. The port should be obtained from the modem manual.
- **Timeout**
  Specifies the maximum amount of time for how long the fax should wait for a response from the destination before timing out.
- **Retries**
  When the line is busy, the report result cannot be faxed, so here you can specify the maximum number of times the modem will retry faxing the report result.

Fax Server

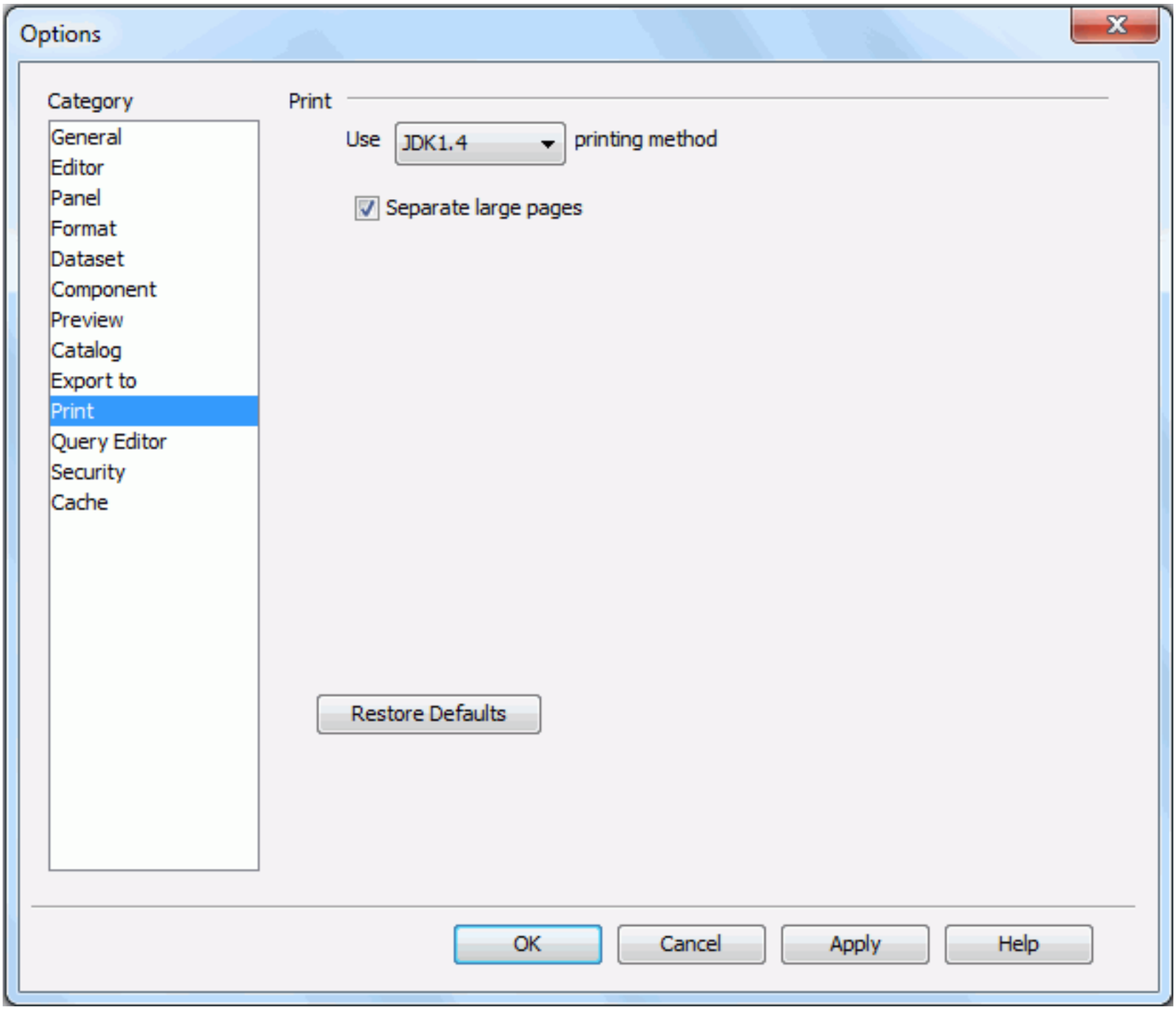
Specifies to export the report result via a fax server.

- **Fax Gateway Connector**
  Specifies the name of the class implemented by users.
- **Server IP**
  Specifies the IP address or domain name of the fax server.
- **Server Port**
  Specifies the port number of the fax server.
- **Login ID**
  Specifies the ID for the class communicating with fax server.
- **Password**
  Specifies the password for the class communicating with fax server.
- **Fax Sender**
  Specifies the user's name that will be shown in the fax server manager.
- **Special Parameters**
  Specifies some parameters for the fax server.
- **Timeout**
  Specifies the maximum amount of time that the fax should wait for a response from the destination before timing out.

  Note: For Hylafax Server the value should not be larger than 59 seconds. It is a limitation of Hylafax Server.
- **Retries**
  Specifies the number of times the modem retries faxing the report result.

Print

The followings are details about options of this category.



Use JDK1.1 / JDK1.2 / JDK1.4 printing method

Specifies the printing method to print reports.

- **JDK1.1**
  This method is quick in speed but the printing quality cannot be guaranteed.
- **JDK1.2**
  This method can give you a satisfactory result, even for a gif file, but it is slow in speed.
- **JDK1.4**
  Compared with the above two methods, this one is more advanced, and will give you a satisfactory result.

Separate large pages

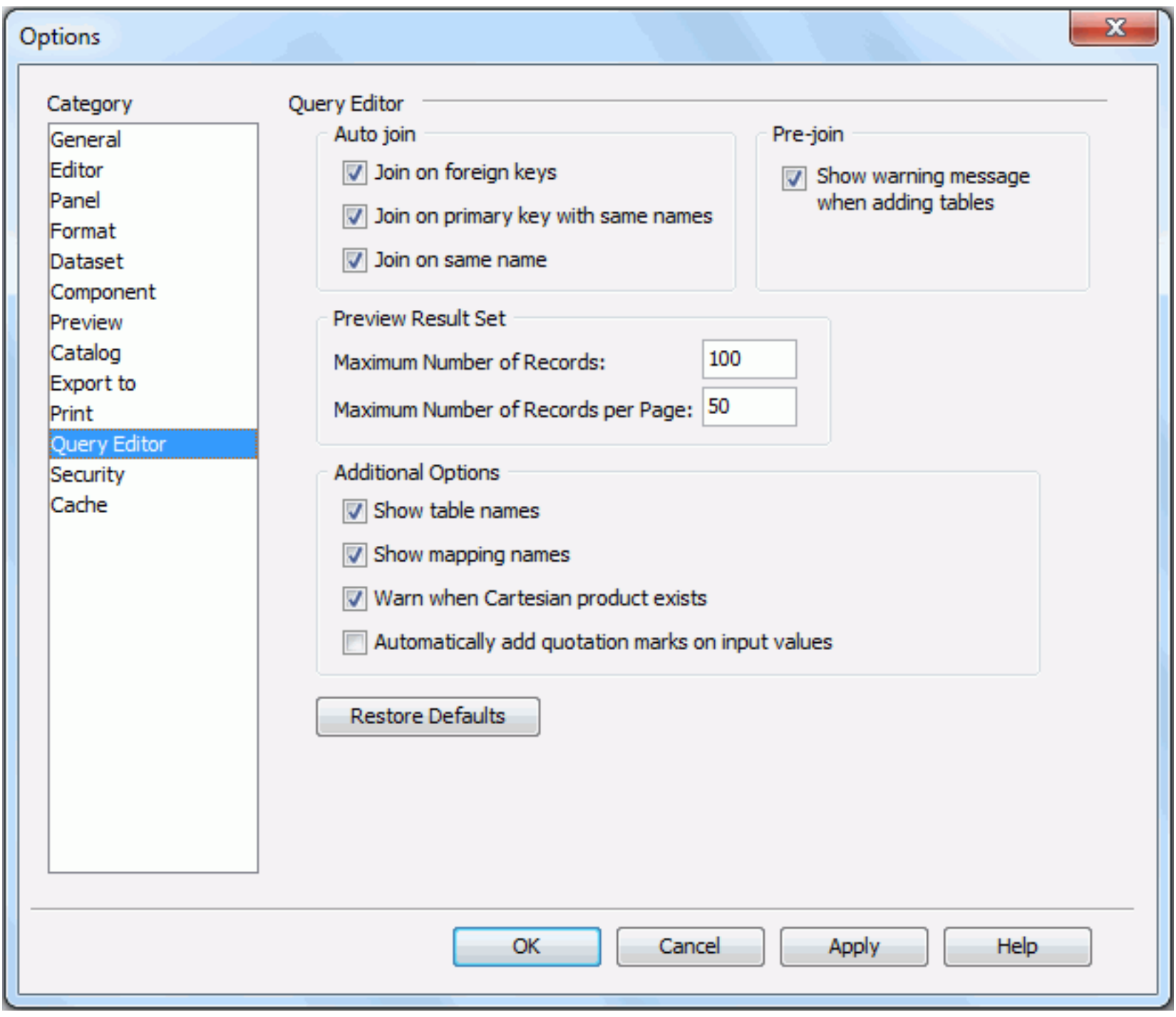
Specifies whether or not to separate large pages during printing. When you print a report with a report page size which is larger than the print paper size, the report will be printed in multiple pages serially. That is, JReport will automatically separate a large page of a report.

Restore Defaults

Restores values of all options to be the default ones.

Query Editor

The following are details about options of this category.



Auto join

Specifies the options for auto joins.

- **Join on foreign keys**
  Specifies to automatically join tables in a query through a reference from Table A to a primary key in Table B. For example, an order form in Table A shows information on purchases that are made by a customer. It can also show this customer's ID #. The customer ID # refers to a record in Table B which lists a specific address, phone number, name, and so on for the customer. The customer ID # in Table A is a foreign key because it links the customer's ordering information in Table A to the customer's information in Table B.

- **Join on primary keys with same names**
  Specifies to automatically join tables in a query through a field or a combination of fields that uniquely and specifically identifies a record. For example, your credit card number is a primary key because it specifically links to your birthday, credit history, address, and so on.

- **Join on same name**
  Specifies to automatically join tables in a query through a link between two columns of the same name in two different tables.

Pre-join

Specifies the default settings for pre-joins.

- **Show warning message when adding tables**
  With the pre-join feature, when a table is added to a query, other tables may also need to be added in order to form a per-join path. This checkbox specifies whether the adding table warning message should be displayed.

Preview Result Set

Specifies the default settings for previewing a result set.

- **Maximum Number of Records**
  Specifies the default total number of records that will be displayed when you previewing a query.

- **Maximum Number of Records per Page**
  Specifies the default maximum number of records to be displayed on one page when previewing a query.

Additional Options

Specifies the additional options.

- **Show table names**
  If selected, the table that each column belongs to will be displayed in the criteria panel of the Query Editor.

- **Show mapping names**
  If selected, the full name of the column in the table will be displayed in the criteria panel of the Query Editor.

- **Warn when Cartesian product exists**
  Specifies whether or not to display a warning when a Cartesian exists. A Cartesian product is used when tables link together with no specifications.

  For example, Table A has three values: A, B and C and Table B has three values: 1, 2 and 3. Value A matches value 1, value B matches value 2, and so on. This is a specific match.

  A Cartesian product could have value A matching with 1, 2 and 3, and value B matching with 1, 2 and 3, and so on. Depending on the data values Cartesian products can produce large and complicated datasets since unnecessary information may be contained.

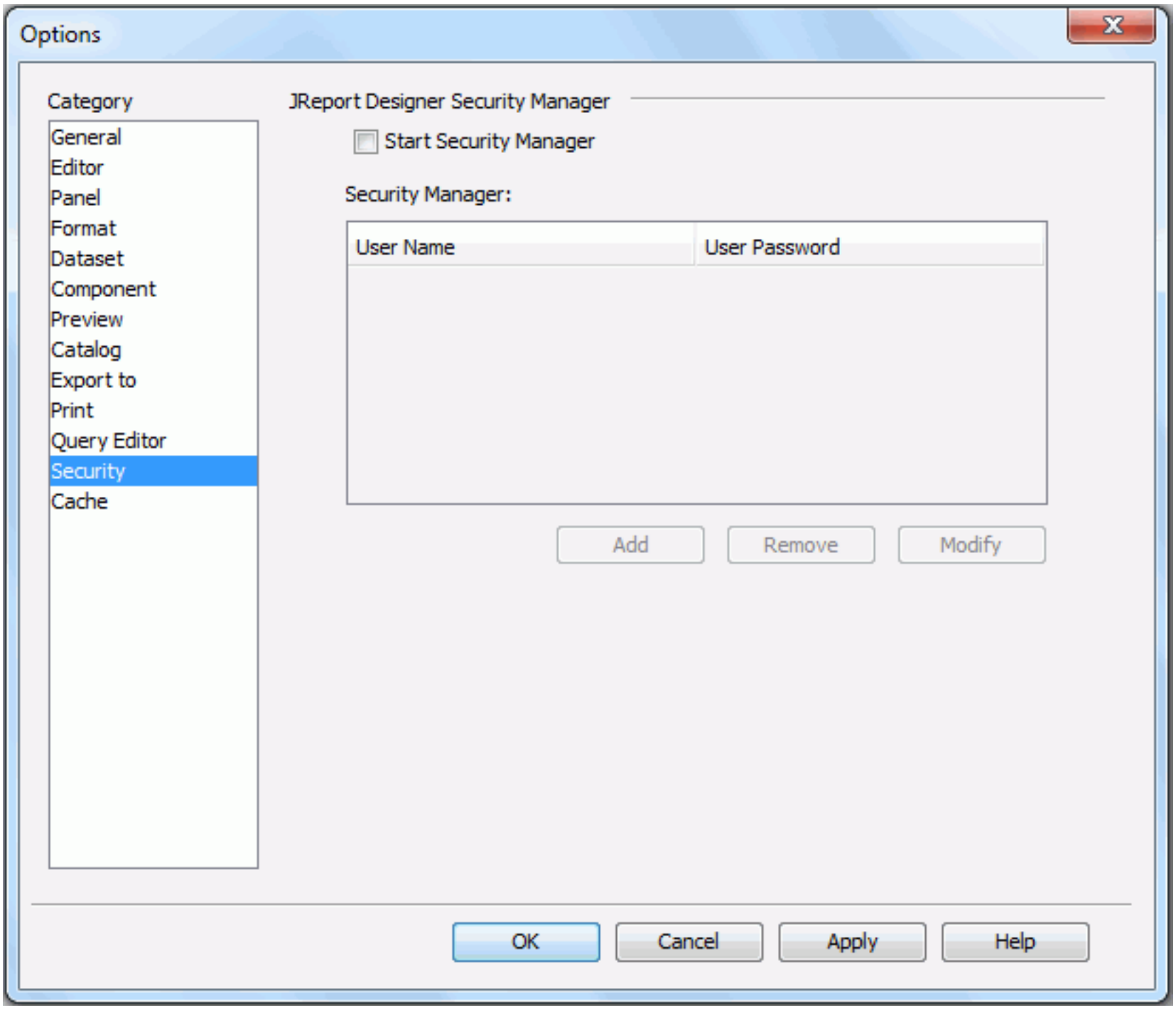
- **Automatically add quotation marks on input values**
  Enables JReport Designer to add quote marks automatically after a String type value is inputted into the second Expression text box in the AND format when you specify certain criteria for the fields to be retrieved from the database.

Restore Defaults

Restores values of all options to be the default ones.

Security

The following are details about options of this category, with which you can set one or more user accounts to use JReport Designer.



Start Security Manager

Specifies whether to enable this feature or not. For the first time you check this checkbox, you are prompted to create a user account in the Report Register dialog:

- **Administrator**
  Specifies the name for the new user account, and at least 4 characters are required.
- **Password**
  Specifies the password to login, and at least 6 digits are required.
- **Confirm Password**
  Confirms the password that you specified in the Password text box.
- **Submit**
  Creates a user account.
- **Cancel**
  Discards the changes.

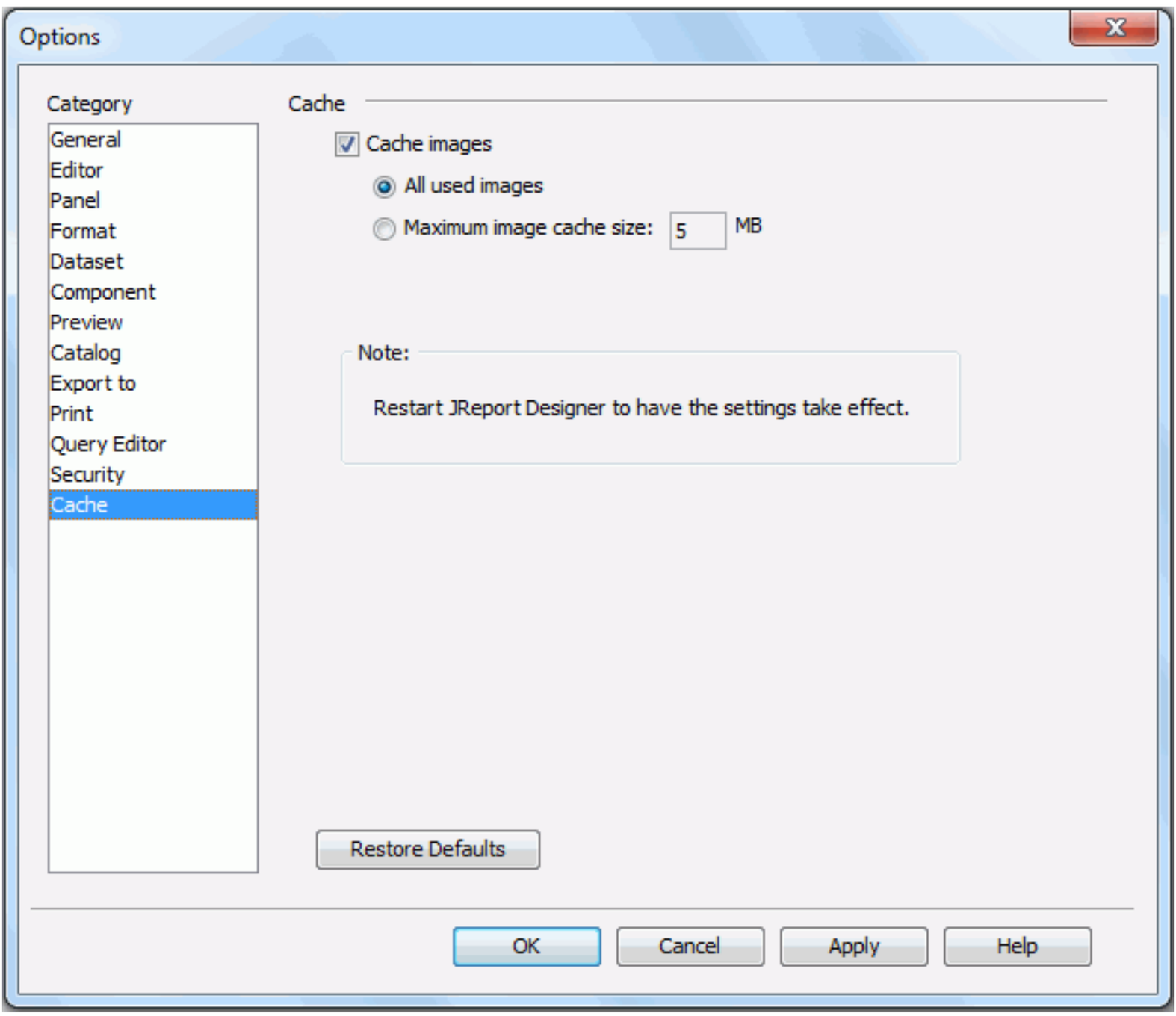
Security Manager

Lists all the user accounts.

- **Add**
  Adds a new user account.
- **Remove**
  Removes the selected user account.
- **Modify**
  Modifies the selected user account.

Cache

The following are details about options of this category, with which you can customize the image cache in JReport Designer.



Cache images

Specifies whether or not to enable image cache.

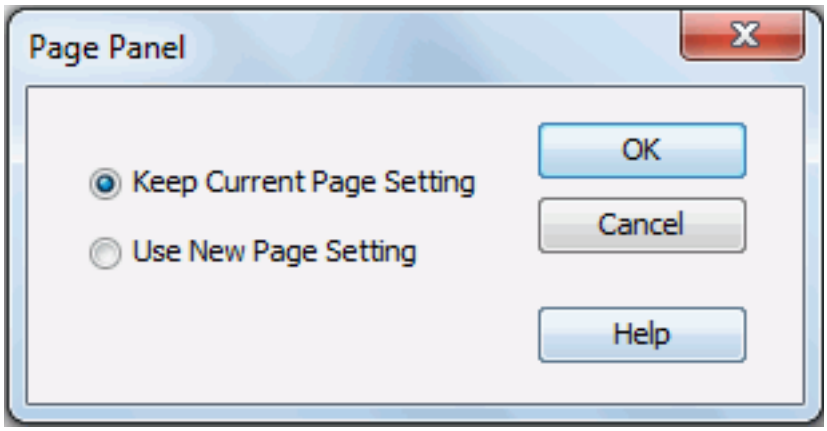
- **All used images**
  If checked, all the images that are used in reports will be cached.
- **Maximum Image Cache Size**
  If checked, specify the maximum size for caching images as required. The default value is 5 MB.

Restore Defaults

Restores values of all options to be the default ones.

Page Panel dialog

This dialog appears when you click Insert > Page Panel on the menu bar. It helps you to create a page panel.



The following are details about the options in the dialog:

Keep Current Page Setting

Specifies to create a new page panel with the current page settings.

Use New Page Setting

Opens the Page Setup dialog to create a new page panel with new page settings.

OK

Accepts all changes and closes the dialog.

Cancel

Does not retain any changes and closes the dialog.

Help

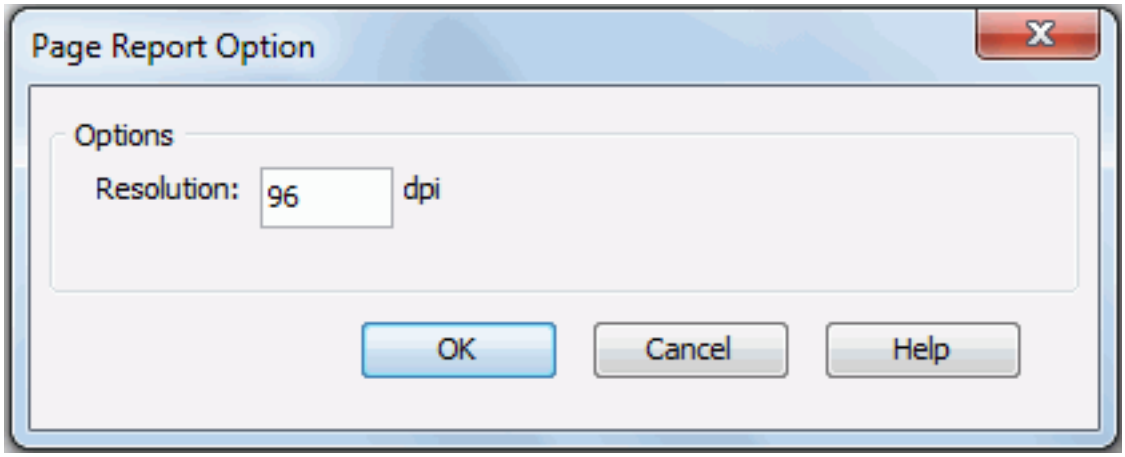
Displays the help document about this feature.

Related topics:

- Page panel

Page Report Option dialog

The dialog appears when you set the Default Format for Viewing Report property of a report tab to Page Report and then click ... in the value cell in the Report Inspector.



The following are details about options in the dialog:

Resolution

Specifies the resolution of the report to zoom in/out, in DPI. The default value is obtained from the operation system, which is the resolution of your monitor, for example, 72 DPI on Unix or 96 on Windows. You can set higher/lower value to zoom in/out.

OK

Applies the changes and closes the dialog.

Cancel

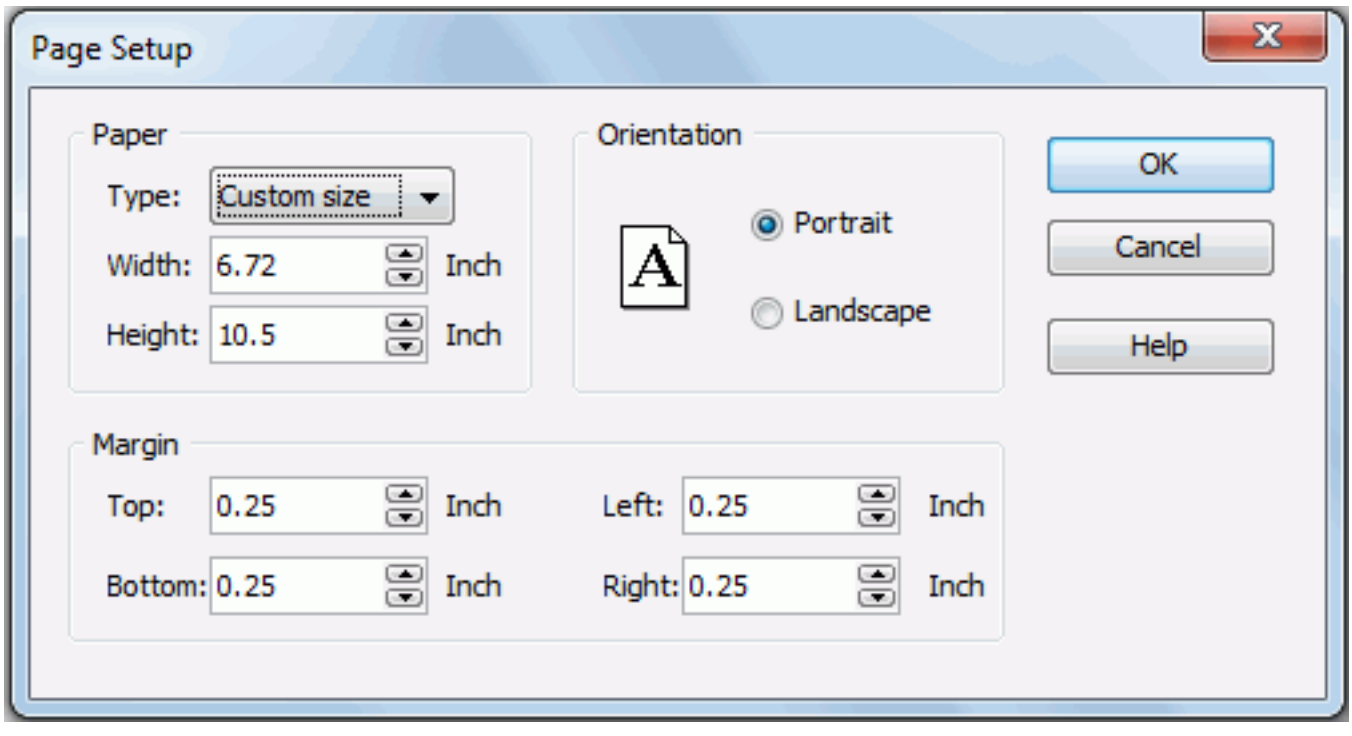
Cancels the changes and exits the dialog.

Help

Displays the help document about this feature.

Page Setup dialog

This dialog appears when you click File > Page Setup on the menu bar, or the Page Setup button on the Standard toolbar. It helps you to set page properties for the active report tab in the current page report, or for the current web report.



The following are details about options in the dialog:

Paper

Specifies the paper size. Select the desired size from the drop-down list, if the given ones cannot meet your requirement, select Custom Size, and then define the paper width and height by yourself.

- **Type**
  Specifies the paper type.
- **Width**
  Specifies the width of the paper, in inches.
- **Height**
  Specifies the height of the paper, in inches.

Orientation

Specifies the paper orientation.

- **Portrait**
  If selected, the page will be displayed as portrait.
- **Landscape**
  If selected, the page will be displayed as landscape.

Margin

Specifies the distance of the report data to the top, bottom, left, or right edge of the page.

- **Top**
  Specifies the top margin of the page, in inches.
- **Bottom**
  Specifies the bottom margin of the page, in inches.
- **Left**
  Specifies the left margin of the page, in inches.
- **Right**
  Specifies the right margin of the page, in inches.

OK

Applies all changes and closes the dialog.

Cancel

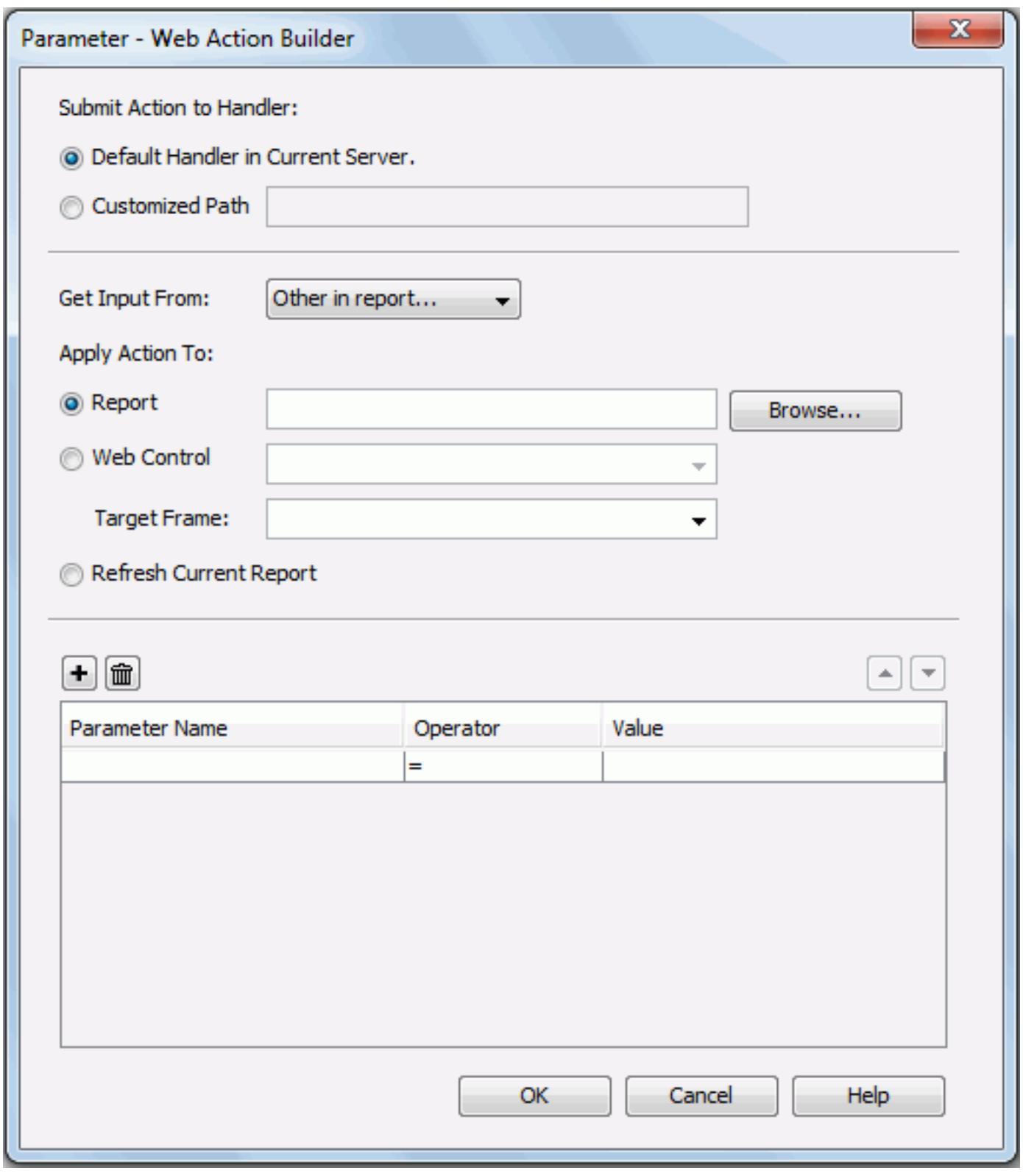
Does not retain any changes and closes the dialog.

Help

Displays the help document about this feature.

Parameter - Web Action Builder dialog

This dialog appears when you select *Parameter and then click OK in the Web Action List dialog. It helps you to build a Parameter web action on an object so as to run a specified report with the parameters defined in the dialog when the event bound with the action is triggered at runtime.



The following are details about options in the dialog:

Submit Action to Handler

Specifies the handler to handle the action.

- **Default Handler in Current Server**
  If selected, the action will be submitted to the current server to be handled.
- **Customized Path**
  If selected, the action will be submitted to a customized path to be handled. For example, `http://127.0.0.1:8888/`.

Get Input From

Specifies where to get the object. You can select Form if you add some objects to a form. Otherwise, only the item Other in Report you can choose.

Apply Action To

- **Report**
  Specifies the report which will be run with the parameters specified in the dialog after the event bound with the action is triggered.
  - **Browse**
    Searches for the desired report. You can also enter the path and file name of the report in the text field manually.
  - **Web Control**
    Specifies the web control from which the report name is retrieved.
- **Target Frame**
  Specifies the window or frame in which the report will be opened.
  - **<Input>**
    Allows you to input the window or frame directly.
  - **Constant**
    - **blank** - Loads the report into a new blank window. This window is not named.
    - **self** - Loads the report into the window in which the link is clicked (the active window).
    - **parent** - Loads the report into the immediate parent window of the active window. If there is no parent window, the active window is used.
    - **top** - Loads the report into the topmost window.
- **Refresh Current Report**
  If checked, after the event bound with the action is triggered, the current report will be re-run with the parameters specified in the dialog.

Adds a column to specify a parameter.

Removes the selected parameter.

Moves the selected parameter up a step.

Moves the selected parameter down a step.

Parameter Name

Lists all the available parameters or web controls that can be selected to use as a parameter in the specified report the action is applied to.

Operator

Shows the operator to compose the expression. It can only be =.

Value

Specifies the value for the parameter.

OK

Accepts the changes and closes the dialog.

Cancel

Does not retain any changes and closes the dialog.

Help

Displays the help document about this feature.

Related topics:

- Applying web actions to a label

Parameter dialog

This dialog helps you to define the parameter values with which you want to run a library component as a response to the message the current data component receives at runtime. Options in the dialog vary according to where the dialog is opened.

OK

Accepts the changes and closes the dialog.

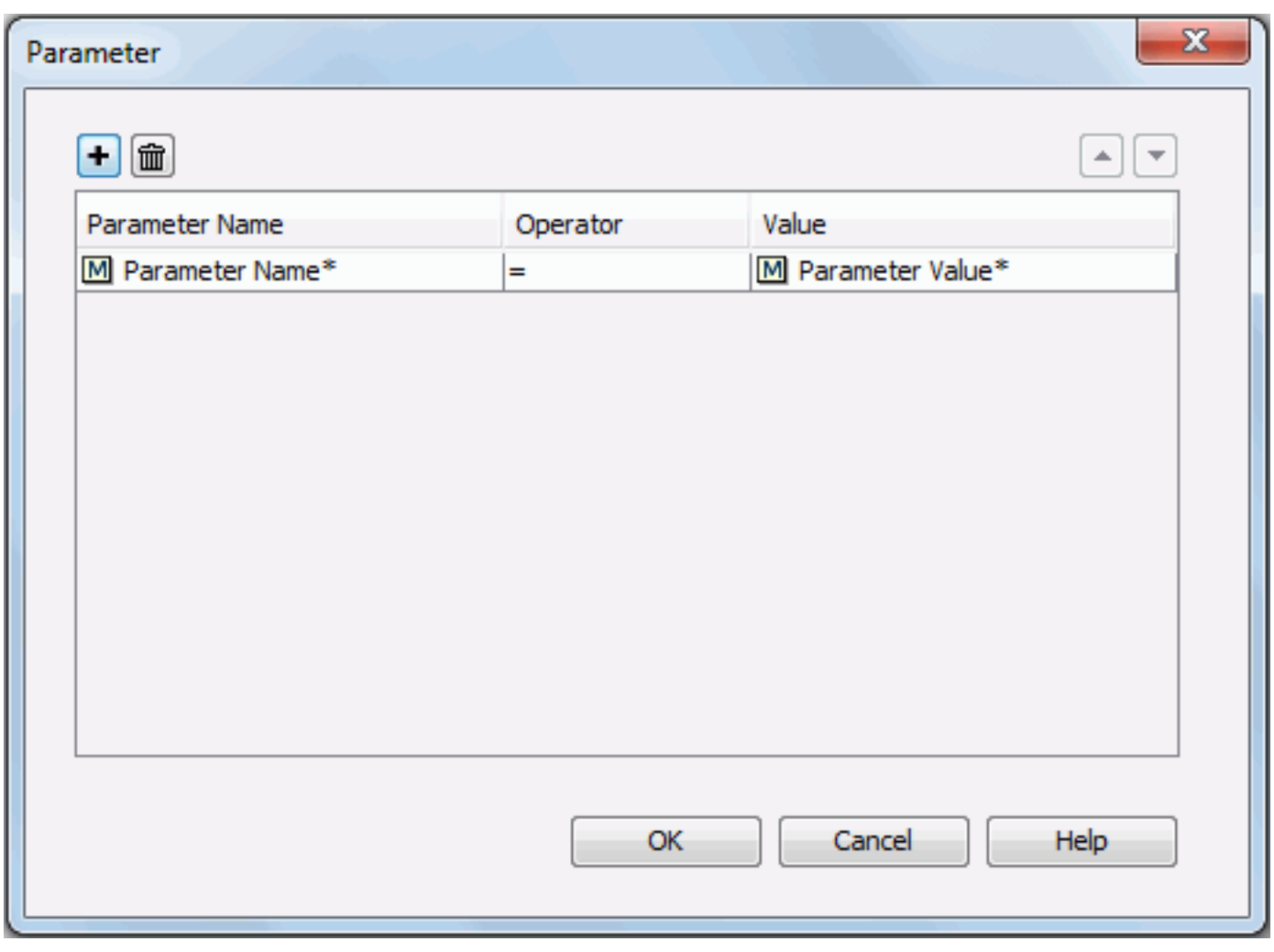
Cancel

Does not retain any changes and closes the dialog.

Help

Displays the help document about this feature.

When the dialog is opened by selecting 0003 - Parameter from the drop-down list of the Message ID column and clicking in the Actions column of the Receive Message dialog, options in the dialog are as follows.



Adds a line to specify parameter value. Not available to the Customized type of the built-in message 0003 - Parameter.

Removes the selected parameter line. Not available to the Customized type of the built-in message 0003 - Parameter.

Moves the selected parameter up a step. Not available to the Customized type of the built-in message 0003 - Parameter.

Moves the selected parameter down a step. Not available to the Customized type of the built-in message 0003 - Parameter.

Parameter Name

Lists all the parameters used by the library component in which the data component that receives the message is. Select one from the drop-down list or input the parameter name by yourself.

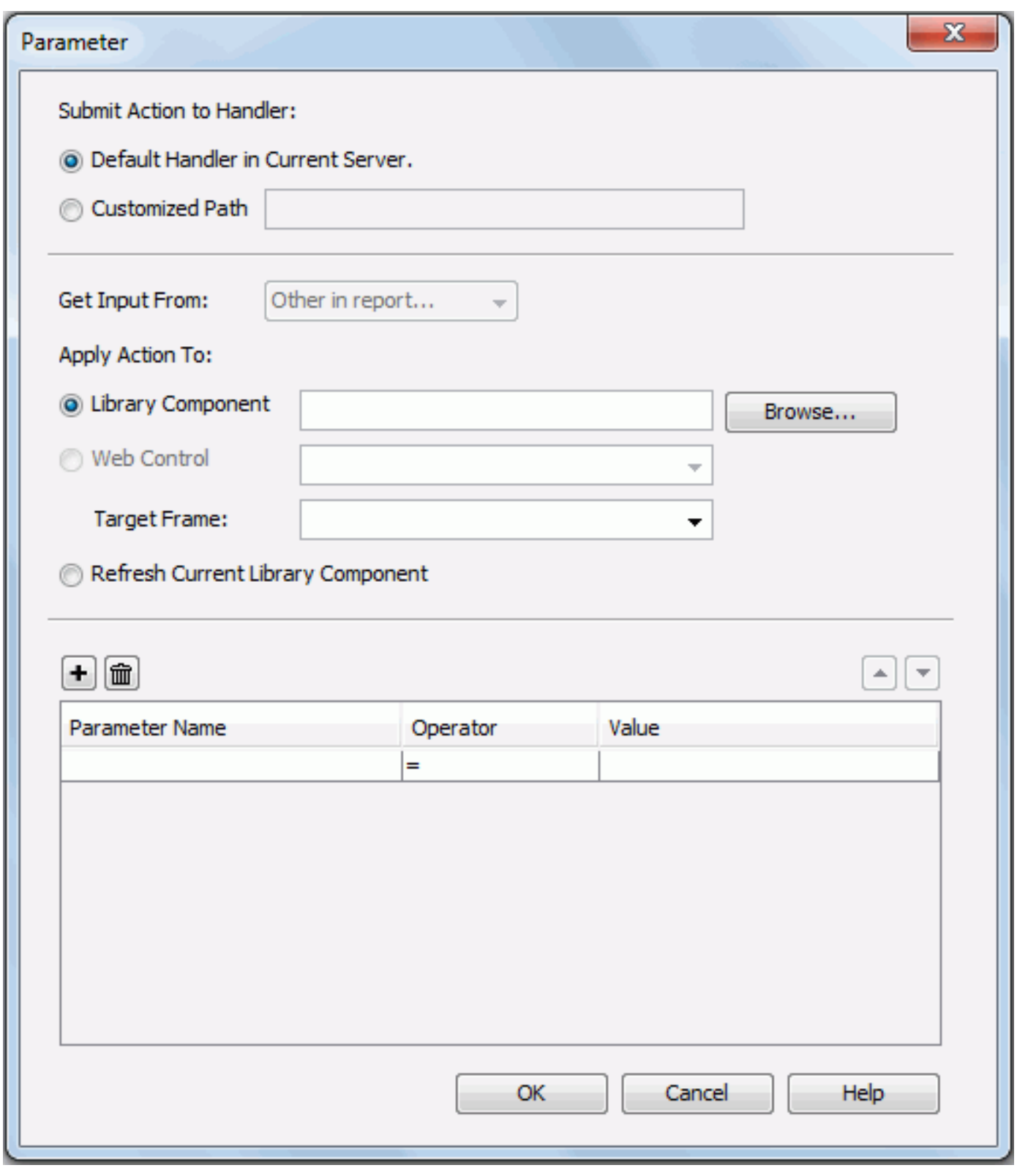
Operator

Shows the operator to compose the expression. It can only be =.

Value

Specifies the value for the parameter.

When the dialog is opened by selecting *Parameter and clicking OK in the Web Action List dialog, options in the dialog are as follows.



Submit Action to Handler

Specifies the handler to handle the action.

- **Default Handler in Current Server**
  If selected, the action will be submitted to the current server to be handled.
- **Customized Path**
  If selected, the action will be submitted to a customized path to be handled. For example, `http://127.0.0.1:8888/`.

Get Input From

Specifies where to get the object. Not supported here.

Apply Action To

- **Library Component**
  Specifies the library component which will be run with the parameter values defined in the dialog when the message is received.
  - **Browse**
    Searches for the desired library component. You can also enter the path and file name of the library component in the text field manually.
  - **Web Control**
    Specifies the web control from which the report name is retrieved. Not supported here.
- **Target Frame**
  Specifies the window or frame in which the library component will be opened.
  - **<Input>**
    Allows you to input the window or frame directly.
  - **Constant**
    - **blank** - Loads the library component into a new blank window. This window is not named.
    - **self** - Loads the library component into the window in which the link is clicked (the active window).
    - **parent** - Loads the library component into the immediate parent window of the active window. If there is no parent window, the active window is used.
    - **top** - Loads the library component into the topmost window.
- **Refresh Current Library Component**
  If checked, the current library component will be re-run with the parameter values defined in the dialog when the message is received.

Adds a line to specify parameter value.

Removes the selected parameter.

Moves the selected parameter up a step.

Moves the selected parameter down a step.

Parameter Name

Lists all the parameters used by the specified library component. Select one from the drop-down list or input the parameter name by yourself.

Operator

Shows the operator to compose the expression. It can only be =.

Value

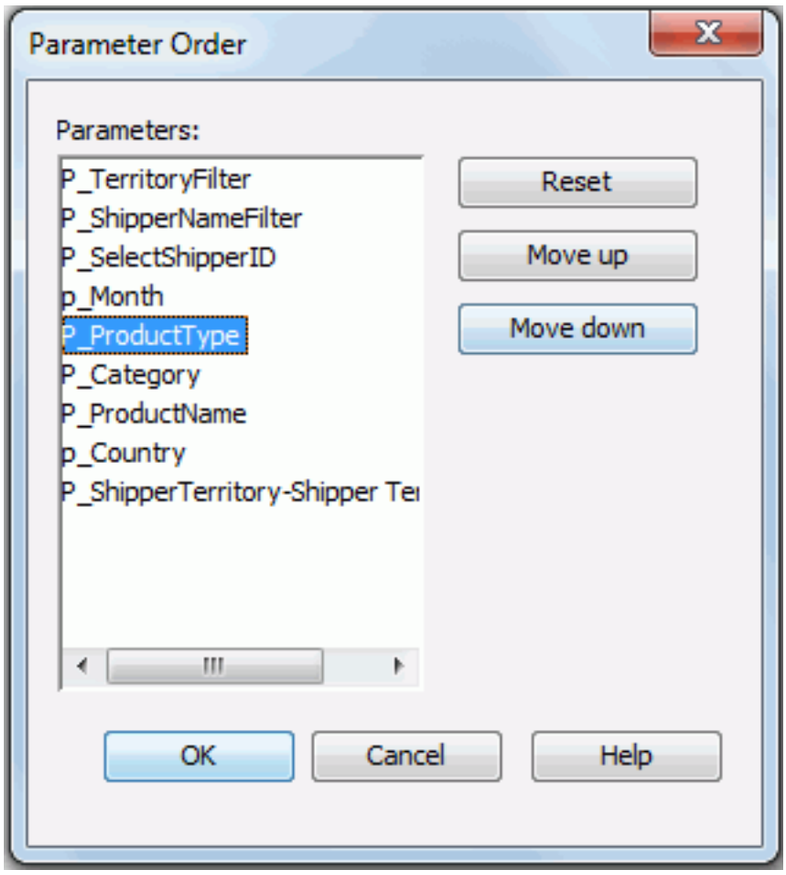
Specifies the value for the parameter.

Related topics:

- Receiving messages

Parameter Run Time Order dialog

The dialog appears when you select the Parameters node or any parameter in the Catalog Browser resource tree and click the Sort Parameter button on the Catalog Browser toolbar, or click Edit > Parameter Order on the menu bar with a report open. It helps you to adjust the order of parameters displayed in the parameter value specifying page. You can set the order either at catalog level or report level.



Parameters

Lists the parameters of all the data sources in the catalog or report. To adjust the order of the parameters, make use of the Move Up and Move Down buttons, or simply drag and drop a parameter.

Reset

If you are setting the order for a catalog, this option is used to apply the Sort setting in the Options dialog > Category panel. If for a report, this option is used to go back to the order for the catalog.

Move Up

Moves a selected parameter closer to the first.

Move Down

Moves a selected parameter further away from the first.

OK

Accepts the change of the order and closes the dialog.

Cancel

Cancels the change of the order and exists the dialog.

Help

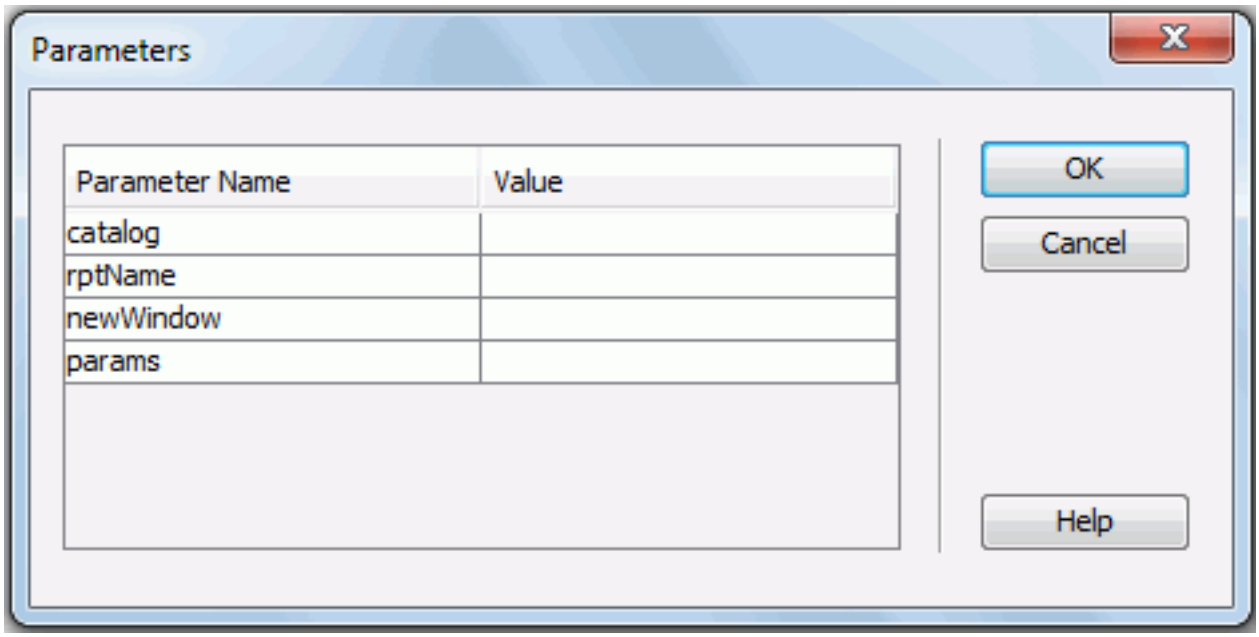
Displays the help document about this feature.

Related topics:

- Editing the display sequence of parameters

Parameters dialog

This dialog appears when you select a web action which requires parameters in the Web Action List dialog and then click OK. It helps you to specify values of the parameters of the web action.



The following are details about options in the dialog:

Parameter Name

Displays the parameters the specified web action requires.

Value

Specifies the values for the parameters. For details about the parameter values, refer to Appendix 3: Parameters of user defined web actions.

OK

Accepts the changes and closes the dialog.

Cancel

Discards the changes and closes the dialog.

Help

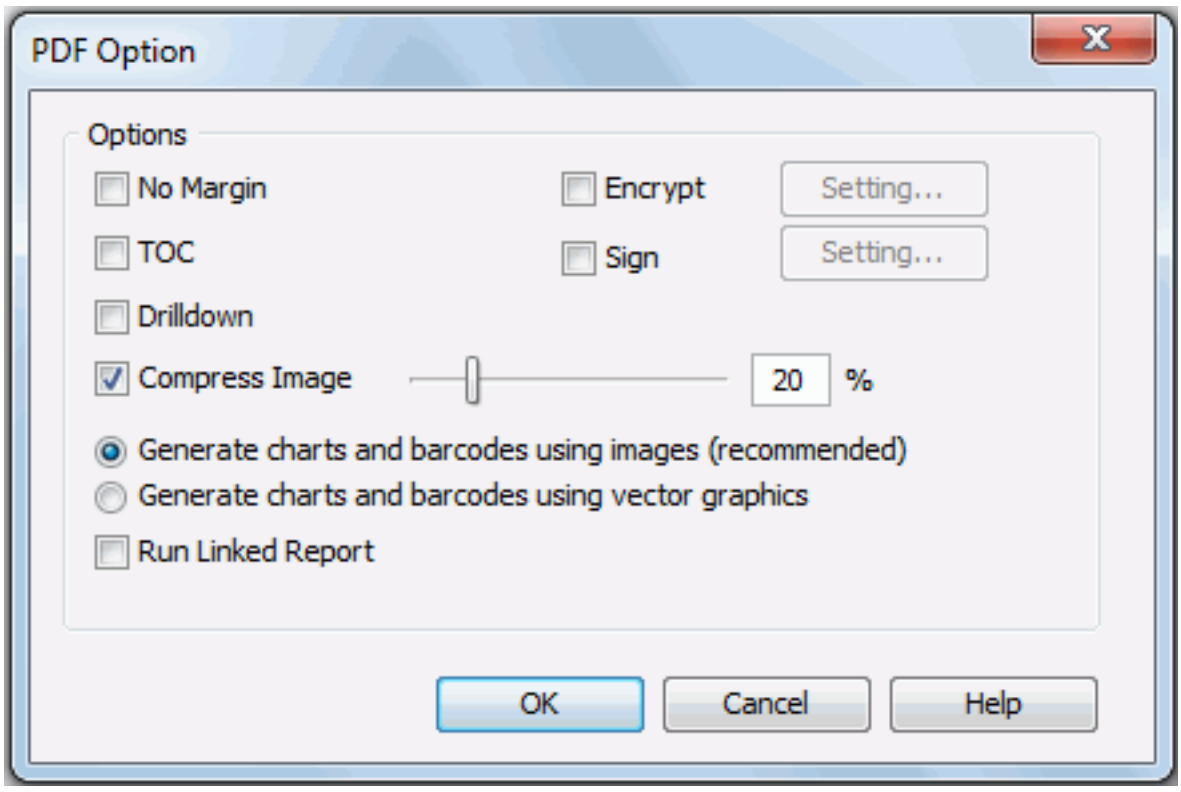
Displays the help document about this feature.

Related topics:

- Applying web actions to a label

PDF Option dialog

The dialog appears when you set the Default Format for Viewing Report property of a report tab or a web report to PDF and then click ... in the value cell in the Report Inspector. It helps you to specify the parameters for running a report in PDF format.



The following are details about options in the dialog:

No Margin

Removes the margins you originally set while designing the report.

TOC

Generates the report result to PDF format with a Table of Contents.

Drilldown

Generates the report result to a PDF file with the Drilldown feature enabled. The Drilldown feature enables you to inspect certain items for further detailed data.

Encrypt

Specifies whether to encrypt the PDF file. Select the option and you can configure the encrypt settings in the Encrypt dialog.

Sign

Specifies whether to add the digital sign to the PDF file. Select the option and you can configure the sign settings in the Sign Digital ID dialog.

Compress Image

Compresses the images in the report by the percentage you specify in the box.

Generate charts and barcodes using images (recommended)

When you run a report in PDF format, JReport will take the result of the whole report as a graphic to transform the report by the method of simulated printer and generate the report result in PDF format.

Generate charts and barcodes using vector graphics

The result of using Generate charts and barcodes using vector graphics is the same as with using Generate charts and barcodes using images (recommended). However, when you run a report in PDF format using Generate charts and barcodes using vector graphics, JReport will take the result of the whole report as a dataset to transform the report by sequence and to generate the report result in PDF format.

Run Linked Report

If the report is linked with other reports, you can check whether or not to generate the linked reports in the PDF result. If you are only interested in the primary report, leave this option unchecked. Generating linked reports at the same time, especially when the linked reports contain a large amount of data, will cause performance issue.

OK

Applies the changes and closes the dialog.

Cancel

Cancels the changes and exits the dialog.

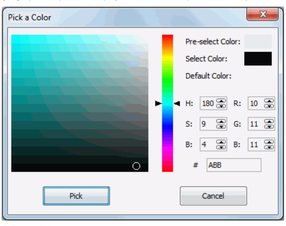
Help

Displays the help document about this feature.

Pick a Color dialog

This dialog helps you to specify a color for an object, and varies according to different source it is opened from.

If it is opened when you specify the color schema for chart elements or for the markers or areas of a geographic map in the Display screen of the map wizard, the options in the dialog are as follows.



Color matrix

Select a color you want in the color matrix. The hue of the matrix is based on the color you define on the color bar.

Color bar

Specifies a color on the bar which will be the hue of the color matrix.

Pre-select Color

Displays the color you have pre-selected in the color matrix. The color result varies when you hover the mouse pointer in the color matrix. When the pre-selected color meets your requirement, click it in the color matrix, then the color will be displayed as the Select Color value.

Select Color

Displays the color you have selected in the color matrix.

Default Color

Displays the default color for the object.

Hue

Specifies the chroma of the color, whose value is from 0 to 359.

Saturation

Specifies the saturation of the color, whose value is from 0 to 100.

Brightness

Specifies the brightness of the color, whose value is from 0 to 100.

Red

Specifies the red value, whose value is from 0 to 255.

Green

Specifies the green value, whose value is from 0 to 255.

Blue

Specifies the blue value, whose value is from 0 to 255.

#

Displays the hexadecimal value of the selected color. You can change its value by typing your desired hexadecimal value manually.

Pick

Clicks to apply the color value you have defined and leaves the dialog.

Cancel

Cancels the changes you have made and closes the dialog.

If it is opened from other sources, the options in the dialog are as follows:

- Swatches tab
- HSB tab
- RGB tab

Preview

Shows the specified color.

OK

Applies all changes and closes the dialog.

Cancel

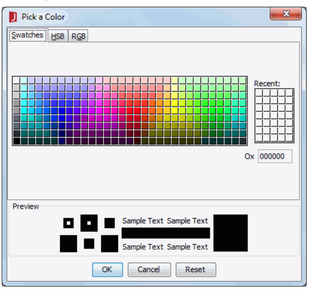
Does not retain changes and closes the dialog.

Reset

Cancels all changes and allows you to reset the values of the color.

Swatches tab

Shows samples of the colors.



Swatches

Shows colors as samples for you to choose.

Recent

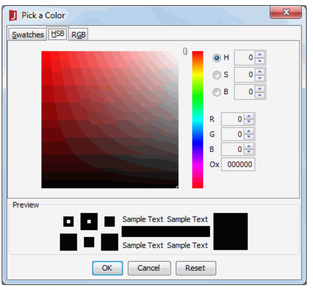
Shows the colors you have chosen recently.

Hex

Shows the RGB hexadecimal value of the color you choose, which can be copied by selecting it and then pressing Ctrl+C on the keyboard, but it cannot be edited.

HSB tab

Specifies the attributes of the color.



Color matrix

Select a color you want in the color matrix. The hue of the matrix is based on the color you define on the color bar.

Color bar

Specifies a color on the bar which controls the hue of the color matrix. You can change its value by dragging the slider between the color matrix and color bar.

Hue

Specifies the chroma of the color, whose value is from 0 to 359.

Saturation

Specifies the saturation of the color, whose value is from 0 to 100.

Brightness

Specifies the brightness of the color, whose value is from 0 to 100.

Red

Specifies the red value, whose value is from 0 to 255.

Green

Specifies the green value, whose value is from 0 to 255.

Blue

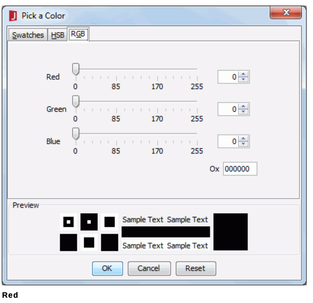
Specifies the blue value, whose value is from 0 to 255.

Hex

Specifies the RGB hexadecimal value of the color, which changes along with the HSB and RGB values, or it can be changed by entering a hexadecimal value manually.

RGB tab

Specifies the RGB of the color.



Red

Specifies the red value, whose value is from 0 to 255.

Green

Specifies the green value, whose value is from 0 to 255.

Blue

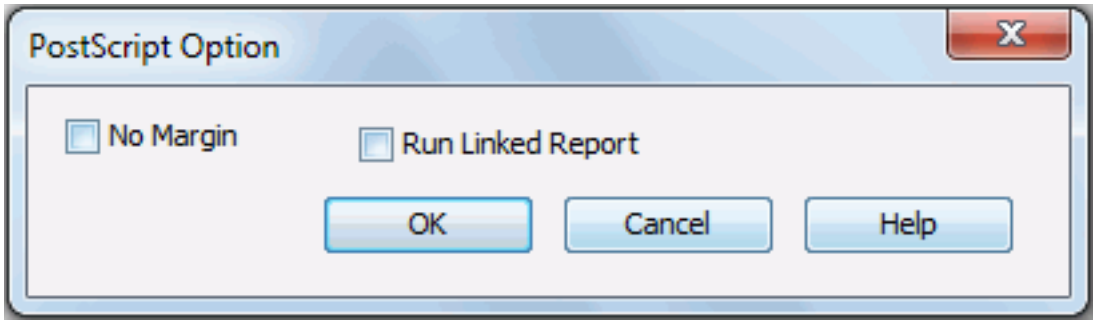
Specifies the blue value, whose value is from 0 to 255.

Hex

Specifies the RGB hexadecimal value of the color, which changes along with the RGB values, or it can be changed by entering a hexadecimal value manually.

PostScript Option dialog

The dialog appears when you set the Default Format for Viewing Report property of a report tab or a web report to PostScript and then click ... in the value cell in the Report Inspector. It helps you to specify the parameters for running a report to PostScript format.



The following are details about options in the dialog:

No Margin

Removes the margins you originally set while designing the report.

Run Linked Report

If the report is linked with other reports, you can check whether or not to generate the linked reports in the PostScript result. If you are only interested in the primary report, leave this option unchecked. Generating linked reports at the same time, especially when the linked reports contain a large amount of data, will cause performance issue.

OK

Applies the changes and closes the dialog.

Cancel

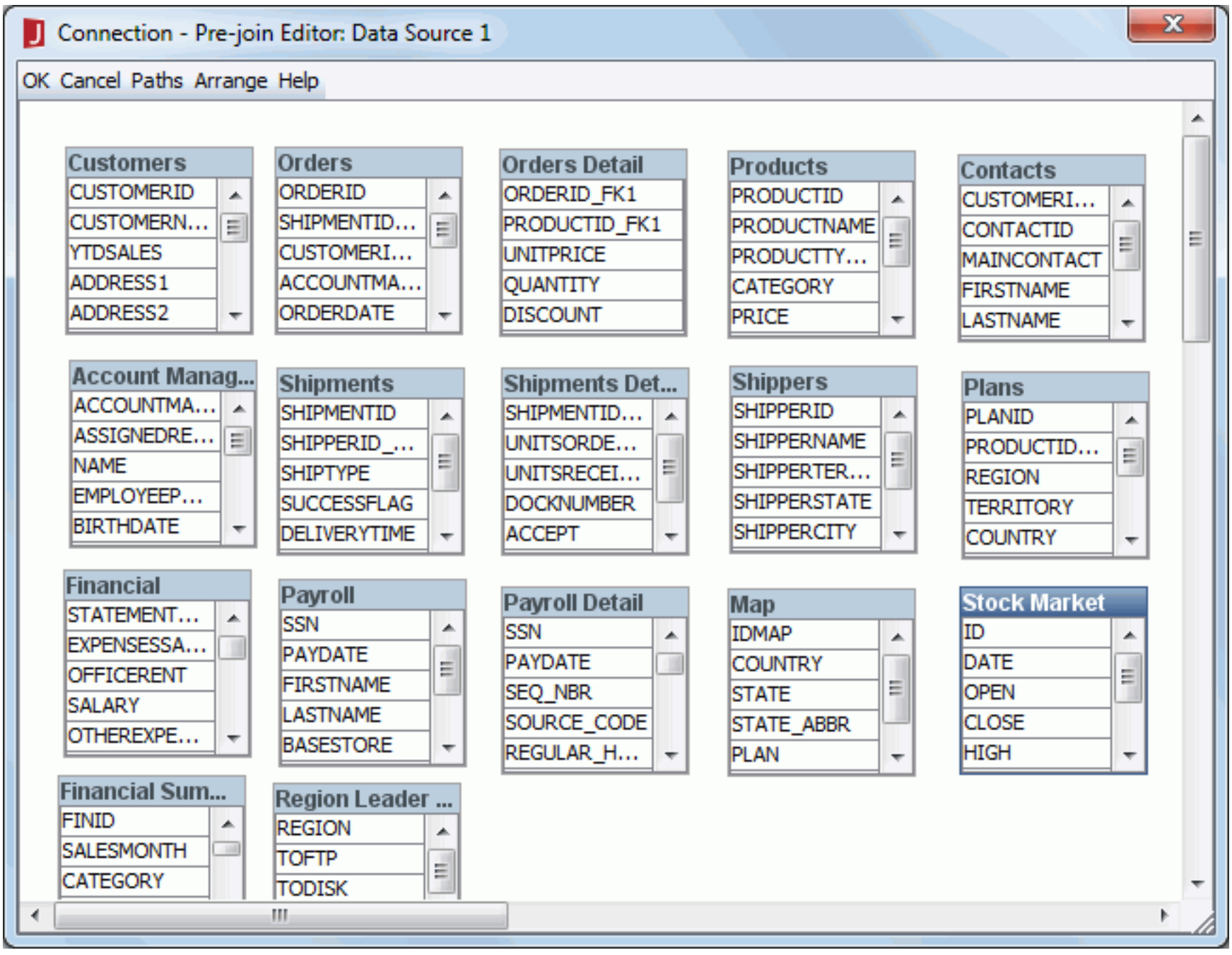
Cancels the changes and exits the dialog.

Help

Displays the help document about this feature.

Pre-join Editor window

This window appears when you click the Pre-join button on the Catalog Browser toolbar, select a data source and then click OK. It helps you to edit pre-joins in the selected data source of the current catalog, which can be used for creating/editing queries.



You can make joins between tables by using mouse drag and drop. Point to one column in the source table, press and hold the mouse button, then move the pointer to the other column in the target table and release the mouse button. A green line with a join button will be shown, linking the two columns.

If you want to further edit a join, click its join button and the Join Options dialog will be displayed for you to edit it.

After you have made some joins between the tables, you can then use them to define paths. To define paths, click the **Paths** button on the toolbar of the editor.

OK

Applies all the changes and closes this dialog.

Cancel

Does not retain any changes and closes this dialog.

Paths

Opens the Save Pre-join Path dialog to define pre-join paths.

Arrange

Organizes the tables in the table panel.

Help

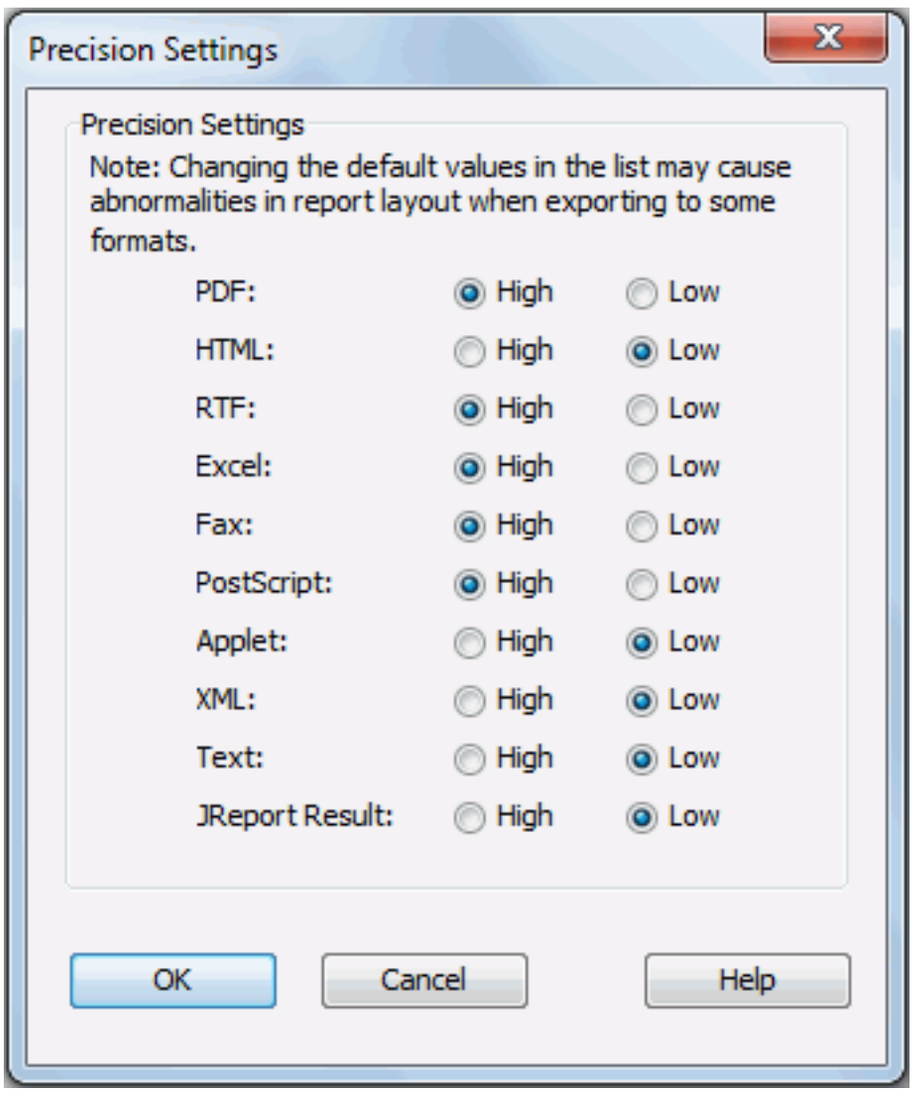
Displays the help document about this feature.

Related topics:

- Defining the join relationships between tables

Precision Settings dialog

This dialog appears when you click the System Default Settings button in the Advanced Export Settings dialog. It helps you to predefine the precision with which you want to export report results to different formats.



The precision level can be one of the following:

- **High**
  If checked, the report results will be exported to the specified format with high precision.
- **Low**
  If checked, the report results will be exported to the specified format with low precision.

OK

Applies the settings and closes the dialog.

Cancel

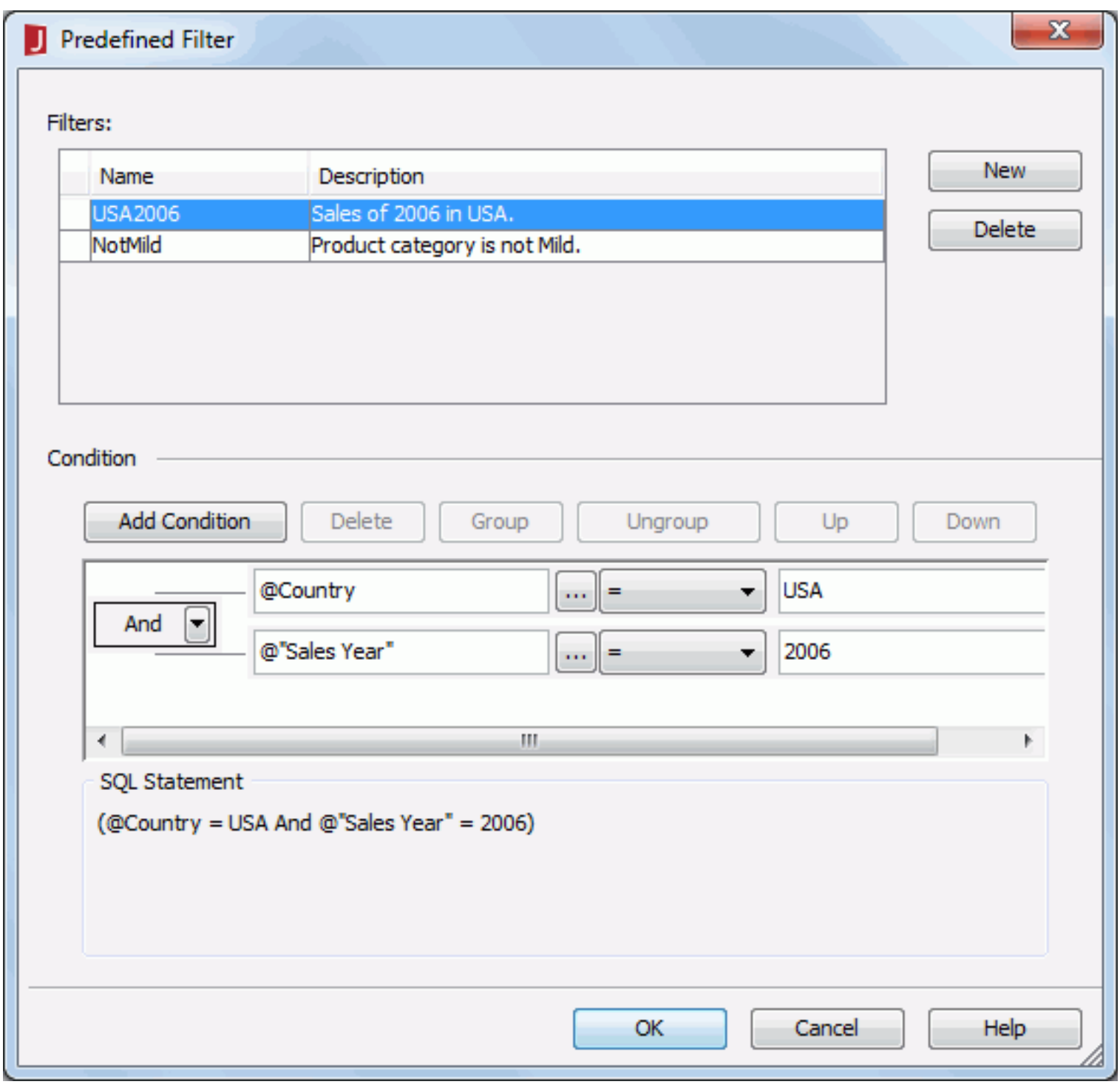
Does not retain any changes and closes the dialog.

Help

Displays the help document about this feature.

Predefined Filter dialog

This dialog appears when you click Tools > Predefined Filter on the menu bar or the Predefined Filter button on the toolbar of the Business View Editor window. It helps you to add some predefined filters to a business view. The predefined filters can be applied when creating/modifying reports based on the business view in JReport Server by end users.



The following are details about options in the dialog:

Filters

Lists all the predefined filters in the business view. You can modify a filter's name and description in the box.

- **New**
  Creates a filter.
- **Delete**
  Removes the selected filter.

Condition

Specifies the filter conditions.

- **Add Condition**
  Adds a new condition line.
- **Delete**
  Deletes the selected condition line.
- **Group**
  Makes the selected conditions in one group.
- **Ungroup**
  Makes the selected condition ungrouped.
- **Up**
  Moves the selected condition or group up to a higher level.
- **Down**
  Moves the selected condition or group down to a lower level.
- **Logic**
  Lists the logic operator.
  - **And**
    Logic operator And which is applied to this and the next line or group.
  - **Or**
    Logic operator Or which is applied to this and the next line or group.
- **Field**

  Specifies the field to be filtered. You can either type in the field directly or click ... next to the text box to specify the field.
- **Operator**
  Specifies the operator to compose the filter expression.
  - **=**
    Equal to
  - **> =**
    Greater than or equal to
  - **>**
    Greater than
  - **<**
    Less than
  - **< =**
    Less than or equal to
  - **!=**
    Not equal to
  - **[not] in**
    Causes an enumerated list of values to appear in the WHERE clause predicate, used for evaluating for a true condition. For operator 'in' or 'not in', it is allowed to input multiple values in the Value input box and separate the values with comma (,).
  - **[not] like**
    Like string pattern matching operator is used to compare the first expression string value to the pattern string (the second expression). If you want to use wildcard character in the pattern string, only "_" and "%" are supported.
  - **[not] between**
    Allows the system to evaluate whether or not data values are located between a range of values indicated in the predicate. For operator 'between' or 'not between', there are two value input boxes for inputting the same type of values.
  - **is [not] null**
    It is used in WHERE clause predicates to match null values occurring in a specified data field. For operator 'is null' or 'is not null', the value input box is hidden.
- **Value**

  Specifies the value of how to filter the field. You can either click ... next to the text box to specify the value or type the value in the text box manually.

  When you click ... to specify the value, the Values dialog is displayed.
  - **Fields tab**
    Lists the fields which can be used in the filter expression. Double-click the required field to add it as the filter value.
  - **Value tab**
    Lists values of the field to be filtered. Double-click the required value to add it as the filter value.

  When you type in the value, if multiple values are required, they should be separated with ",", and if "," or "\" is contained in the values, write it as "\," or "\\".
- **SQL Statement**
  Displays the SQL statement of the filter condition.

OK

Applies the changes and closes the dialog.

Cancel

Cancels the changes and exits the dialog.

Help

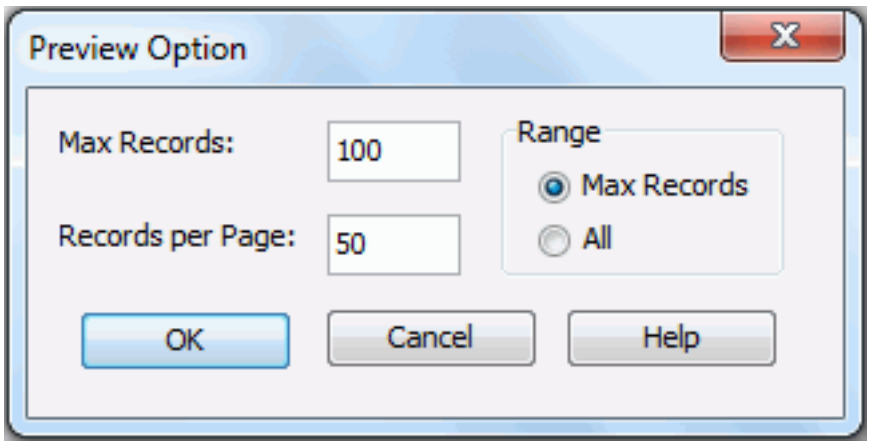
Displays the help document about this feature.

Related topics:

- Creating predefined filters

Preview Option dialog

This dialog appears when you click the Preview button on the toolbar of the Query Editor window. It helps you to set settings for previewing a query.



The following are details about options in the dialog:

Max Records

Specifies the maximum number of records that will be displayed when previewing the query.

Records per Page

Specifies the number of records that will be displayed on one page when previewing the query.

Range

Specifies the range of the records for previewing.

- **Max Records**
  If checked, the Max Records option will take effect.
- **All**
  If checked, all the records will be displayed when you preview the query, and the Max Records option will be disabled.

OK

Applies the changes and brings out the Preview dialog or the Enter Parameter Values dialog (if there are parameters in this query).

Cancel

Does not retain any changes and closes this dialog.

Help

Displays the help document about this feature.

Related topics:

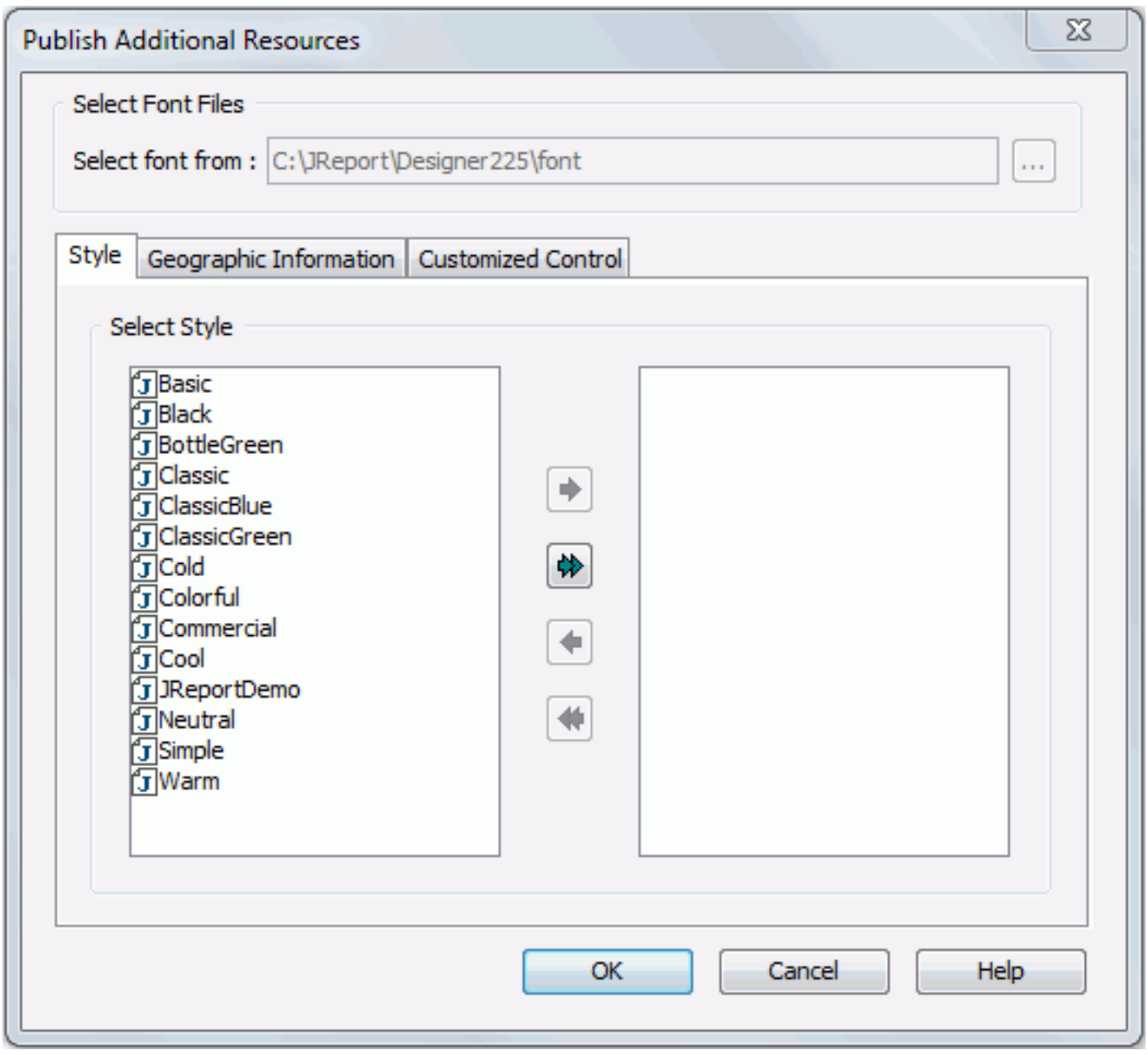
- Previewing a query

Publish Additional Resources dialog

This dialog is used to publish additional resources used by reports and library components to JReport Server. The resources include font files, style groups, geographic information, and customized control files.

The dialog is displayed when you do either of the following:

- On the File menu, click **Publish and Download** > **Publish to Server** > **Publish Other Files to JReport Server**.
- Click **OK** in the Publish to JReport Server dialog when you are publishing reports or library components that use special fonts, style groups, geographic information, or customized control files.
- Click **OK** in the Publish to JReport Server dialog and then click **Yes** in the message dialog which asks whether you would like to publish additional resources.



The following are details about options in the dialog:

Select Font Files

- **Select font from**
  Specifies the font file to publish. Available when special font is used.

Style tab

Specifies the style groups to publish.

Use the two right-headed-arrow buttons to add the selected style groups or all of them from the left list to the right list. Use the two left-headed-arrow buttons to remove the selected style group or remove all of them in the right list.

Geographic Information tab

Specifies the geographic information to publish.

Use the two right-headed-arrow buttons to add the selected xml file where geographic information is stored or all of them from the left list to the right list. Use the two left-headed-arrow buttons to remove the selected xml file or remove all of them in the right list.

Customized Control tab

Specifies the customized control files to publish.

Use the two right-headed-arrow buttons to add the selected customized control files or all of them from the left list to the right list. Use the two left-headed-arrow buttons to remove the selected customized control files or all of them from the right list.

OK

Publishes the specified resources.

Cancel

Does not publish additional resources.

Help

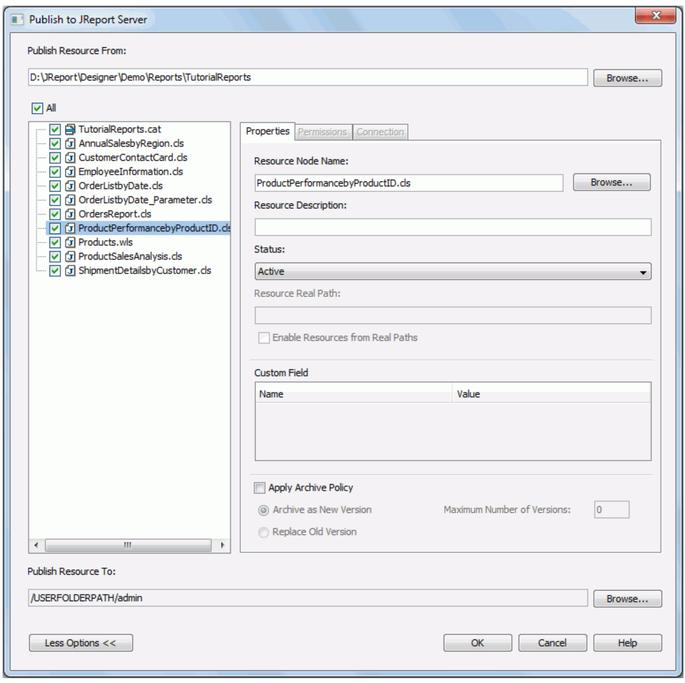
Displays the help document about this feature.

Related topics:

- Publishing additional resources

Publish to JReport Server dialog

This dialog appears when you click File > Publish and Download > Publish to Server > Publish Report to Server/Publish Component to Server on the menu bar, specify the information in the Connect to JReport Server dialog and then click Connect. It helps you to publish resources from JReport Designer to JReport Server.



The following are details about options in the dialog:

Publish Resource From

Specifies the directory in which the resources you want to publish are located. By default, the path of the currently opened catalog is shown here. You can click the Browse button to find another directory.

Select All

Specifies to publish all the resources in the specified directory.

Resource tree box

Lists all the resources in the specified directory in a tree structure. Check the checkboxes ahead of the resources you want to publish.

Publish Resource To

Specifies the file path on the server to which the resources will be published. You can click the Browse button to select the path in the Select Folder dialog.

More/Less Options

Shows/hides resource options. Select a to-be-published resource in the resource tree box and set its properties in the following tabs:

- **Properties tab**
  Specifies properties for the selected resource.
  - **Resource Node Name**
    Specifies the name of the selected resource. This name is required and is used as the display name of the resource in the server resource tree. The Browse button is used to display all the same type resource names under the Publish Resource To path. Select one of them as the name of the selected resource.
  - **Resource Description**
    Specifies the description for the selected resource.
  - **Status** (available only when a report is selected)
    Specifies the status of the selected report. If not specified, the status will be Active by default.
    - **Active**
      The report can be run, advanced run and scheduled on JReport Server.
    - **Inactive**
      The report cannot be run, advanced run or scheduled on JReport Server.
    - **Incomplete**
      The report is not completely designed and cannot be run, advanced run or scheduled on JReport Server.
  - **Resource Real Path** (available only when a folder is selected)
    Specifies the real path of the folder.
  - **Enable Resources from Real Paths** (available only when a folder is selected)
    Specifies whether or not to enable getting resources from the folder's real path. Once enabled, the real path resources will be mapped into the resource node of the folder in the server resource tree and the server will always be able to get the resources and updates from the real path.

    Note: This option can only be checked when the Enable Resources from Real Paths option on the JReport Server Administration page > Configuration > Advanced panel has also been checked.
- **Custom Field**
  Specifies custom field values for the selected resource if there are custom fields enabled on JReport Server. A custom field can be regarded as a resource property.
  - **Name**
    Displays name of the custom fields.
  - **Value**
    Specifies value of the custom fields.
- **Apply Archive Policy**
  Specifies whether to apply an archive policy to the selected resource.
  - **Archive as New Version**
    Specifies whether to use multiple versions for the selected resource.
    - **Max Number of Versions**
      Specifies the maximum number of versions that will be listed in the version table of the selected resource. The default value is 0, which means that the version amount is unlimited.
  - **Replace Old Version**
    Specifies whether to replace the old version when the new version is generated.
- **Permissions tab** (disabled when publishing resources to the My Reports or My Components folder)
  ecifies user permissions to the selected resource.
- **Enable user permission**
  Specifies whether to apply user permissions to the selected resource.
- **Role/User/Group**
  he tabs list JReport Server roles, users, and groups. You can apply permissions to them by selecting a role/user/group and then hecking the checkboxes before the required permissions. For unwanted roles/users/groups, you can remove them by selecting one of hem and then clicking the Remove button.
- **Permissions**
  ets permissions to the selected role/user/group by checking the checkboxes before the required permissions.

ermissions in JReport Server include:

Permission	Description
Visible	Allows or denies viewing object names in the resource tree or version table, such as folders, resources, and archive versions.
Read	Allows or denies viewing object properties, versions, and, if it is a folder, folder content.
Write	Allows or denies publishing folders and resources, changing the properties (not including permission settings) of the objects in the resource tree or version table, such as folders, resources, and archive versions, and modifying version table settings.
Delete	Allows or denies deleting objects in the resource tree or version table, such as folders, resources, and archive versions.
Execute	Allows or denies running resources in normal and advanced mode (report type resources only).
Schedule	Allows or denies submitting resources to schedules (report type resources only).
Grant	Allows or denies granting permissions to other users or roles.
Update Status	Allows or denies updating report status, and if it is a folder, the status of reports in the folder.

- **Add**
  dds roles/users/groups and assigns permissions to them.
- **Remove**
  emoves the selected role/user/group from the Role/User/Group tab. The removed roles/users/groups are still available for the case hen you want to add them back.
- **Connection tab** (only for catalog)
  ecifies properties of the data source connection.
- **Catalog Data Source**
  pecifies the data source in the catalog.
- **Modify**
  f the selected data source is built on a JDBC connection, the Modify button is activated. You can click it to modify properties of the data ource connection in the Get JDBC Connection Information dialog.
- **Name**
  ists the name of the connection properties.
- **Property**
  ists the value of the properties.

OK

Applies the changes and publishes the resources to JReport Server.

Cancel

Closes the dialog, leaving any changes unsaved.

Help

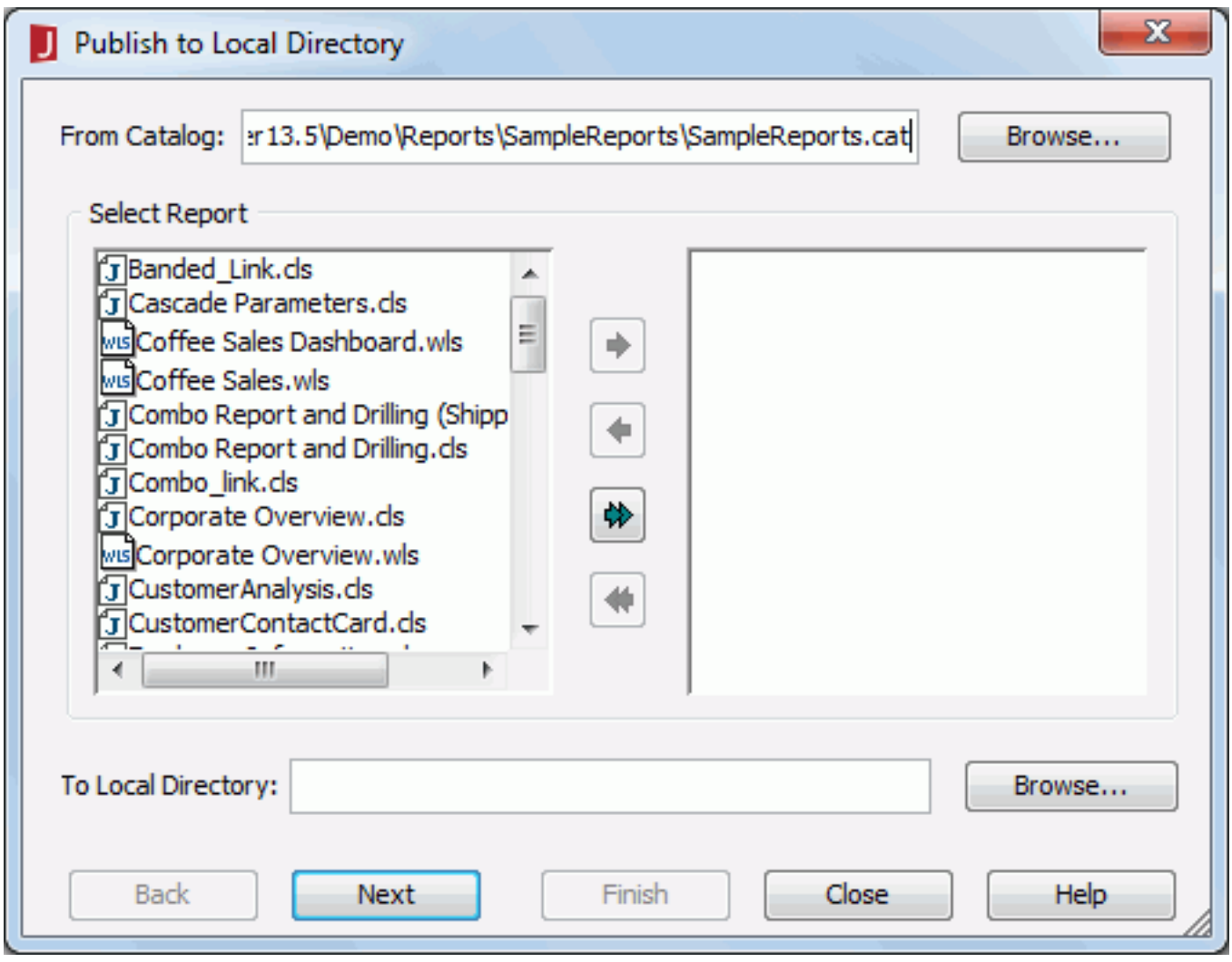
Displays the help document about this feature.

Related topics:

- Publishing resources remotely

Publish to Local Directory dialog

This dialog appears when you click File > Publish and Download > Publish to Local Directory on the menu bar. It helps you to publish catalogs and reports locally.



The following are details about options in the dialog:

From Catalog

Specifies the catalog of the report to be published. Input the name with its full path of the catalog file or click the Browse button to search it in the Open Catalog File dialog.

Select Report

Lists all the reports of the selected catalog. Select the reports you want to publish by highlighting them.

Adds the highlighted reports from the Select Report list to the list on the right.

Removes the highlighted reports from the list in the right to the Select Report list.

Adds all the reports in the Select Report list to the list in the right.

Removes all the reports from the list in the right to the Select Report list.

To Local Directory

Specifies the directory where the catalog and reports are to be published to. Type the directory in the text box or click Browse to find a directory in the Get Directory dialog.

Note: You can type a directory that does not exist, and JReport Designer will create the new directory automatically.

Back

Retains the changes and returns to the previous page.

Next

Retains the changes and go to the next page to view and modify the connection properties:

- **Catalog Data Source**
  Specifies the data source in the catalog.
- **Modify**
  If the selected data source is built on a JDBC connection, you can click this button to modify properties of the selected data source connection in the Get JDBC Connection Information dialog.
- **Name**
  Lists the name of the connection properties.
- **Property**
  Lists the value of the properties.

Finish

Saves the changes and makes JReport Designer publish the catalog and reports.

Close

Does not retain the changes and closes the dialog.

Help

Displays the help document about this feature.

Related topics:

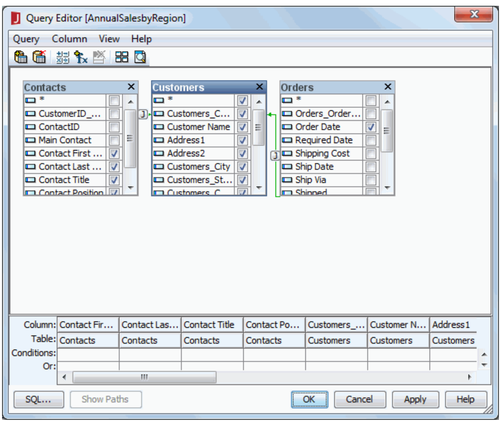
- Publishing resources locally

Query Editor window

The window appears when you do one of the following:

- In the Catalog Browser, right-click the Queries node, select Add Query from the shortcut menu, input a name for the new query and then click OK.
- In the Catalog Browser, right-click a query and click Edit Query on its shortcut menu.
- In the Catalog Browser, right-click the Business View node, select Add Business View from the shortcut menu, input a name in the displayed dialog and click OK, then in the Add Table/View/Query dialog, add more than one resource to use in the business view and click OK.
- Click Tools > Join Editor or click on the toolbar in the Business View Editor window.

It helps you to edit a query, or select the table columns and set joins for a business view.



The following are details about options in the window:

Query Menu

- **Apply**
  Accepts the changes on the query or business view.
- **Save As**
  Saves the query with the name you specify.
- **Add Table / View / Query**
  Opens the Add Table/View/Query dialog to add tables, views, synonyms, queries, imported SQLs, stored procedures, and user defined data sources in the current catalog data source to the query or business view. When a query, imported SQL, stored procedure or user defined data source is added, it will be added as a single table with all of its columns.
- **Delete Table / View / Query**
  Removes the selected resource.
- **Union**
  Creates and modifies a union for the query.
- **Filter**
  Opens the Search Condition dialog to specify more criteria to limit the amount of data.
- **Select Distinct**
  When selected, in the SQL statement, the SELECT DISTINCT command will be used instead of SELECT.
- **Ignore Predicate If Parameter Value Is Null**
  If the query or business view uses a parameter, and the parameter value is null at runtime, then this condition will be removed from the query's or business view's criterion.

  For string type parameters, when the value is left blank, if Ignore Predicate If Parameter Is Null is set to true, the value of this parameter will be considered as NULL, and this predicate will not appear in the WHERE clause; if false, it will be treated as an empty string(¡®¡¯).
- **Customized SQL Mode**
  Specifies whether to use the customized SQL mode, in which JReport will disable all the functions in the Query Editor and you can only edit the query by modifying the SQL statement in the SQL window.
- **Arrange Table / View / Query**
  Organizes the resources added into the query or business view.
- **Show Mapping Names**
  Specifies to show the full names of the columns in the tables in the criteria panel of the Query Editor.
- **Show Table Names**
  Specifies to show which table each column belongs to in the criteria panel of the Query Editor.
- **Show Paths**
  Specifies to show information of the pre-join paths among tables in the query, when the pre-join feature is enabled.
- **Join Columns**
  Specifies to join the highlighted columns in tables together.
- **Auto Join**
  Specifies to join tables together based upon the default criteria set in the Options dialog (**File** > **Options** > **Query Editor** > **Auto Join**). You can also reset the criteria to join tables in the query or business view by checking/unchecking the items on the Auto Join submenu.
  - **Foreign Keys**
    Specifies to automatically join tables in the query or business view through a reference from Table A to a primary key in Table B. For example, an order form in Table A shows information on purchases that are made by a customer. It can also show this customer's ID #. The customer ID # refers to a record in Table B which lists a specific address, phone number, name, and so on for the customer. The customer ID # in Table A is a foreign key because it links the customer's ordering information in Table A to the customer's information in Table B.
  - **Primary Key with Same Names**
    Specifies to automatically join tables in the query or business view through a field or a combination of fields that uniquely and specifically identifies a record. For example, your credit card number is a primary key because it specifically links to your birthday, credit history, address, and so on.
  - **Same Column Names**
    Specifies to automatically join tables in the query or business view through a link between two columns of the same name in two different tables.
- **Current Query Options**
  Opens the Query Option dialog to change Auto Join and showing options.

Column menu

- **Create Computed Column**
  Opens the Create Computed Column dialog to create functions used for a particular query. It helps you to quickly and easily create columns with the functions available instead of writing a formula.
- **Add Formula Field**
  Opens the Add Formula Field dialog to add formulas to the query or business view.
- **Delete Column**
  Deletes the selected column in the Criteria menu. To undelete, find the field in the table you deleted and place a check mark beside it.

View menu

- **Edit SQL**
  Shows the SQL statement of the query.
- **Preview**
  Opens the Preview Option dialog to preview the records.

Help menu

- **Query Editor Help**
  Displays the help document about this feature.

Toolbar

The following are commands on the toolbar:

- **Add Table / View / Query**
  Opens the Add Table/View/Query dialog to add tables, views, synonyms, queries, imported SQLs, stored procedures, and user defined data sources in the current catalog data source to the query or business view. When a query, imported SQL, stored procedure or user defined data source is added, it will be added as a single table with all of its columns.
- **Delete Table / View / Query**
  Removes the selected resource.
- **Create Computed Column**
  The create functions are used for a particular query. It helps you to quickly and easily create columns with the functions available instead of writing a formula.
- **Add Formula Field**
  Opens the Add Formula Field dialog to add formulas to the query.
- **Delete Column**
  Removes some unwanted columns from a table.
- **Arrange**
  Organizes the resources added into the query or business view.
- **Preview**
  Opens the Preview Option dialog to preview the records.

Criteria panel

Lists all columns added to the query. You can specifically define criteria for the columns in this panel.

SQL

Shows the SQL statement of the query.

Show Paths

Shows information of the pre-join paths among tables in the query, when the pre-join feature is enabled.

OK

Accepts all changes and closes the window.

Cancel

Does not retain any changes and closes the window.

Apply

Accepts all changes in the window.

Help

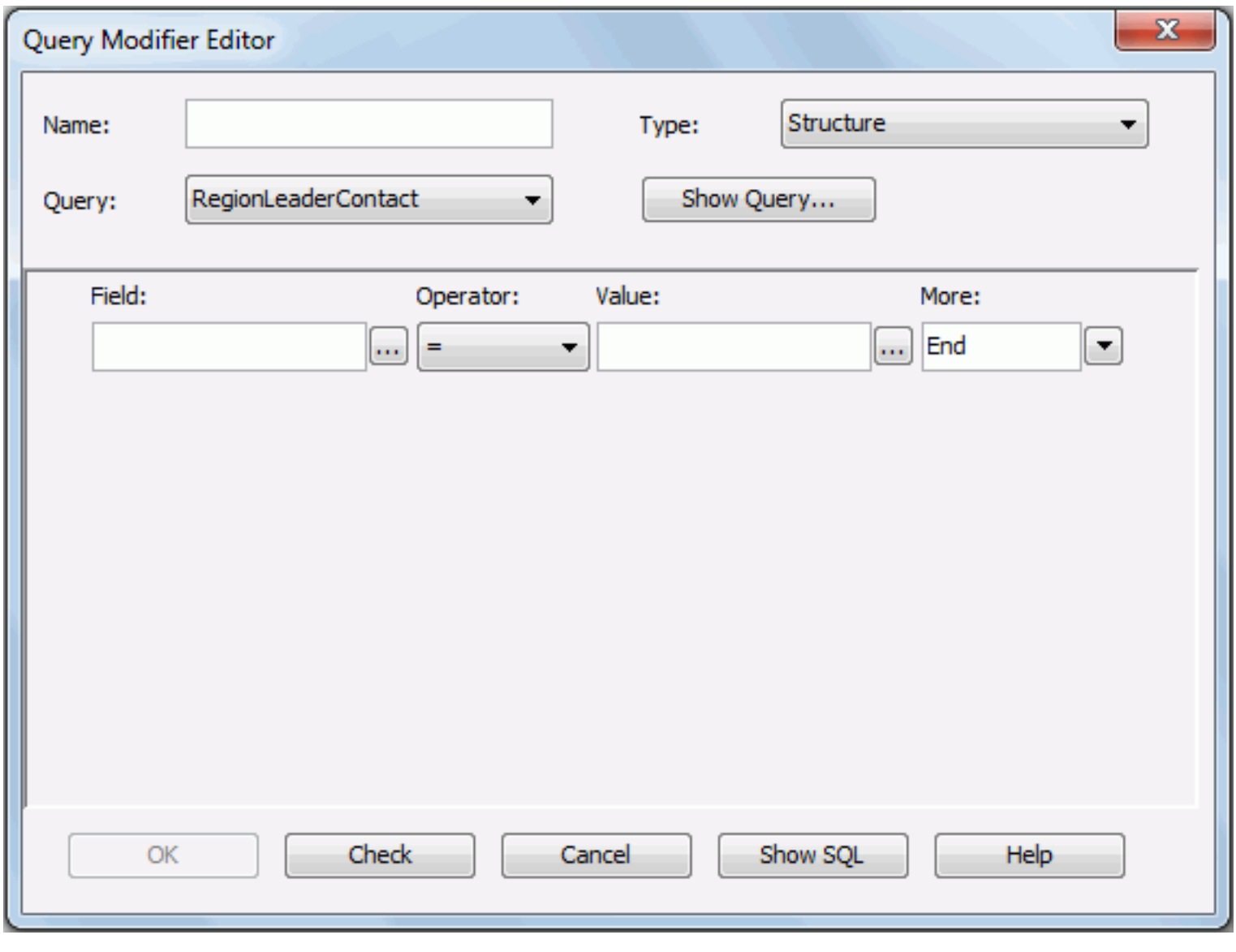
Displays the help document about this feature.

Related topics:

- Creating a query
- Editing a query
- Previewing a query
- Creating and modifying a union query

Query Modifier Editor dialog

This dialog appears when you right-click the Query Modifiers node and select Add Query Modifier from the shortcut menu in the Catalog Browser. It helps you to create a query modifier which can be specified via the Engine Bean in order to generate report on different data.



The following are details about options in the dialog:

Name

Specifies the name of the query modifier.

Type

Specifies the type of the query modifier, Structure or String.

Query

Specifies the query in the catalog to build or edit the query modifier.

Show Query

Shows the information of the query that is used.

Editing area

Specifies the query modifier in the editing area. When editing the String type query modifier, a text editor will be shown and you can key in the WHERE clause. When editing the Structure type query modifier, a condition editor will be displayed and you can construct the query modifier using the buttons provided.

OK

Closes this query modifier dialog and accepts the changes.

Check

Checks the validation of the query modifier according to the query. The validation of a query modifier is defined as: all the fields and formulas referred to by the query modifier are available to the chosen query. The validation check will also ensure that the query modifier can work with the specified query.

Cancel

Does not retain changes and closes the dialog.

Show SQL

Displays the string of the Query Modifier.

Help

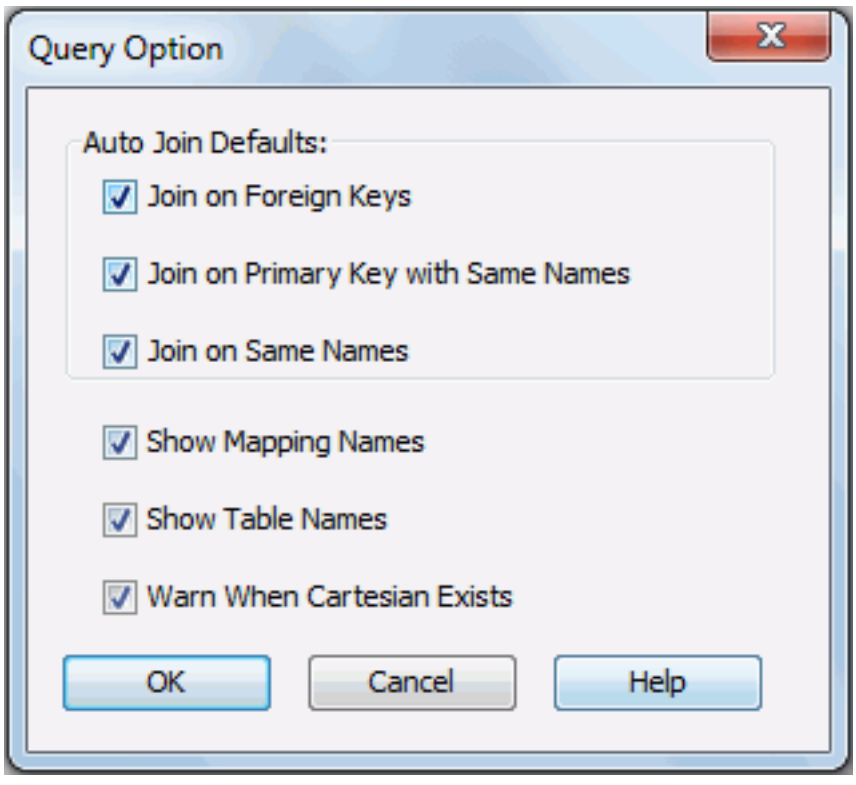
Displays the help document about this feature.

Related topics:

- Query modifiers

Query Option dialog

This dialog appears when you click Query > Current Query Option on the menu bar of the Query Editor window. It helps you to specify the settings for a query.



The following are details about options in the dialog:

Join on Foreign Keys

Specifies to automatically join tables in the query through a reference from Table A to a primary key in Table B. For example, an order form in Table A shows information on purchases that are made by a customer. It can also show this customer's ID #. The customer ID # refers to a record in Table B which lists a specific address, phone number, name, and so on for that customer. The customer ID # in Table A is a foreign key because it links the customer's ordering information in Table A to the customer's information in Table B.

Join on Primary Keys with Same Names

Specifies to automatically join tables in the query through a field or a combination of fields that uniquely and specifically identifies a record. For example, your credit card number is a primary key because it specifically links to your birthday, credit history, address, and so on.

Join on Same Names

Specifies to automatically join tables in the query through a link between two columns of the same name in two different tables.

Show Mapping Names

If selected, the full names of the columns in the tables will be displayed in the criteria panel of the Query Editor.

Show Table Names

If selected, the table that each column belongs to will be displayed in the criteria panel of the Query Editor.

Warn When Cartesian Exists

Specifies whether or not to display a warning when a Cartesian product exists. A Cartesian product is used when tables link together with no specifications.

For example, Table A has three values: A, B and C and Table B has three values: 1, 2 and 3. Value A matches value 1, value B matches value 2, and so on. This is a specific match.

A Cartesian product could have value A matching with 1, 2 and 3, and value B matching with 1, 2 and 3, and so on. Depending on the data values Cartesian products can produce large and complicated datasets since unnecessary information may be contained.

OK

Applies the settings for the query and leaves the dialog.

Cancel

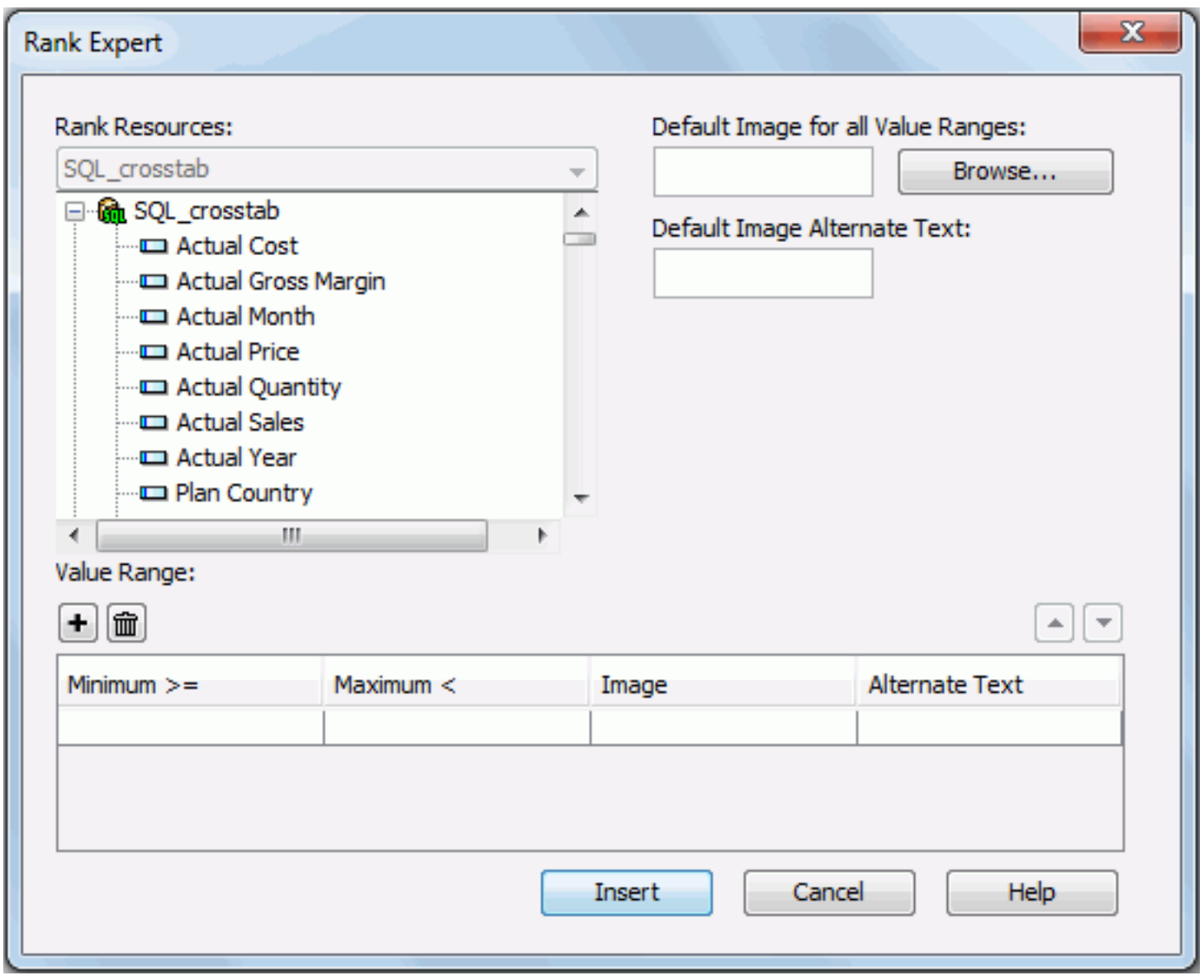
Does not retain the settings for the query and leaves the dialog.

Help

Displays the help document about this feature.

Rank Expert dialog

This dialog appears when you click Insert > Rank on the menu bar, or click the Insert Rank button on the Insert toolbar. It helps you to insert a data field, of which the display type is rank into a report.



The following are details about options in the dialog:

Rank Resources

Lists all resources of which display types can be set as rank.

Default Image for all Value Ranges

Specifies the default image for all value ranges, which will be applied when the value of a field does not fall into any of the defined ranges. Click Browse button to select the desired image.

Default Image Alternate Text

Specifies the alternate text for the default image, which will be shown when the default image cannot be displayed.

Value Range

Specifies the value ranges.

- Adds a new value range.
- Removes the selected value range.
- Moves the selected value range up a step.
- Moves the selected value range down a step.
- **Minimum (>=)**
  Specifies the minimum value for this range.
- **Maximum (<)**
  Specifies the maximum value for this range.
- **Image**
  Specifies the image of this value range, then when a field's value falls into this range, this image will be displayed.
- **Alternate Text**
  Specifies the alternate text for this value range's image, which will be shown when the image cannot be displayed.

Insert

Inserts the data field into the report as a rank display type.

Cancel

Does not retain changes and closes the dialog.

Help

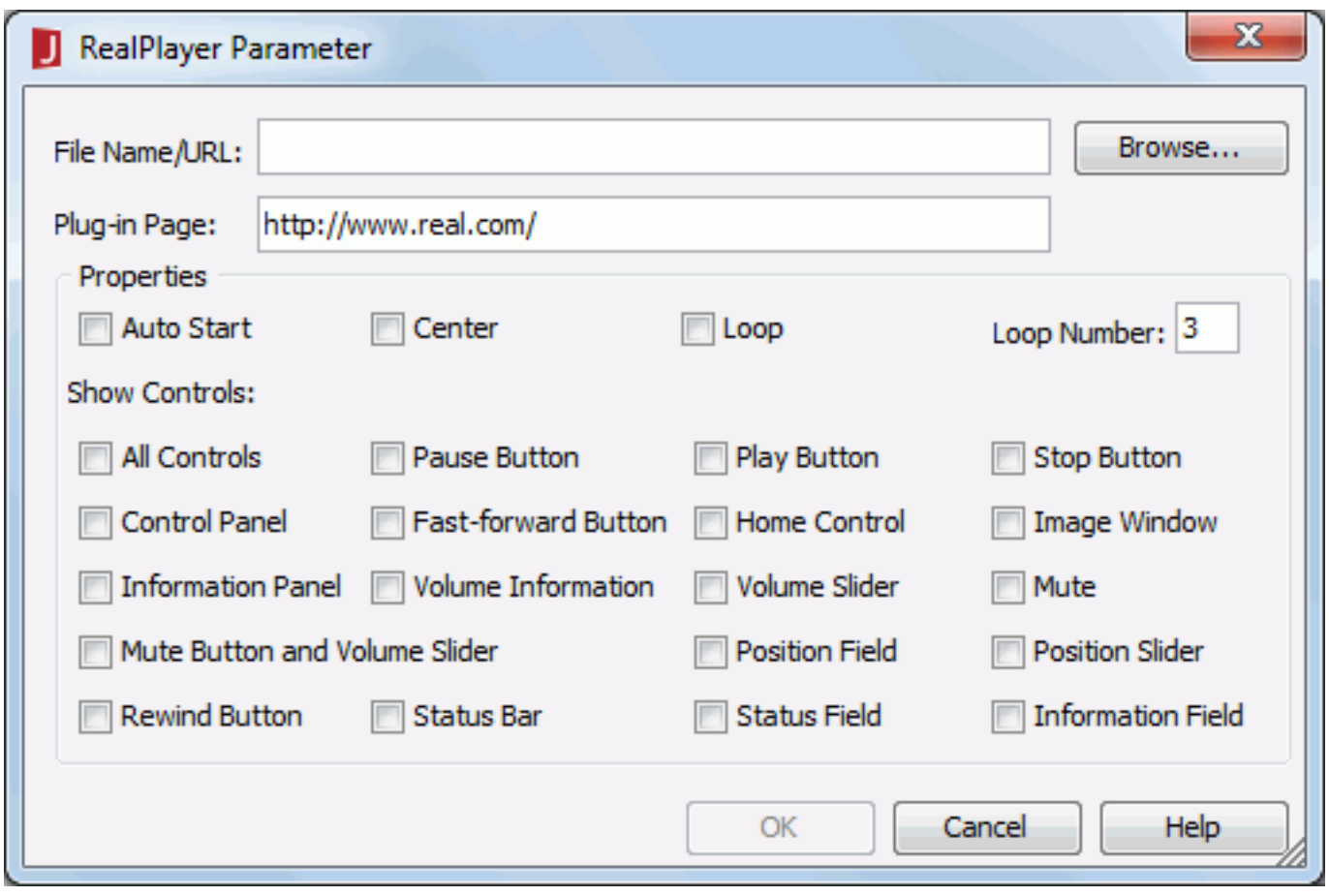
Displays the help document about this feature.

Related topics:

- Components - Ranks

RealPlayer Parameter dialog

This dialog appears when you click Insert > Multimedia > RealMedia File on the menu bar, or drag RealMedia File from the Toolbox panel to a report. It helps you to insert a RealMedia file into a report.



The following are details about options in the dialog:

File Name/URL

Specifies the name or the URL of the media file.

Plug-in Page

Specifies the URL of the plug-in page from which you can download the player with which to play the media file, if it isn't already installed on your local disk.

Properties

Specifies properties for the media file to control the options when playing the media file.

- **Auto Start**
  Specifies whether to start the media file automatically when the report is opened in Page Report Studio or Web Report Studio.
- **Center**
  Specifies whether to place the presentation in the center of the image window and display it in its original size (the size is determined by the Width and Height properties of the multimedia object in the Report Inspector).
- **Loop**
  Specifies whether to play the media file repeatedly.
- **Loop Number**
  Specifies the number of the times the media file will loop during playback.

  > **Note:** If you have checked the Loop option and specified the Loop Number at the same time, the media file will be played back according to the specified number, which means the Loop option will be ignored even though the loop number is set to zero.

- **Show Controls**
  Specifies the RealPlayer controls you want to show, which support specific RealPlayer functionality. The controls will be reproduced in the report as they appear in RealPlayer.
  - **All Controls**
    Specifies whether to show all controls. When it is checked, all the following controls will be selected automatically.
  - **Pause Button**
    Specifies whether to show the Pause Button.
  - **Play Button**
    Specifies whether to show the Play Button.
  - **Stop Button**
    Specifies whether to show the Stop Button.
  - **Control Panel**
    Specifies whether to show the Control Panel, which contains the following playback controls: Play Button, Pause Button, Stop Button, Fast-forward Button, Rewind Button, Position Slider and Home Button.
  - **Fast-forward Button**
    Specifies whether to show the Fast-forward Button.
  - **Home Control**
    Specifies whether to show the Home Button, which is connected to the www.real.com website.
  - **Image Window**
    Specifies whether to show the Image Window, which is used for displaying presentations.
  - **Information Panel**
    Specifies whether to show the Information Panel that displays the title, author, and copyright for the currently playing clip.
  - **Volume Information**
    Specifies whether to show the Information/Volume Bar, which consists of the Information Panel and the Mute/Volume Bar.
  - **Mute Button and Volume Slider**
    Specifies whether to show the Mute Button and Volume Slider. If checked, the Volume Slider and Mute checkboxes will be checked correspondingly. Otherwise, you can check Volume Slider or Mute respectively.
  - **Position Field**
    Specifies whether to show the Position Field, which shows the position of the current clip that identifies the clip's current place within the presentation timeline and the total clip length.
  - **Position Slider**
    Specifies whether to show the Position Slider, which shows the currently playing position within the clip.
  - **Rewind Button**
    Specifies whether to show the Rewind Button.
  - **Status Bar**
    Specifies whether to show the Status Bar, which consists of a text message area, the network congestion LED, and the current clip position indicator.
  - **Status Field**
    Specifies whether to show the Status Field, which consists only of a text message area.
  - **Information Field**
    Specifies whether to show the Information Field, which displays the title, author, and copyright for the currently playing clip or portion of a multi-clip.

OK

Accepts all changes and closes the dialog.

Cancel

Does not retain any changes and closes the dialog.

Help

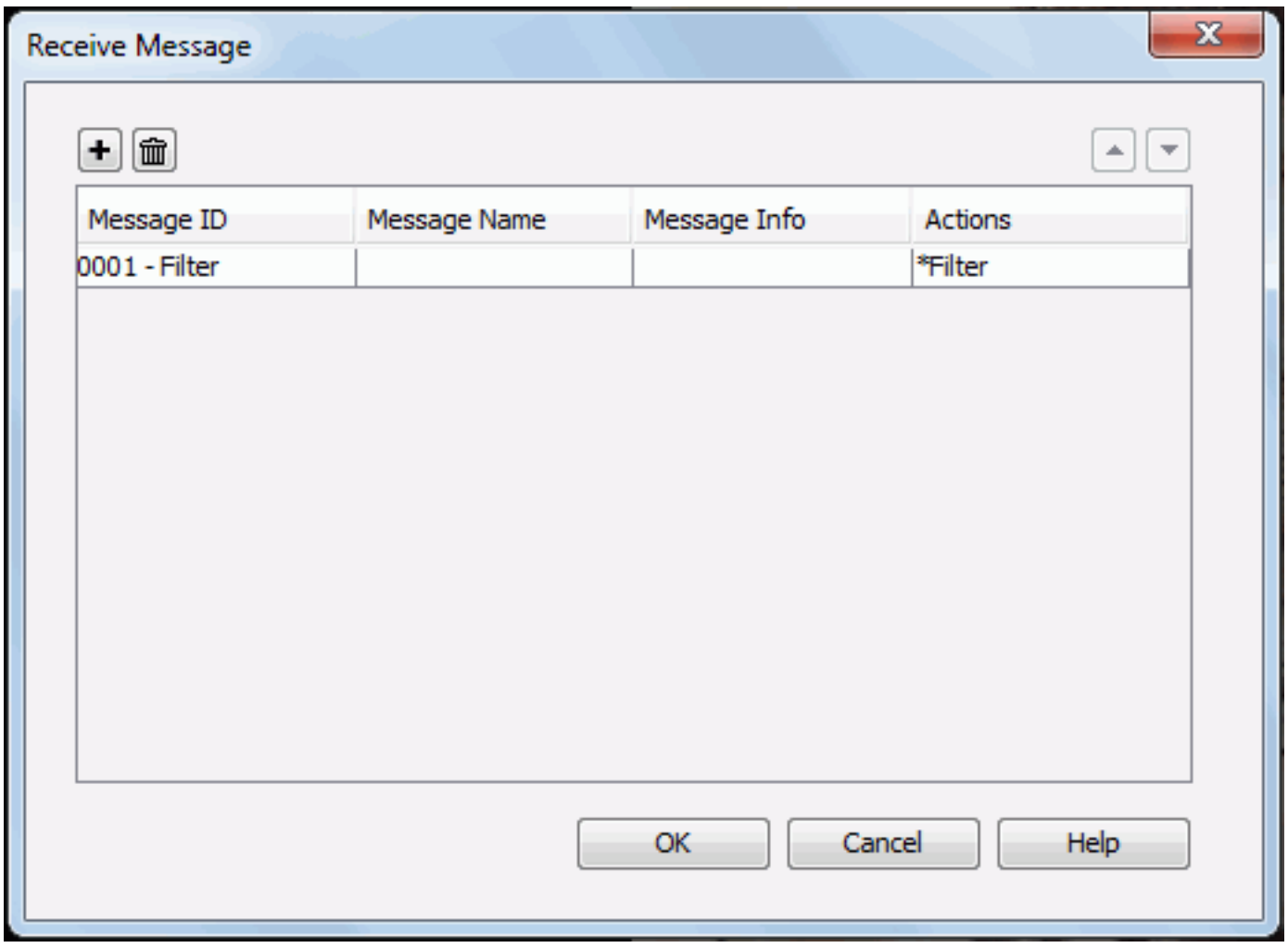
Displays the help document about this feature.

Related topics:

- Multimedia objects

Receive Message dialog

This dialog appears when you right-click a table, chart, crosstab or geographic map in a library component and select Receive Message from the shortcut menu. It helps you to define the messages the data component is going to receive and specify the actions that will be triggered as a response to each of the messages it receives at runtime.



The following are details about options in the dialog:

Adds a new message line.

Removes the selected message line.

Moves the selected message line up a step.

Moves the selected message line down a step.

Message ID

Specifies the ID of the message to be received. You can select the ID from the drop-down list or input it into the text field manually.

Message Name

Specifies the name of the message to be received.

Message Info

Available to the 0003 - Parameter message only. When Message ID is 0003 - Parameter, you need to choose a message type from the Message Info column: Automatic or Customized.

Actions

Specifies the action to respond to the message to be received.

-
  Opens the Web Action List dialog when you input a message ID in the Message ID column, or opens the corresponding dialog when you select a message ID from the Message ID drop-down list for you to specify the action you want to trigger when the message is received.

OK

Accepts the changes and closes this dialog.

Cancel

Does not retain any changes and closes the dialog.

Help

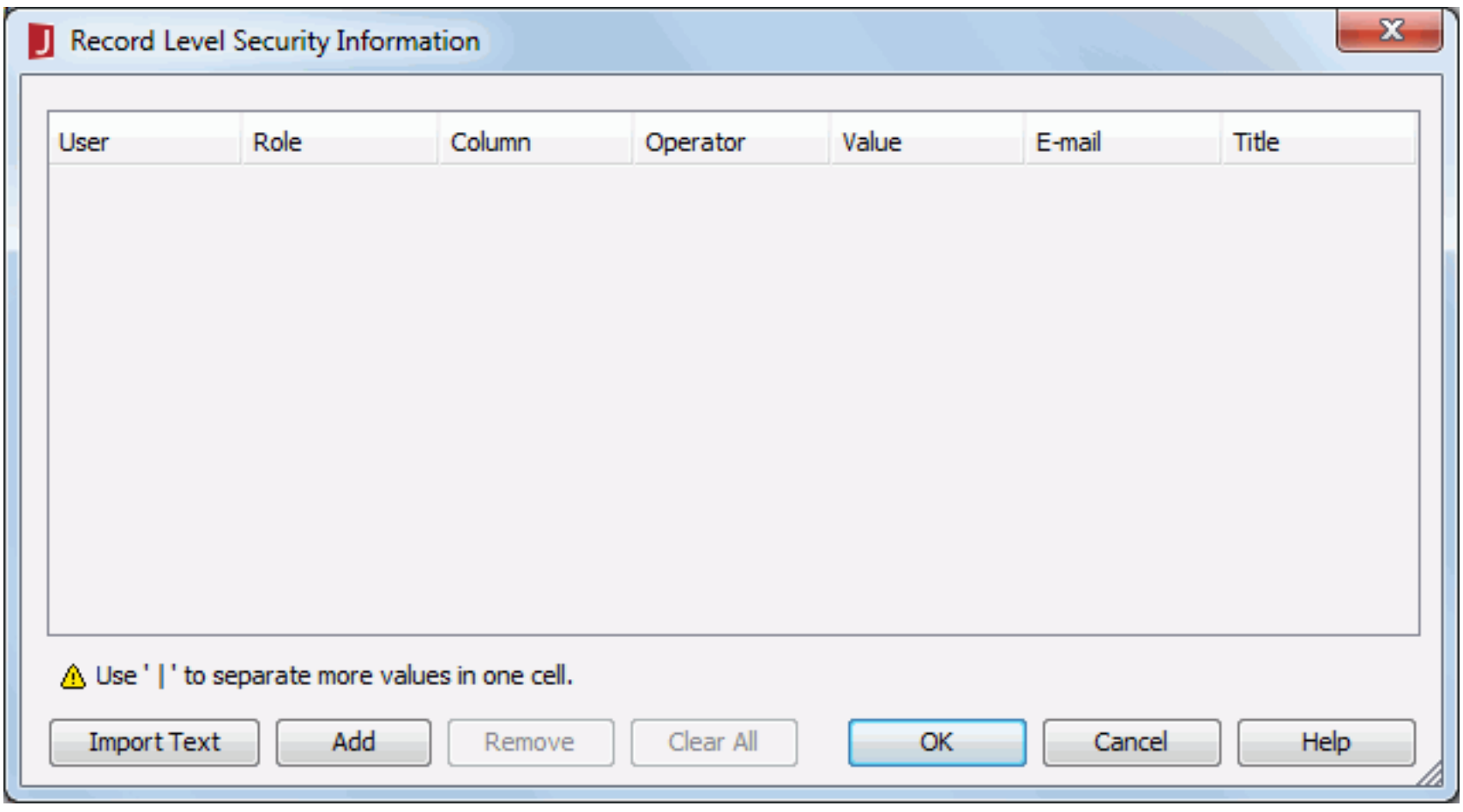
Displays the help document about this feature.

Related topics:

- Receiving messages

Record Level Security Information dialog

This dialog appears when you click ... in the Report Inspector to specify the property Record Security of a dataset. It helps you to change the record level security settings.



The following are details about options in the dialog:

Use "|" to separate more values in one cell

If you want to use more values in one cell, use "|" to separate them. Applies to User, E-mail, and Title column. This is useful when you want to apply the same conditions to multiple users. For example, if you want user1, user2, and user3 to share the same security setting, list user1, user2, and user3 in the same cell, separated by "|", and then define the security conditions.

Columns

- **User**
  The user ID to whom the security settings will be applied.
- **Role**
  The role of the user. More than one user can share one role; for one role, you only need to define once, and other users can share this role without type the same conditions again.
- **Column**
  The condition expression is comprised of three parts - Column, Operator, and Value.

  The column name, can be the name of a DBField in the current query, the name of a formula (summary) based on any DBField in the current query, and the name of a parameter. You do not need to quote the name when the name contains spaces. However, you must make sure the names you type actually exist.
- **Operator**
  The operator can be one of the following operators: =, <>, <, >, <=, >=, IN or NOT IN.
- **Value**
  The value part of the condition expression.
  - For String type values, use single quotation mark (') to quote the values, for example, 'Absolute Java';
  - For Boolean type values, use 0 (false) or 1 (true).
  - For Date type values, make sure the format of the value you enter is consistent with that of your database.

    Note: In the Record Level Security Information dialog, the date values provided on the drop-down list may not be valid for your actual database, because they are date values that have already been reformatted using your date format settings in JReport. For detailed information on how to set the date format in JReport Designer, see Date Format.
- For the values of other types, type them in their original form.
- **E-mail**
  e e-mail address of the user.
- **Title**
  e title for the user.

Import Text

Imports record level security settings from an external .txt file.

Add

Inserts a new row of user condition after the selected row.

Remove

Removes the selected row of user condition.

Clear All

Clears all the rows in the list.

OK

Accepts the changes and closes this dialog.

Cancel

Closes this dialog, leaving any changes unsaved.

Help

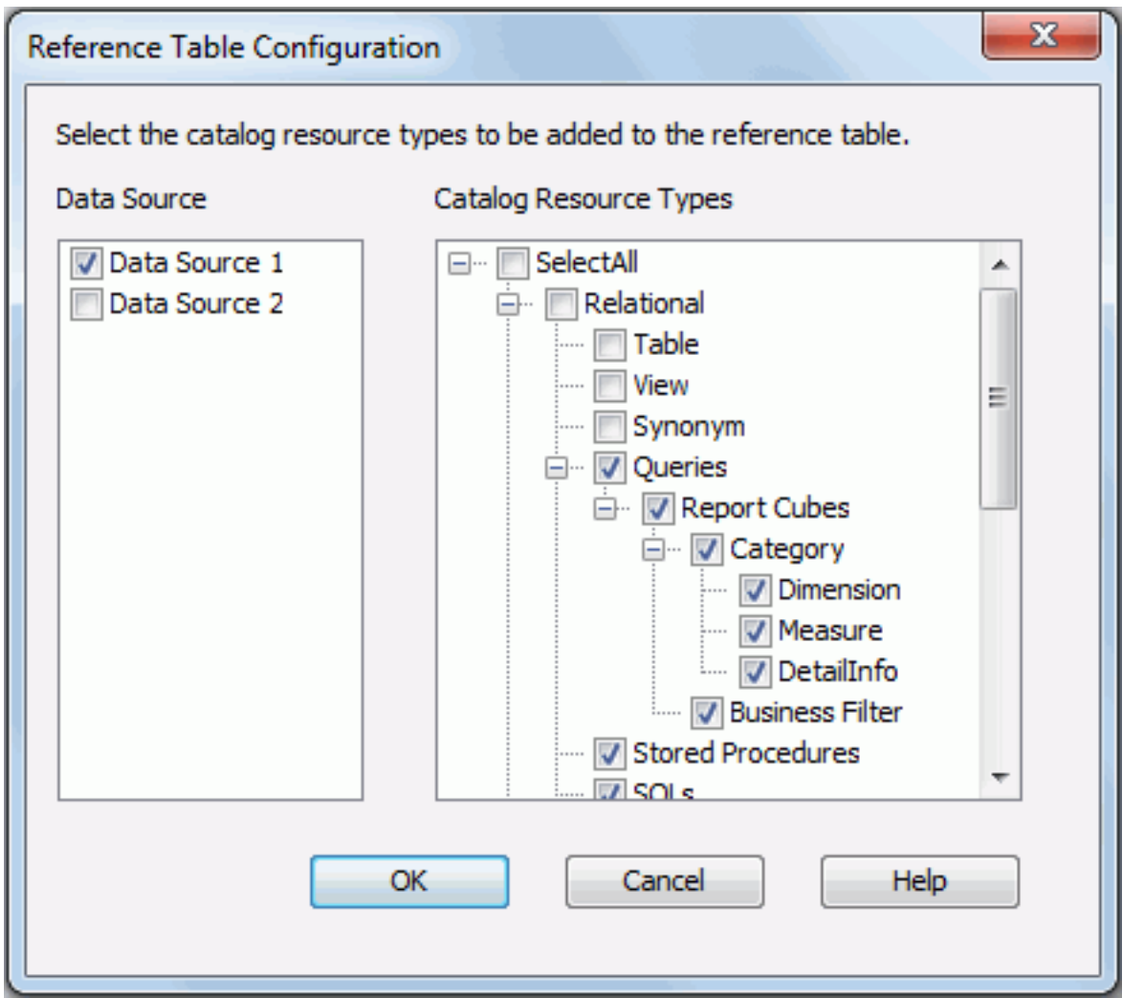
Displays the help document about this feature.

Related topics:

- Report-scope security

Reference Table Configuration dialog

This dialog appears when you click the Configure Reference button on the Catalog Browser toolbar. It helps you to select the catalog reference types to be added into the reference table.



The following are details about options in the dialog:

Data Source

Lists all the data sources available in the catalog.

Catalog Resource Type

Lists all the resources types available for you to select to add into the reference table.

OK

Closes the dialog and applies the changes.

Cancel

Closes the dialog, leaving the changes unsaved.

Help

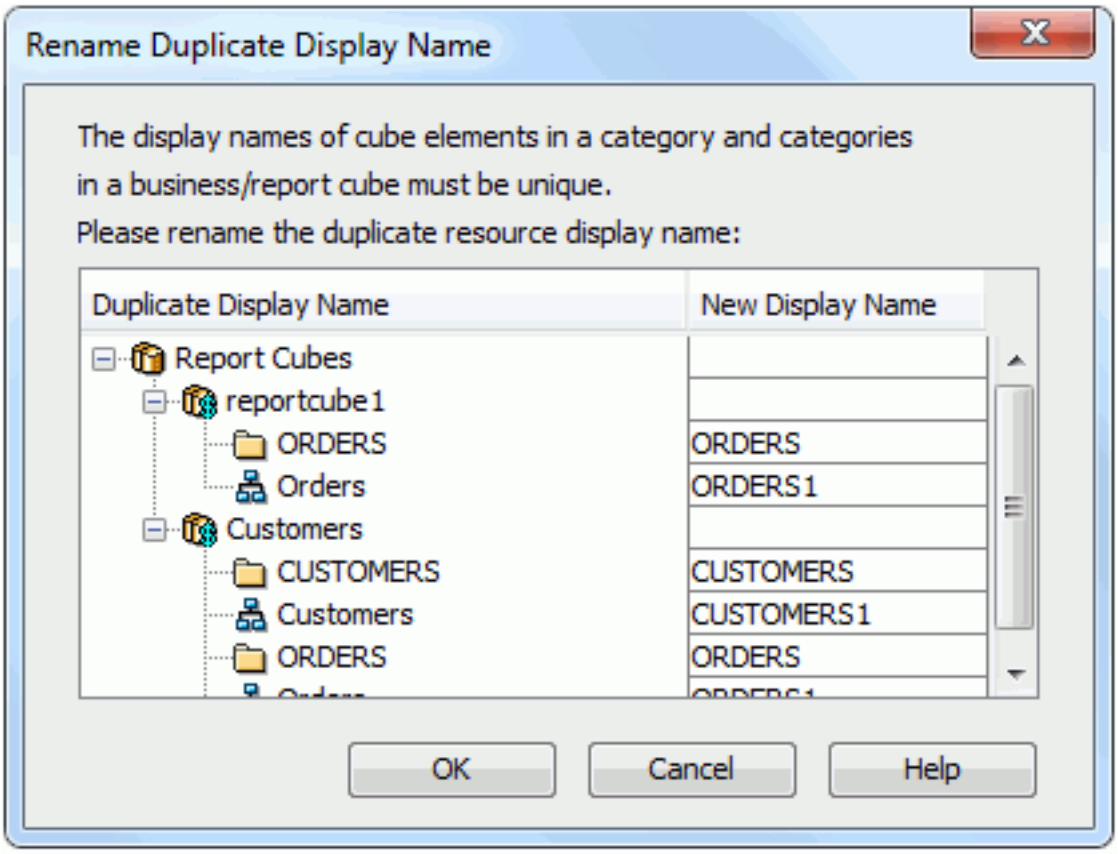
Displays the help document about this feature.

Related topics:

- Using a reference table to clarify resource relationships

Rename Duplicate Display Name dialog

This dialog appears when you load a catalog which was created in a previous version and contains business/report cubes that have duplicate display names. It helps you to rename the duplicate names.



The following are details about options in the dialog:

Duplicate Display Name

Lists all the cube elements and categories that have duplicate display names.

New Display Name

Specifies the new display names for the cube elements and categories.

OK

Applies the changes and closes the dialog.

Close

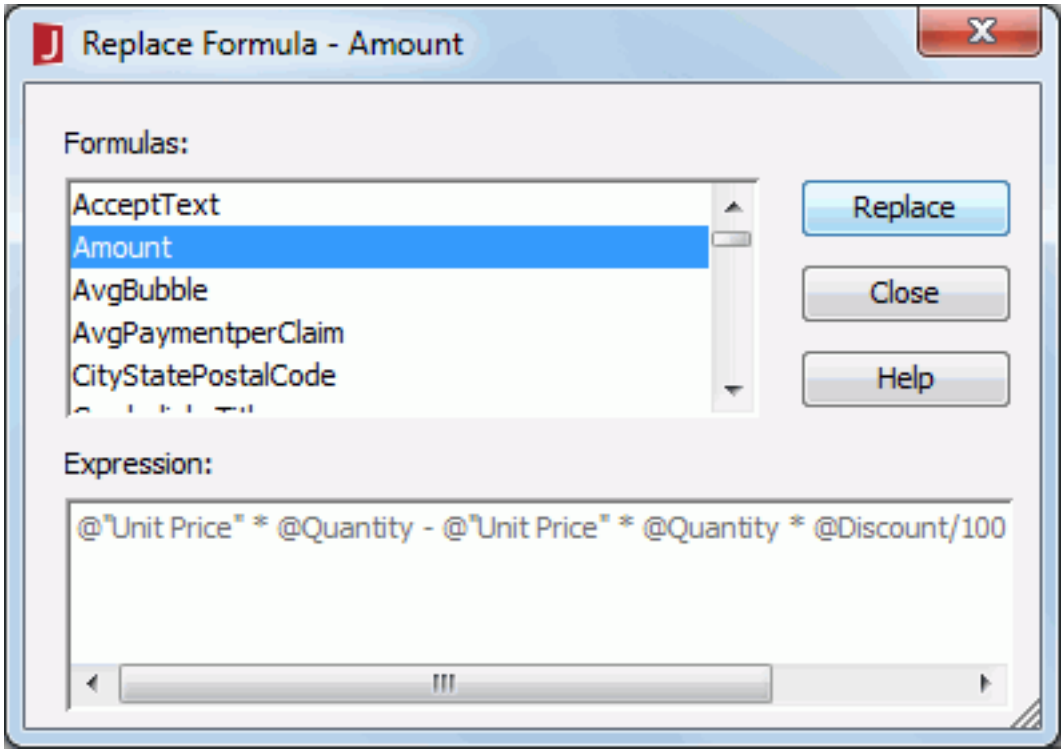
Does not retain any changes and closes the dialog.

Help

Displays the help document about this feature.

Replace Formula dialog

This dialog appears when you double-click any formula added to be column of a query in the Query Editor dialog. It helps you to select another formula to replace the current one.



The following are details about options in the dialog:

Formulas

Lists all the formulas available to be added as query columns.

Expression

Shows the expressions of the selected formula.

Replace

Replaces the current formula with the selected one.

Close

Closes the dialog.

Help

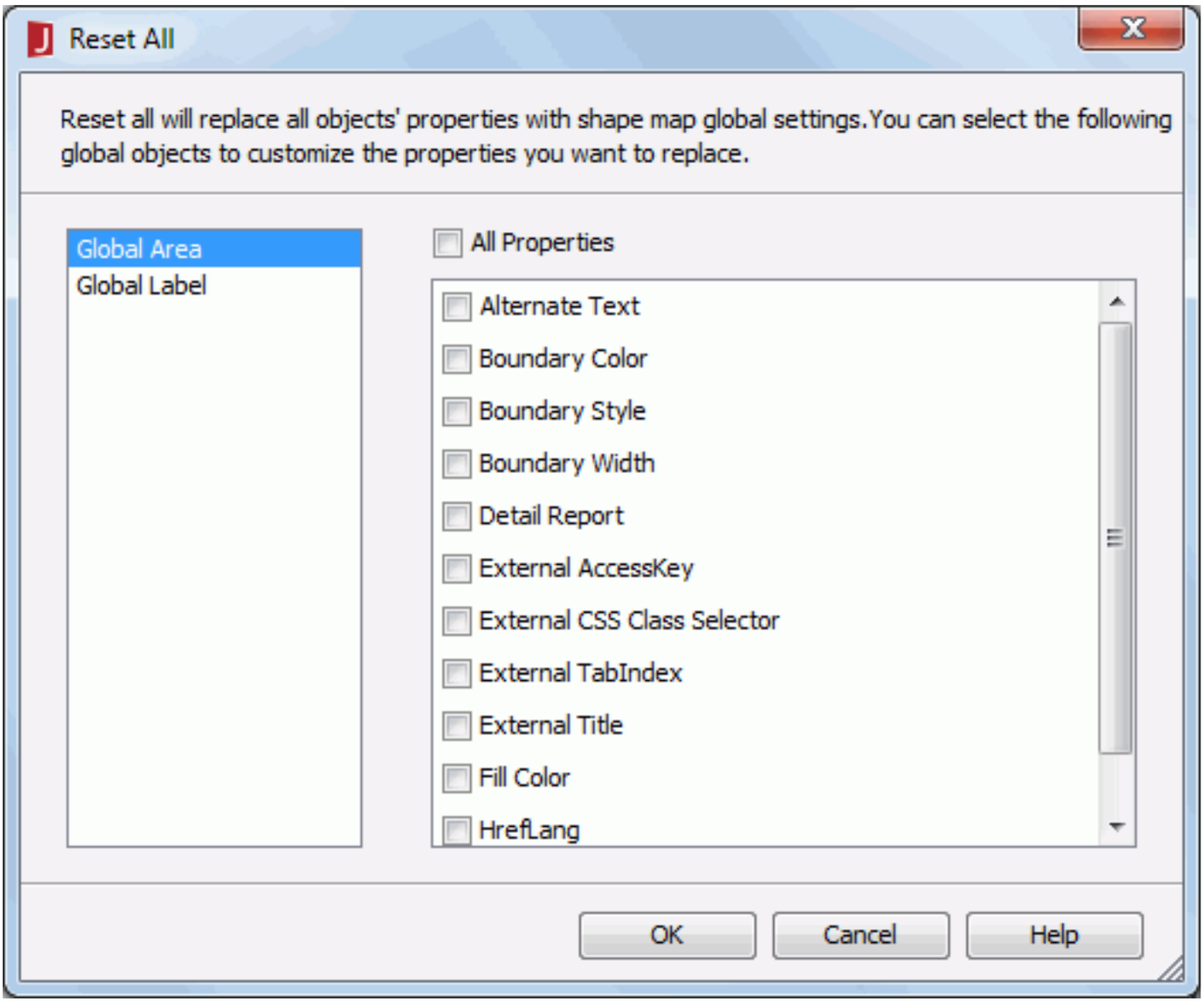
Displays the help document about this feature.

Related topics:

- Adding formula fields to a query

Reset All dialog

The dialog appears when you select Reset All from the Edit menu in the Shape Map Editor window. It helps you to reset properties of all areas and labels in a shape map to the global settings.



To reset properties of all areas to the global area settings, select **Global Area** in the left box, then in the right box, check the checkboxes ahead of the required properties. Then after you click OK in the dialog, the selected properties of all areas in the map will be reset to the corresponding global area settings, which are specified on the Global Area object in the Map Area Inspector tree. If you want to reset all properties of the areas to the global settings, check the **All Properties** checkbox. After resetting all properties to the global settings, the Use Global Setting property of all areas will be changed to true.

Do the same if you want to reset properties of all labels to the global label settings.

OK

Applies the changes and closes the dialog.

Cancel

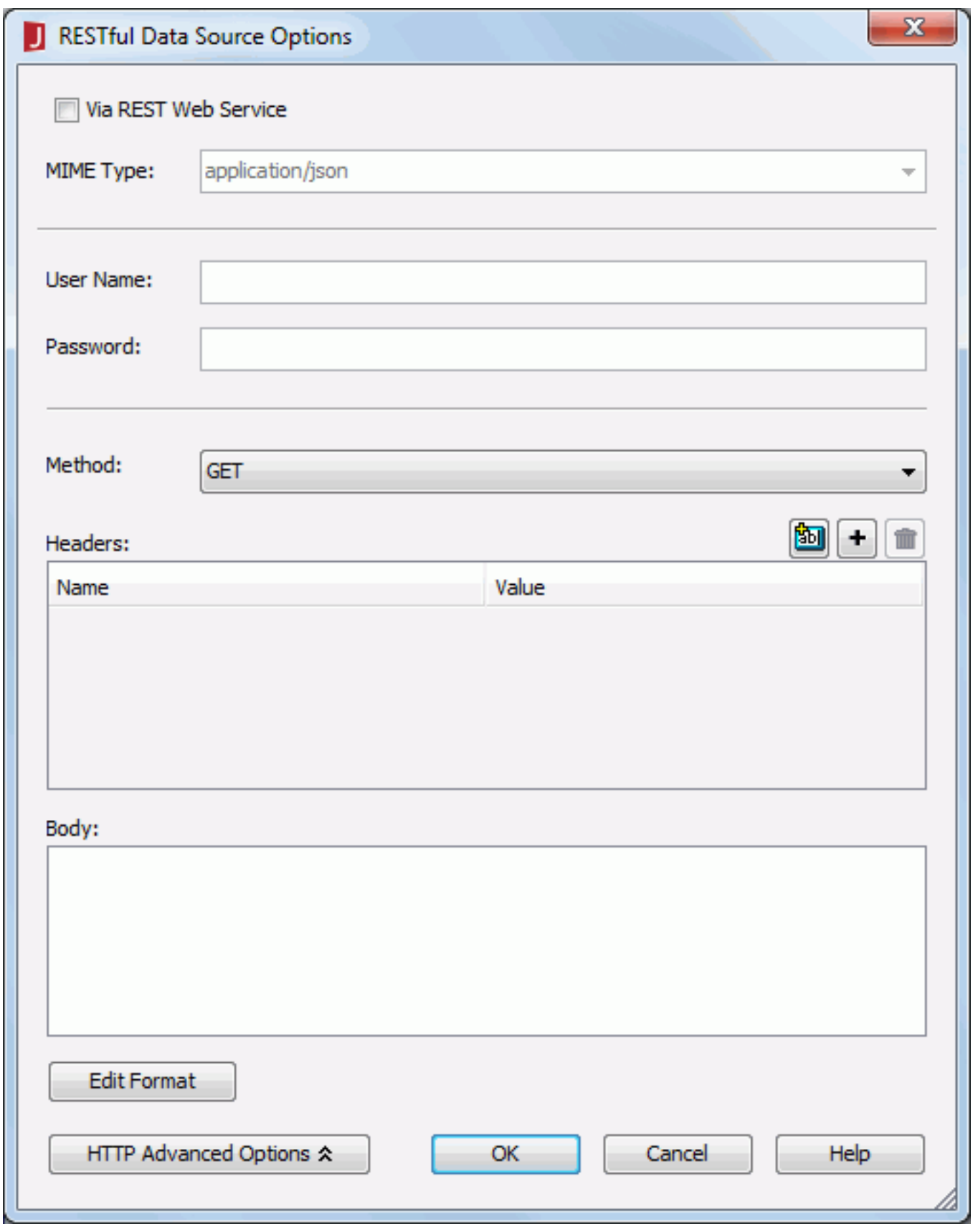
Does not retain any changes and closes the dialog.

Help

Displays the help document about this feature.

RESTful Data Source Options dialog

This dialog appears when you click the RESTful button in the JSON Connection Wizard. It helps you to receive remote data via the REST web service.



The following are details about options in the dialog:

Via REST Web Service

Specifies whether or not to receive remote data via the REST web service. It is disabled for a local file. If not, remote data is received via the protocol in the URL. If true, the remote data is provided by REST web services on the application server. The engine will use the REST web service client API (such as the JAX-RS client API of Java EE) to get remote data from REST web services on the application server and the URL, the HTTP method (GET, POST etc.), MIME types and so on need to be specified to setup the connection. Generally only the HTTP.GET method can be used to get remote data.

MIME Type

Specifies the MIME type for REST web service data sources. Select the type from the drop-down list or input the type in the text box. Available only when Via REST Web Service is true.

User Name

Specifies the user name for remote data authentication.

Password

Specifies the password for remote data authentication.

HTTP Advanced Options

Click to show/hide the advanced options for HTTP.

- **Method**
  Select an HTTP method from the drop-down list to send the request, which can be GET or POST.
- **Headers**
  Specifies the user defined HTTP headers.
  - Opens the Create Parameter dialog to create a parameter.
  - Adds a new HTTP header.
  - Removes the selected HTTP headers.
  - **Name**
    Specifies the name of the HTTP header. Double-click in the line to edit the name.
  - **Value**
    Specifies the value of the HTTP header. You can use parameters or constant level formulas predefined in the current catalog data source or the special field User Name in the URI or parameter string, and the format of the predefined parameters used in the string is "@" + parameter name. For example, if the parameter string of a class is SQL=select * from employee, and you want to use the parameter sql in a catalog to replace the part after "=" in the string, then the parameter string of the class will be SQL=@sql. Moreover, if a parameter string contains characters, such as @, '.' or double quotation marks, or other strings that do not need to be parsed, quote the whole string with double quotation marks. Double-click in the line to edit the value.
- **Body**
  Specifies the user defined HTTP body. You can use parameters or constant level formulas predefined in the current catalog data source or the special field User Name in the URI or parameter string, and the format of the predefined parameters used in the string is "@" + parameter name. For example, if the parameter string of a class is SQL=select * from employee, and you want to use the parameter sql in a catalog to replace the part after "=" in the string, then the parameter string of the class will be SQL=@sql. Moreover, if a parameter string contains characters, such as @, '.' or double quotation marks, or other strings that do not need to be parsed, quote the whole string with double quotation marks.
- **Edit Format**

  Opens the Edit Format dialog to specify the format of the values referenced in the HTTP headers or body.
  - **Name**
    Displays names of the referenced parameters/formulas.
  - **Format**
    Specifies formats for the referenced parameters/formulas.
  - **OK**
    Accepts changes and closes the dialog.
  - **Cancel**
    Does not retain changes and closes the dialog.

OK

Applies the changes and closes the dialog.

Cancel

Cancels the changes and exits the dialog.

Help

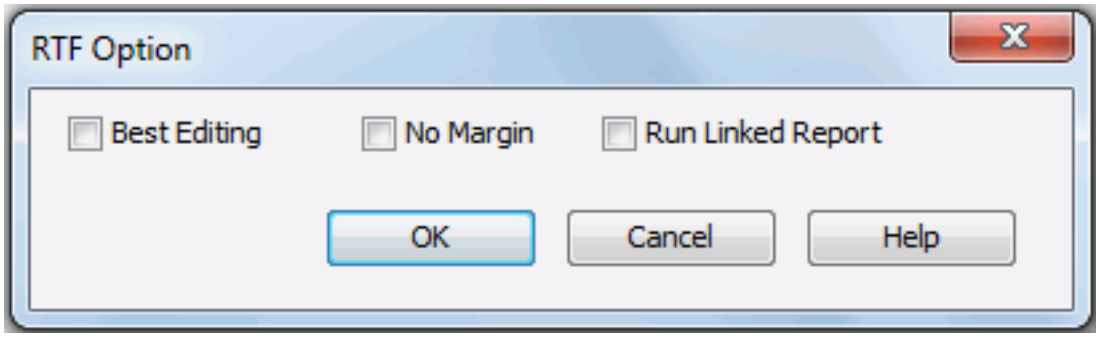
Displays the help document about this feature.

Related topics:

- Setting up a JSON connection

RTF Option dialog

The dialog appears when you set the Default Format for Viewing Report property of a report tab or a web report to RTF and then click ... in the value cell in the Report Inspector. It helps you to specify the parameters for running a report in RTF format.



The following are details about options in the dialog:

Best Editing

Specifies whether to apply flow layout for the RTF result.

No Margin

Removes the margins you originally set while designing the report.

Run Linked Report

If the report is linked with other reports, you can check whether or not to generate the linked reports in the RTF result. If you are only interested in the primary report, leave this option unchecked. Generating linked reports at the same time, especially when the linked reports contain a large amount of data, will cause performance issue.

OK

Applies the changes and closes the dialog.

Cancel

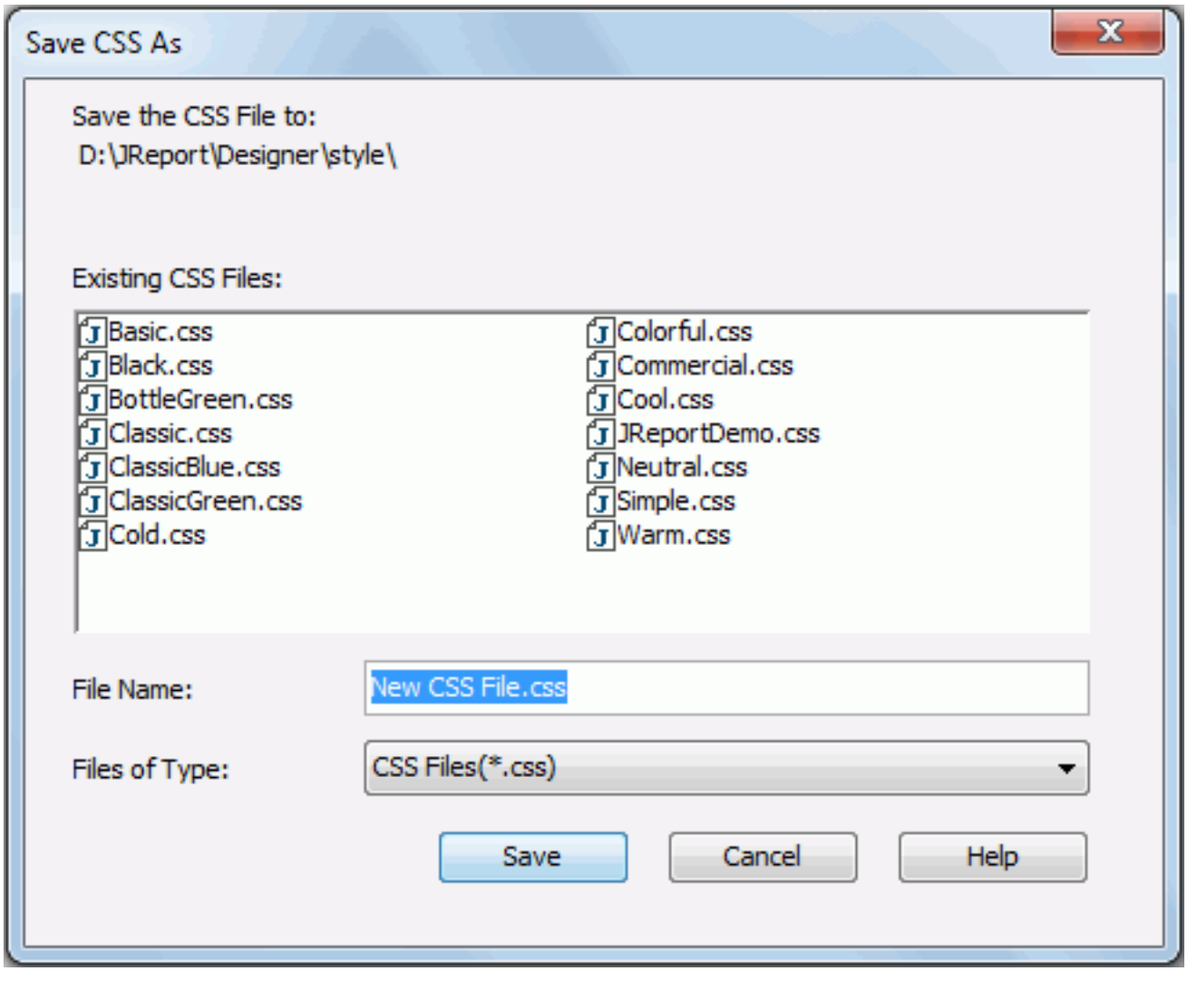
Cancels the changes and exits the dialog.

Help

Displays the help document about this feature.

Save CSS As dialog

This dialog appears when you click OK in the New CSS Style dialog if you select New Style Sheet File from the Define in drop-down list. After building a report, if you are pleased with the format of certain object, you can save it in order to use the same format again in other reports, without having to set all the properties again.



The following are details about options in the dialog:

Save the CSS File to

Shows the directory that the CSS File will be saved to.

Existing CSS Files

Lists all the existing CSS files.

File Name

Specifies the name for the new CSS file. By default, it is New CSS File.css.

Files of Type

Specifies the type for the new file. Here it should be CSS File(*.css).

Save

Saves the CSS style and goes into the CSS Style Definition dialog.

Cancel

Does not retain any changes and closes the dialog.

Help

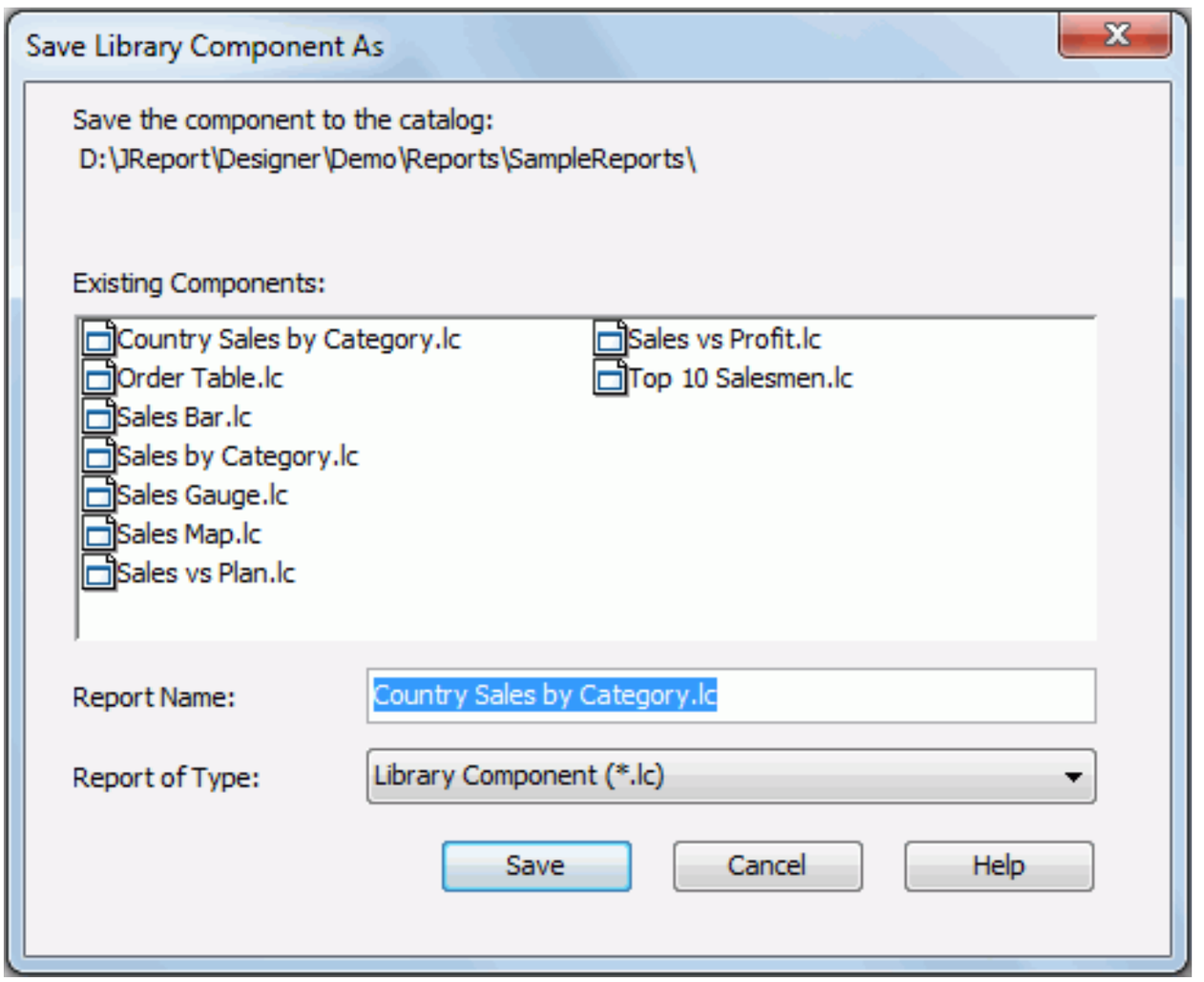
Displays the help document about this feature.

Related topics:

- Creating a CSS style

Save Library Component As dialog

This dialog appears when you click File > Save As on the menu bar when working with library components. It helps you to save the current library component with another name in the current location.



The following are details about options in the dialog:

Report Name

Specifies the name of the library component.

Report of Type

Displays the type of the library component which is .lc.

Save

Saves the library component with the name you specified.

Cancel

Closes the dialog, leaving any changes unsaved.

Help

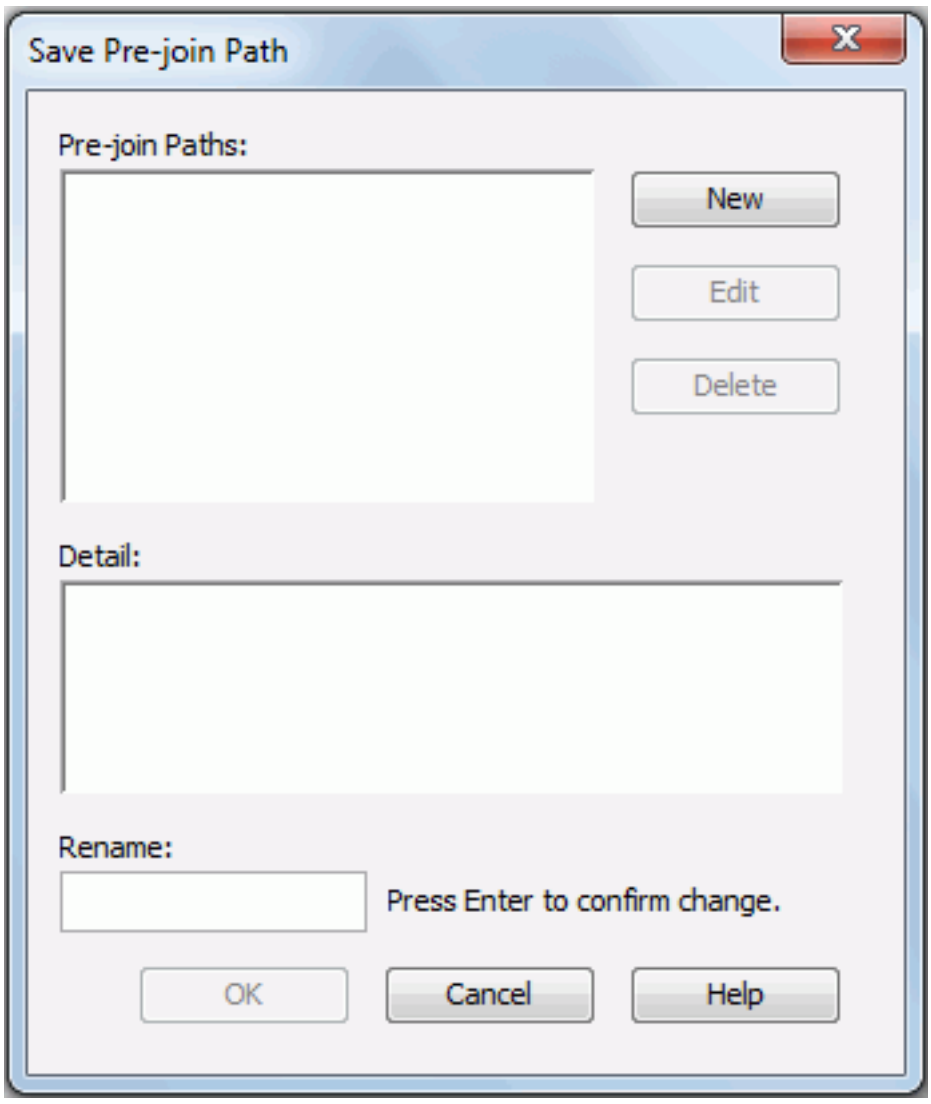
Displays the help document about this feature.

Related topics:

- Saving a library component

Save Pre-join Path dialog

This dialog appears when you click the Paths button in the Pre-join Editor window. It helps you to create and save paths for the joins.



The following are details about options in the dialog:

Pre-join Paths

Lists all the pre-join paths.

New

Opens the Select Pre-join dialog to create a new pre-join path.

Edit

Edits the highlighted pre-join path.

Delete

Deletes the highlighted pre-join path.

Detail

Shows the joins information of the highlighted pre-join path.

Rename

Specifies the new name for the highlighted pre-join.

OK

Applies the changes and closes the dialog.

Cancel

Does not retain any changes and closes the dialog.

Help

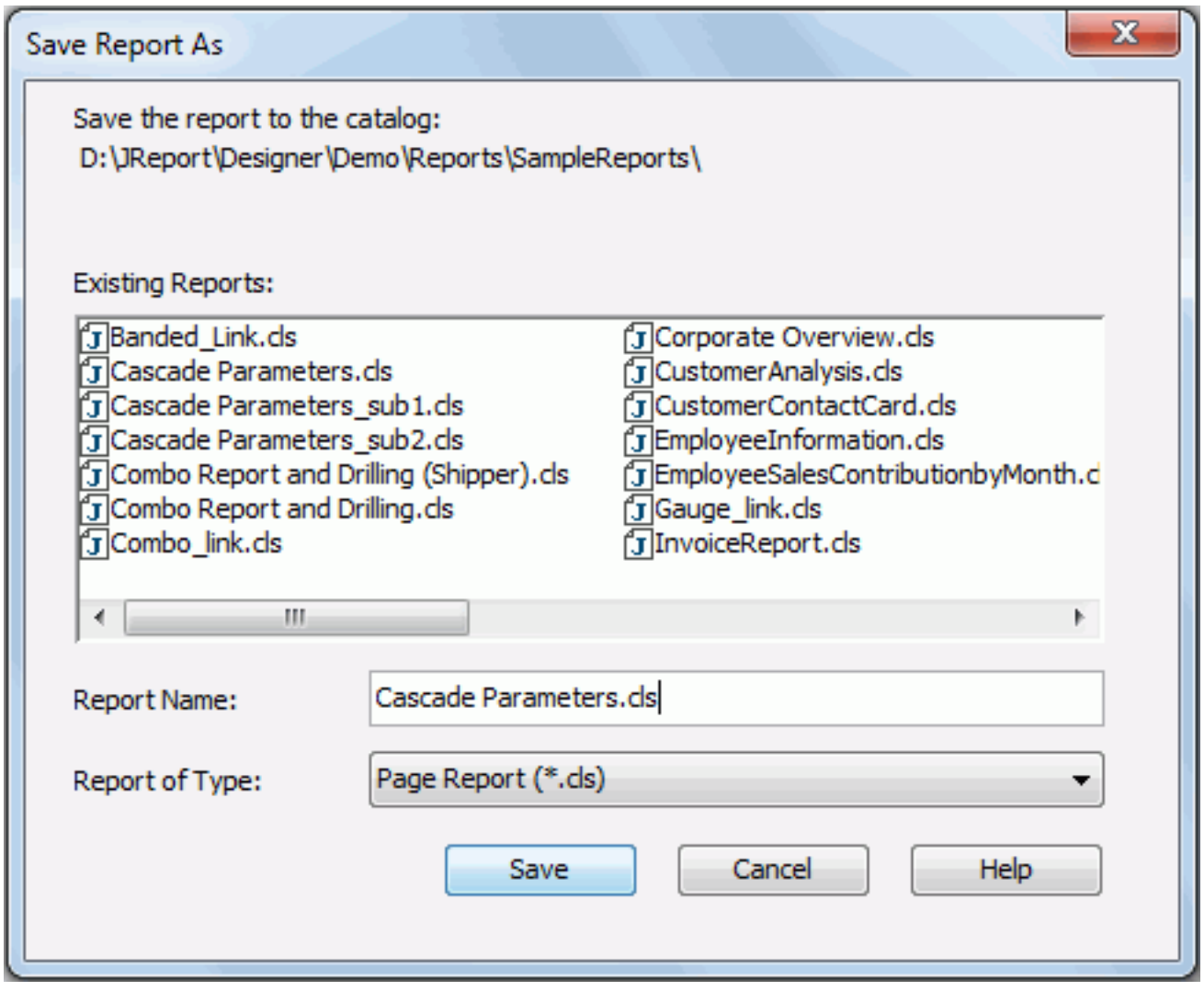
Displays the help document about this feature.

Related topics:

- Defining the join relationships between tables

Save Report As dialog for page report

This dialog appears when you click File > Save As on the menu bar. It helps you to save the current page report with another name in the current location.



The following are details about options in the dialog:

Report Name

Specifies the name of the report.

Report of Type

Specifies the report type:

- **Page Report (*.cls)**
  Saves the report as a binary formatting file.
- **Report Template (*.rpt)**
  Saves the report as a text formatting file. A *.rpt file can be opened and edited as a text file.
- **Self Contained Page Report (*.clx)**
  Saves the report as a binary formatting file which self-contains its own catalog.
- **Page Report XML Format (*.cls.xml)**
  Saves the report as an XML (Extensible Markup Language) formatting file.

Notes:

- When you save a report to Page Report XML Format (*.cls.xml), a special key is needed. If you want, contact Jinfonet support (support@jinfonet.com).
- Report that contains customized UDOs cannot be saved to XML format. If this option is required, contact Jinfonet support (support@jinfonet.com).

Save

Saves the report with the name you specified.

Cancel

Closes the dialog, leaving any changes unsaved.

Help

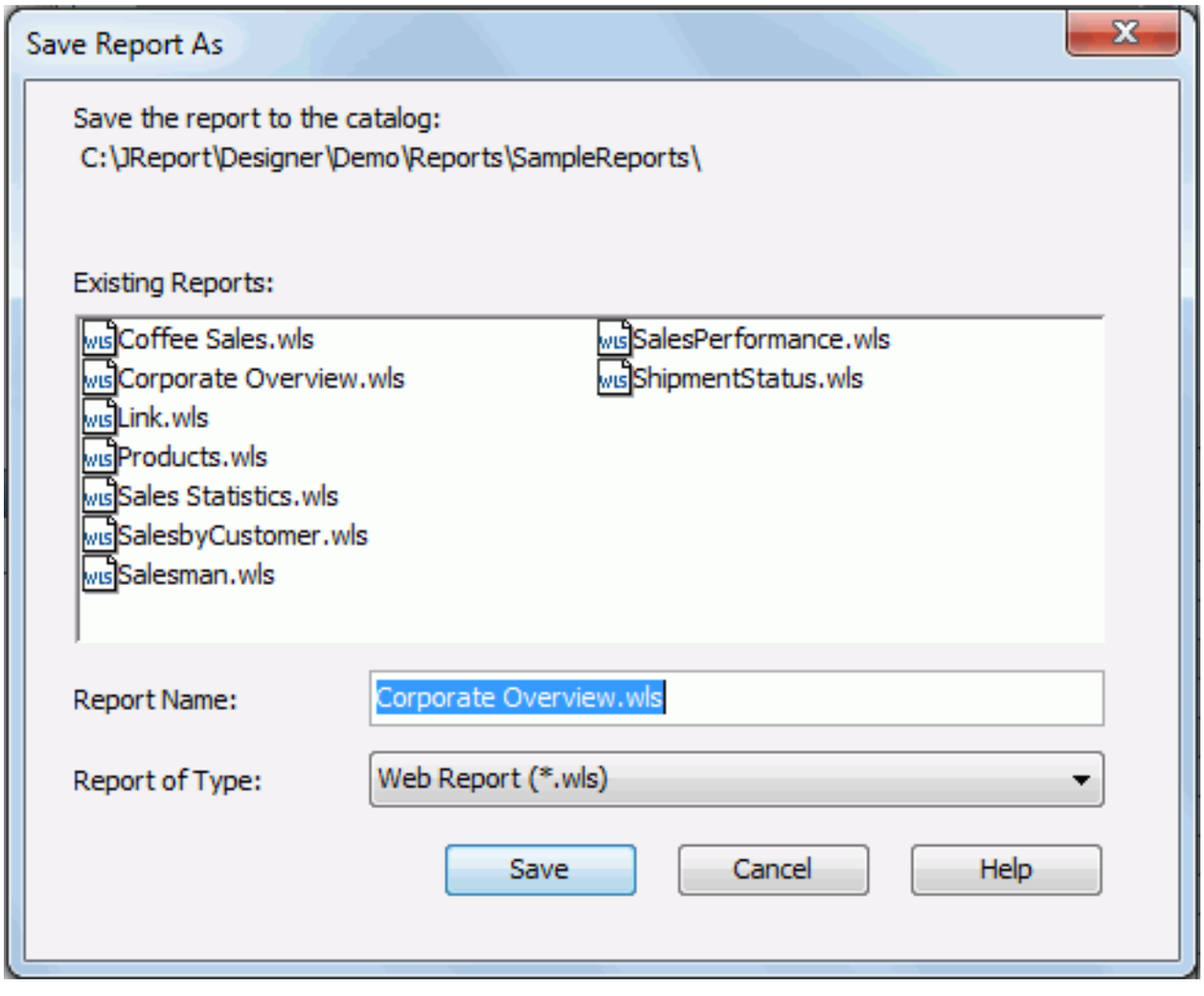
Displays the help document about this feature.

Related topics:

- Saving a page report

Save Report As dialog for web report

This dialog appears when you click File > Save As on the menu bar. It helps you to save the current web report with another name in the current location.



The following are details about options in the dialog:

Report Name

Specifies the name of the report.

Report of Type

Displays the report type .wls by default.

Save

Saves the report with the name you specified.

Cancel

Closes the dialog, leaving any changes unsaved.

Help

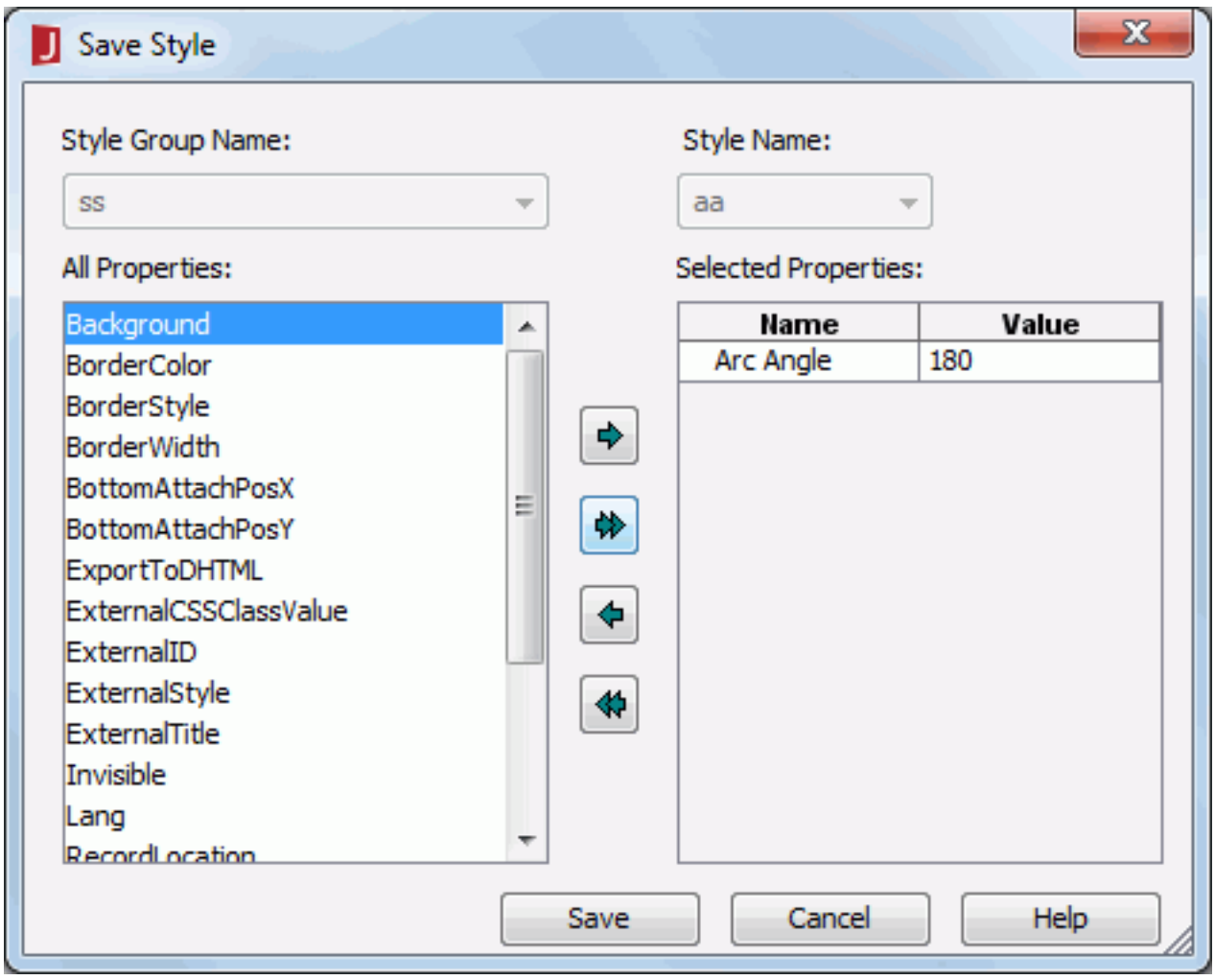
Displays the help document about this feature.

Related topics:

- Saving a web report

Save Style dialog

This dialog is for you to save the style you create in the Styles tab of the Catalog Browser into a catalog.



The following are details about options in the dialog:

Style Group Name

Specifies the name of the style group.

Style Name

Specifies the name of the saved style.

All Properties

Lists all the properties available. Select the properties you want to save in the style from the box.

Selected Properties

Lists the selected properties.

Adds a specified property to the style.

Adds all the properties listed in the All Properties box to the style.

Removes the specified property from the Selected Properties box, so that it will not be applied to the style.

Removes all the properties from the Selected Properties box, so that they will not be applied to the style.

Save

Saves the style and closes the dialog.

Cancel

Does not retain any changes and closes the dialog.

Help

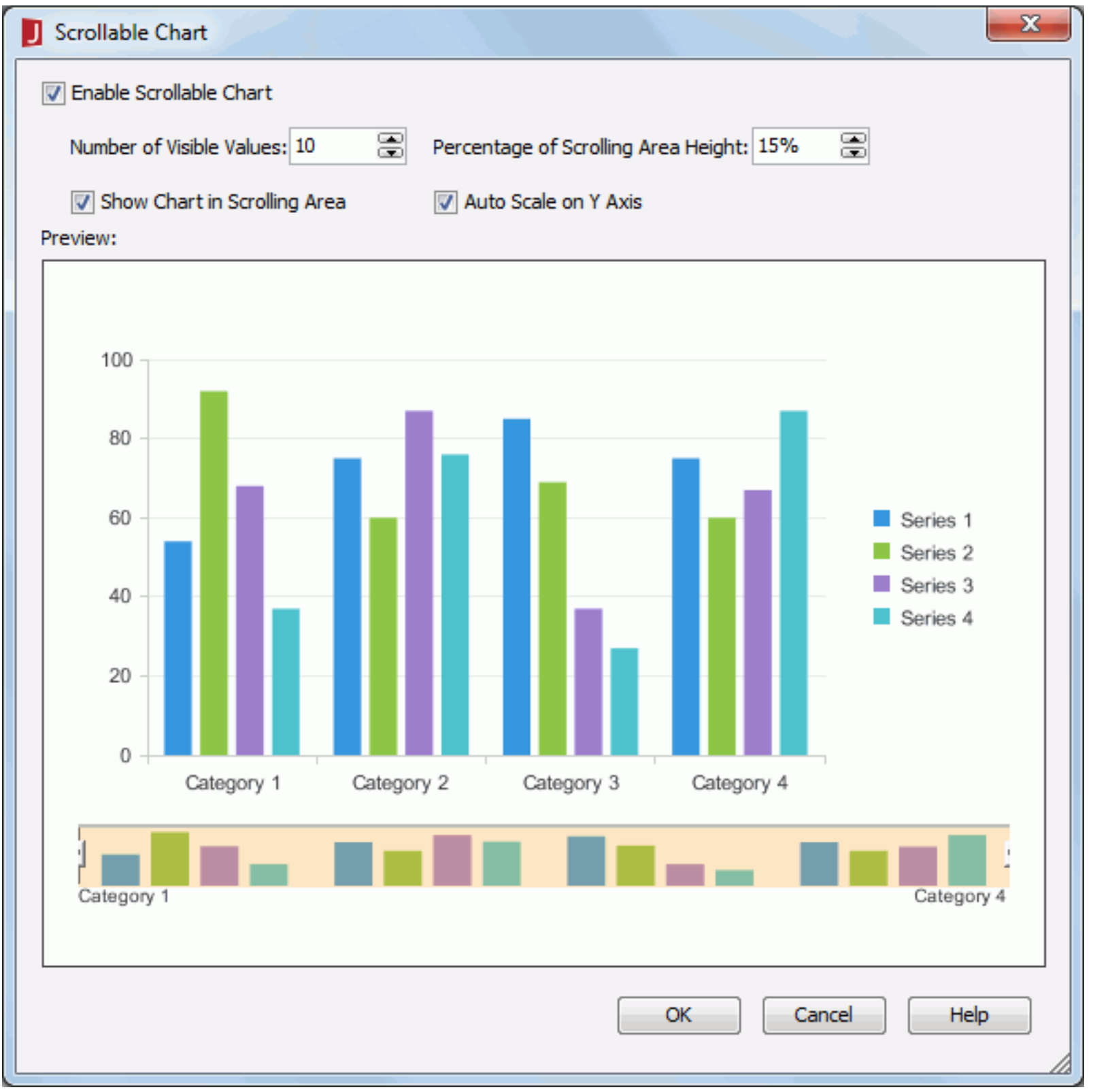
Displays the help document about this feature.

Related topics:

- Creating an XSD style

Scrollable Chart dialog

This dialog appears when you right-click anywhere of a 2-D chart of bar, bench, line, or area type except the chart object and select Scrollable Chart from the shortcut menu. It helps you to specify whether to make the chart in a web report or library component scrollable.



The following are details about options in the dialog:

Enable Scrollable Chart

Specifies whether to make the chart scrollable. If checked, a scrollbar will be added in the chart, with which you can control the number of visible values on the axis.

- **Number of Visible Values**
  Specifies how many data items will be selected on the scrollbar and displayed on the axis by default.
- **Percentage of Scrolling Area Height**
  Specifies the percentage the scrollbar occupies the whole size of the chart.
- **Show Chart in Scrolling Area**
  Specifies whether to show the thumbnail chart on the scrollbar.
- **Auto Scale on Y Axis**
  Specifies whether to recalculate the scale of Value (Y) Axis when the the value of the option Number of Visible Values is changed.

Preview

Displays the effect of your settings for the chart.

OK

Saves the settings for the chart and closes the dialog.

Cancel

Does not retain any changes and closes the dialog.

Help

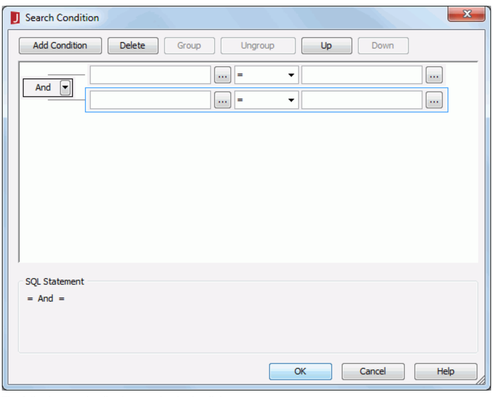
Displays the help document about this feature.

Related topics:

- Creating a scrollable chart

Search Condition dialog

This dialog appears when you click Query > Filter in the Query Editor window. It helps you to specify more criteria for the fields to be retrieved from the database.



The following are details about option in the dialog:

Add Condition

Adds a new condition line.

Delete

Deletes the selected condition line.

Group

Makes the selected conditions in one group. Conditions can also be added to an existing group by selecting the conditions and the group while holding the Ctrl button, and then clicking the Group button.

Ungroup

Makes the selected condition ungrouped.

Up

Moves the selected condition up to a higher level.

Down

Moves the selected condition or group down to a lower level.

Logic

Lists the logic operator.

- **And**
  Logic operator And which is applied to this and the next line.
- **Or**
  Logic operator Or which is applied to this and the next line.
- **And Not**
  Logic operator And Not which is applied to this and the next line.
- **Or Not**
  Logic operator Or Not which is applied to this and the next line.

Field

Specifies the field to be filtered. You can either type in the field manually or click ... to specify the field in the Expressions dialog. The following are the options in this dialog:

- **Real Name tab**
  Lists the tables added in the query with all their columns, also the formulas and parameters in the catalog. Double-click the required field to create the filter on it.

  The following symbols at the bottom of the Expressions dialog allow you to modify the expression according to your requirements.

  - **+**
    Adds numbers or fields together in the Expression menu.
  - **-**
    Subtracts numbers or fields together in the Expression menu.
  - *****
    Multiplies numbers or fields together in the Expression menu.
  - **/**
    Divides numbers or fields together in the Expression menu.
  - **=**
    Equates fields together
  - **"**
    Places quotations on long character strings or name that have blanks in them. For example, you should place quotes on values such as "New York" or "Washington DC"
  - **||**
    Allows you to place fields together in the same Expression menu. For example, "New York" || "Washington DC")
  - **( )**
    Places your fields in parentheses
- **Subquery tab**
  Allows you to create a subquery by editing a query or creating a new query. For details about using sub-queries, see Using sub-queries in filters.
  - **Edit Subquery**
    Opens the Query Editor window to edit the query.
  - **New Subquery**
    Creates a new subquery here.
- **Field Value tab** (disabled when the Expressions dialog is opened while setting the filter field)
  Lists all column names of the related table, in which the field to be filtered is. Check the column name of the field to be filtered, click OK, then the values of the field will be displayed. Choose the required value and click Set to use it to filter the field.

Operator

Lists all the operators that can be used in the filter condition:

- **=**
  Equal to
- **< >**
  Not equal to
- **>**
  Greater than
- **> =**
  Greater than or equal to
- **<**
  Less than
- **< =**
  Less than or equal to
- **[not] in**
  Causes an enumerated list of values to appear in the WHERE clause predicate, used for evaluating for a true condition. The [not] in operator supports multi-value parameters as the value.
- **[not] between**
  Allows the system to evaluate whether or not data values are located between a range of values indicated in the predicate.
- **is [not] null**
  It is used in WHERE clause predicates to match null values occurring in a specified data column.
- **[not] like**
  Like string pattern matching operator is used to compare the first expression string value to the pattern string (the second expression). If you want to use wildcard character in the pattern string, only "_" and "%" are supported.
- **sounds like**
  Sound like operator uses a soundex algorithm to pattern match strings that sound alike.
- **[not] exists**
  It is used to test if a subselect retrieves any records.

Value

Specifies the value of how to filter the field. You can either click ... next to the text box to specify the value using the Expressions dialog or type the value in the text box manually.

When you input the value manually:

- Multiple values should be separated with ",". If "," or "\" is contained in the values, write it as "\," or "\ \".
- **For String type values**
  Quote the values with single quotes.

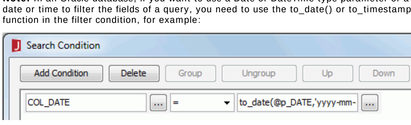
  Example1: Customers_Country='USA'.

  Example2: Customers_Country in 'Australia','Germany','Mexico'

  The quote marks can be typed by yourself or be added by JReport Designer automatically. To have the quote marks automatically added by JReport Designer, follow the steps below:

  1. Click the **Options** button on the Catalog Browser toolbar.
  2. In the Options dialog, switch to the Query Editor category and check **Automatically add quotation marks on input values**.
  3. Click **OK** to save the settings.
- **For Date type values**
  Make sure the format of the value you enter is consistent with that of your database. Filters are done in the RDBMS side and some RDBMS have special requirements for the date format, and you need to make sure the date formats specified in the Data Format tab of the Get JDBC Connection Information dialog are supported by your database.

  Note: In an Oracle database, if you want to use a Date or DateTime type parameter or a specific date or time to filter the fields of a query, you need to use the to_date() or to_timestamp() function in the filter condition, for example:



SQL Statement

Displays the SQL statement of the filter condition.

OK

Applies all changes and closes the dialog.

Cancel

Closes the dialog, leaving any changes unsaved.

Help

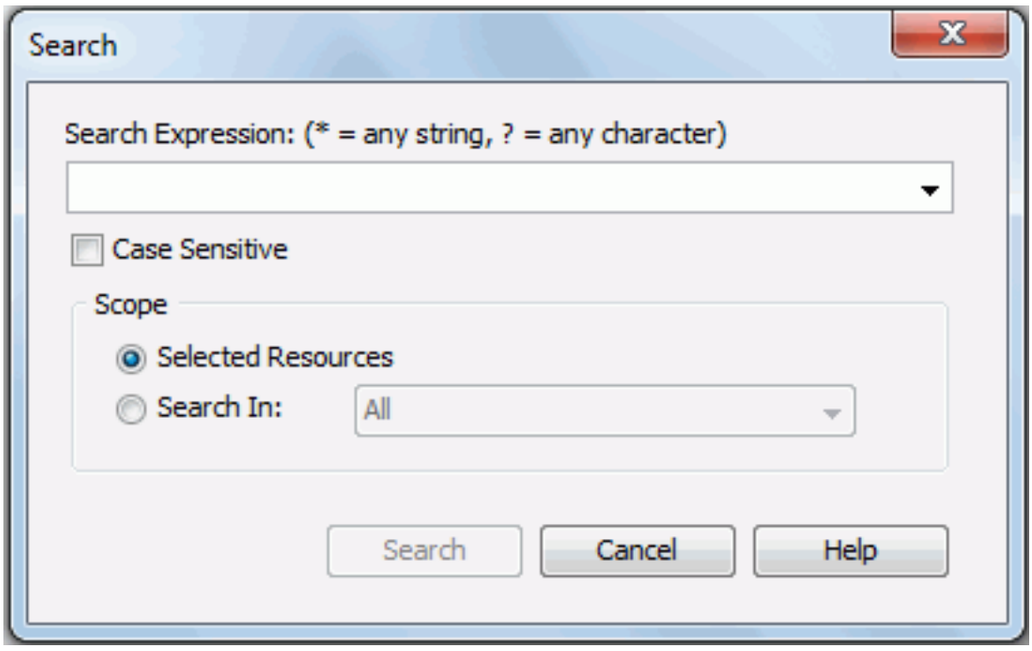
Displays the help document about this feature.

Related topics:

- Filtering the fields of a query

Search dialog

This dialog appears when you click the Search button on the toolbar of the Catalog Browser or the Resource View panel. It enables you to search leaf node resources that you need in the Catalog Browser/Resource View panel.



The following are details about options in the dialog:

Search Expression

Specifies the text to search for.

Case Sensitive

Specifies whether to match whole word only or match case.

Scope

Specifies the search range.

- **Selected Resources**
  If selected, you will search what you need in the selected resources.
- **Search In**
  If selected, specify in which resources of the Catalog Browser/Resource View panel to search what you need. Select the desired resource from the drop-down list.

Search

Begins the search process and displays the results in the Search Result panel.

Cancel

Cancels the search process and closes the dialog.

Help

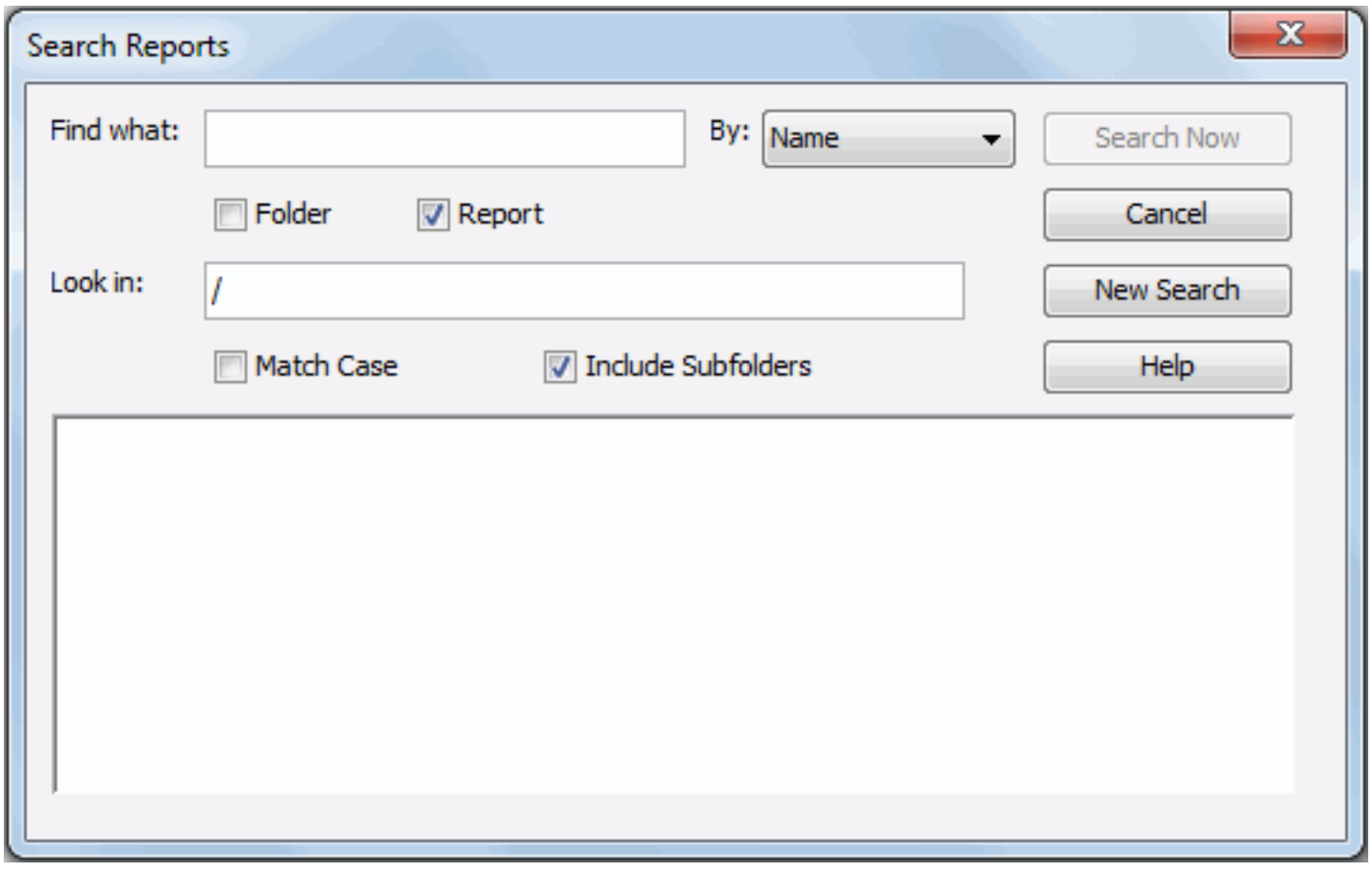
Displays the help document about this feature.

Related topics:

- Searching for resources
- Searching for resources in the Resource View panel

Search Reports dialog

This dialog appears when you right-click a folder in the Reports tab of the Catalog Browser and then select Search from the shortcut menu. It helps you to search for a report or a folder.



The following are details about options in the dialog:

Find what

Specifies the search content, which can be report or folder name, or report keywords. Input the text you want to search for in the text box directly.

If you just input part of the report or folder name, or report keywords, use * to replace the other part of the name or keywords, take the report name EmployeeInformation for example, if you just input infor and want to search for this report, input *Infor* in the text box.

By

Specifies the searching criteria, by name, by keywords or by name or keywords.

Folder

Specifies to search for a folder.

Report

Specifies to search for a report.

Look in

Specifies the search path.

Match Case

Limits the search to text that exactly matches the case of the search text.

Include Subfolders

If checked, the search results includes subfolders.

Search Now

Begins the search process.

Cancel

Cancels the search process and closes the dialog.

New Search

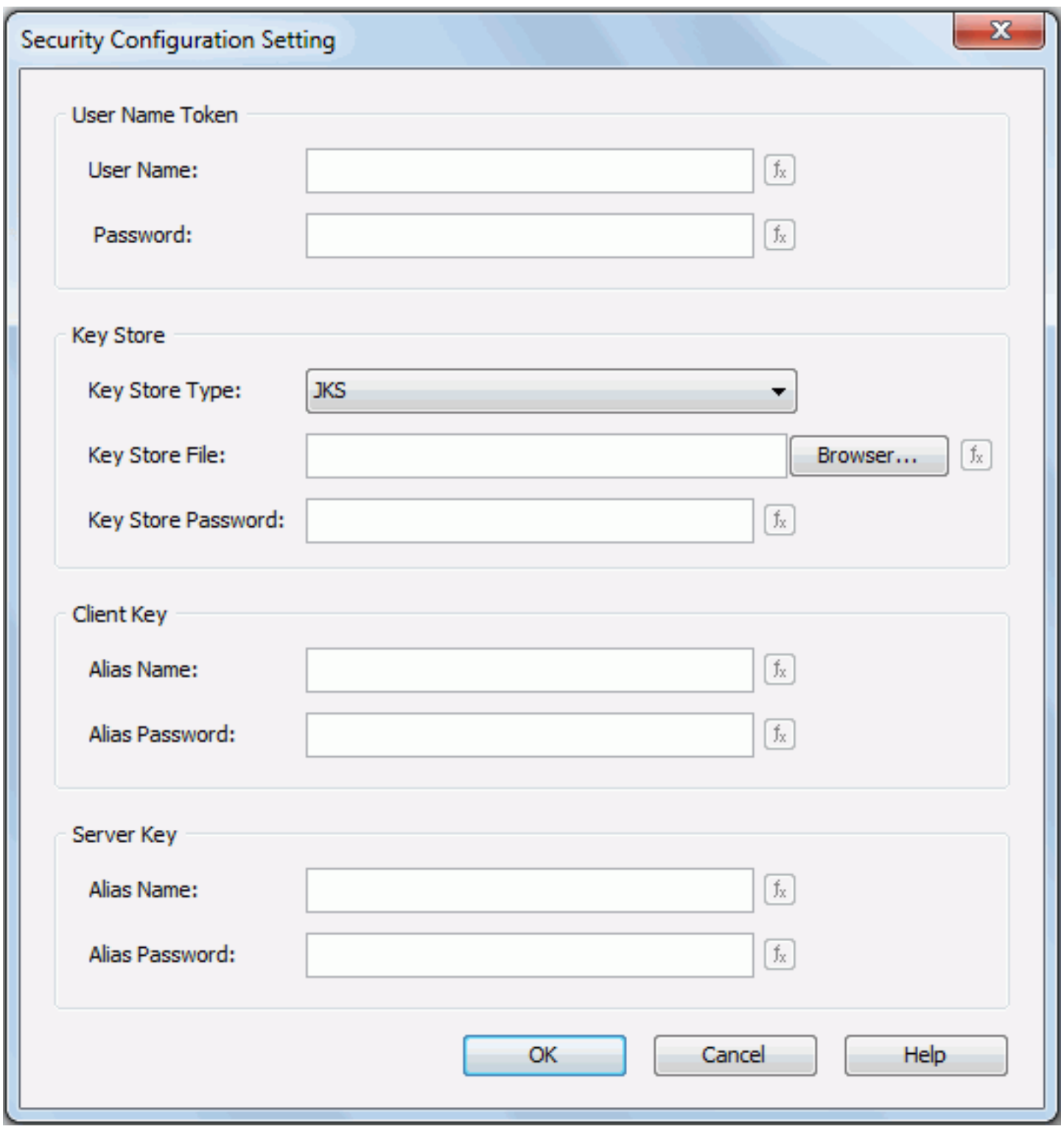
Refreshes this dialog in order to set a new search.

Help

Displays the help documentation.

Security Configuration Setting dialog

The dialog appears when you click the Security Configuration button in the Web Service Data Source dialog. It helps you to configure security for the web service data source.



The followings are details about options of the dialog:

User Name Token

Specifies the properties for the user name token.

- **User Name**
  Specifies the user name for the user name token to be used in the security policy.
- **Password**
  Specifies the password for the user name token to be used in the security policy.

Key Store

Specifies the properties of the key store.

- **Key Store Type**
  Specifies the type for the key store. It could be JKS or PKCS12.
- **Key Store File**
  Specifies the URI to get the key store file. You can click the Browse button to find the file or input the file path directly in the text field.
- **Key Store Password**
  Specifies the password to open the key store file.

Client Key

Specifies the properties for the client key.

- **Alias Name**
  Specifies the alias name which is used as client signature in the key store.
- **Alias Password**
  Specifies the password for the alias name.

Server Key

Specifies the properties for the server key.

- **Alias Name**
  Specifies the alias name which is used to get the server-side certification or public key in the key store.
- **Alias Password**
  Specifies the password for the alias name.

Indicates the value of the option can be controlled by a formula. Click this button and select a formula from the drop-down list. If the given formulas cannot meet your requirements, click <CREATE FORMULA> to create one.

OK

Accepts the configuration settings and closes the dialog.

Cancel

Cancels the configuration settings and exists the dialog.

Help

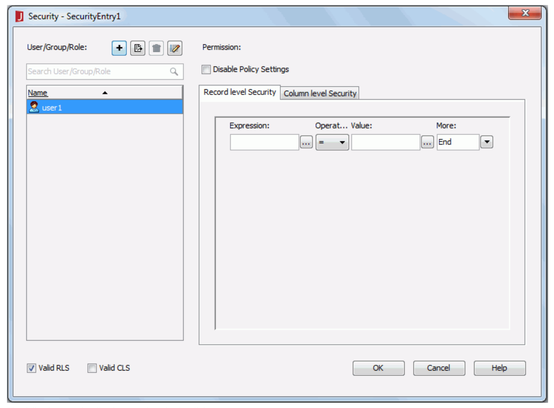
Displays the help document about this feature.

Related topics:

- Setting up a web service connection

Security dialog

This dialog appears when you right-click Security Entry in Catalog Browser > Data Source > Security, select Add Security Entry from the shortcut menu, input a name for the new security policy and then click OK. This dialog allows you to specify the security settings for a connection-scope security policy.



The following are details about options in the dialog:

User/Group/Role

Lists all the users, roles, groups for the security policy.

- Adds a user/group/role. After you click the button, a drop-down menu will be shown:
  - **Add User**
    Creates a new user in the Add User dialog.
  - **Add Group**
    Creates a new group in the Add Group dialog.
  - **Add Role**
    Creates a new role in the Add Role dialog.
  - **Import from JReport Server**
    Opens the Connect to JReport Server dialog to import users, roles, and groups from JReport Server.
  - **Import from File**
    Imports users, roles, and groups from an xml file.
- Exports the users, roles, and groups to an xml file.
- Deletes the selected user/group/role.
- Edits the selected user/role/group in the displayed dialog.
- **Quick search toolbar**
  Type in text in the toolbar to search for the user/role/group in the list below. The names containing the matched text will be shown.

Valid RLS

Specifies whether to use RLS policy for this security entry. If unchecked, the RLS policy will not be applied and all the records will be available to the users.

Valid CLS

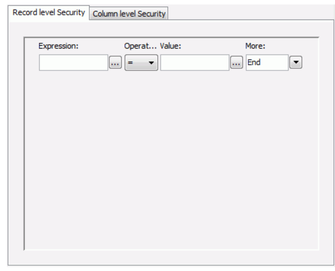
Specifies whether to use CLS policy for this security entry. If unchecked, the CLS policy will not be applied and all the records will be available to the users.

Disable Policy Settings

- When it is unchecked, you are able to specify a security policy for the principal. However, if the record level security and column level security are left blank, the principal will not be able to view any data records, unless it inherits some permissions from another group or role.
- When it is checked, you are not allowed to specify record level security or column level security. The principal will have full access to all the data records in the reports which adopt this security policy entry.

Record Level Security tab

Specifies the condition statements.



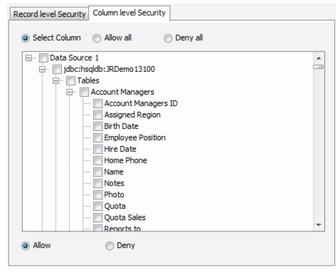
- **Expression**
  Specifies the field on which the condition is based. You can type in the field directly or click the button ... next to the text box to select the desired field in the Expressions dialog.
- **Operator**
  Specifies the operator to compose the condition expression.
  - **=**
    Equal to
  - **>**
    Greater than
  - **<**
    Less than
  - **>=**
    Greater than or equal to
  - **<=**
    Less than or equal to
  - **<>**
    Not equal to
  - **in**
    Causes an enumerated list of values to appear in the WHERE clause predicate, used for evaluating for a true condition. For operator "in", it is allowed to input multiple values in the Value input box and separate the values with comma (,).
- **Value**

  Specifies the value of how to build the condition. You can type in the value manually or click the button ... next to the value text box to specify the value in the Expressions dialog.
- **More**
  Lists some additional commands.
  - **AND**
    Specifies the relationship between two expression statements as logical AND. If this line is the last line in the expression list, when you select AND or OR, and a new line will be appended to the end of the list as well.
  - **OR**
    Specifies the relationship between two expression statements as logical OR. If this line is the last line in the expression list, when you select AND or OR, and a new line will be appended to the end of the list as well.
  - **Insert Row**
    Inserts a new line after the current line.
  - **Delete Row**
    Deletes the current line.
  - **New Group**
    Adds a new expression group to the list. The relationships between two groups can be:
    - **AND**- Logical AND relationship between two groups. Records satisfying both condition groups will be retrieved.
    - **OR**- Logical OR relationship between two groups. Records satisfying either one of the condition groups will be retrieved.
    - **AND NOT** - Records which satisfy the first condition group but not the second will be retrieved.
    - **OR NOT** - Records which satisfy the first condition group, or which do not satisfy the second condition group will be retrieved.

Note: If the policy settings are left blank, the user in this role will not be able to view any records, unless he/she belongs to more than one role, and the other roles have the necessary permissions.

Column Level Security tab

Lists all the available resources in the catalog. Select the required resources to allow or deny them to be revealed to the principal.



- **Select Column**
  Specifies the columns that are to be shown to (or hidden from) the principal.
- **Allow all**
  Allows all the columns in the field list to be revealed to the principal.
- **Deny all**
  Hides all the columns in the field list from the principal.
- **Allow**
  Allows the selected columns in the field list to be revealed to the principal.
- **Deny**
  Hides the selected columns in the field list from the principal.

OK

Accepts changes and closes the dialog.

Cancel

Does not retain the changes and closes the dialog.

Help

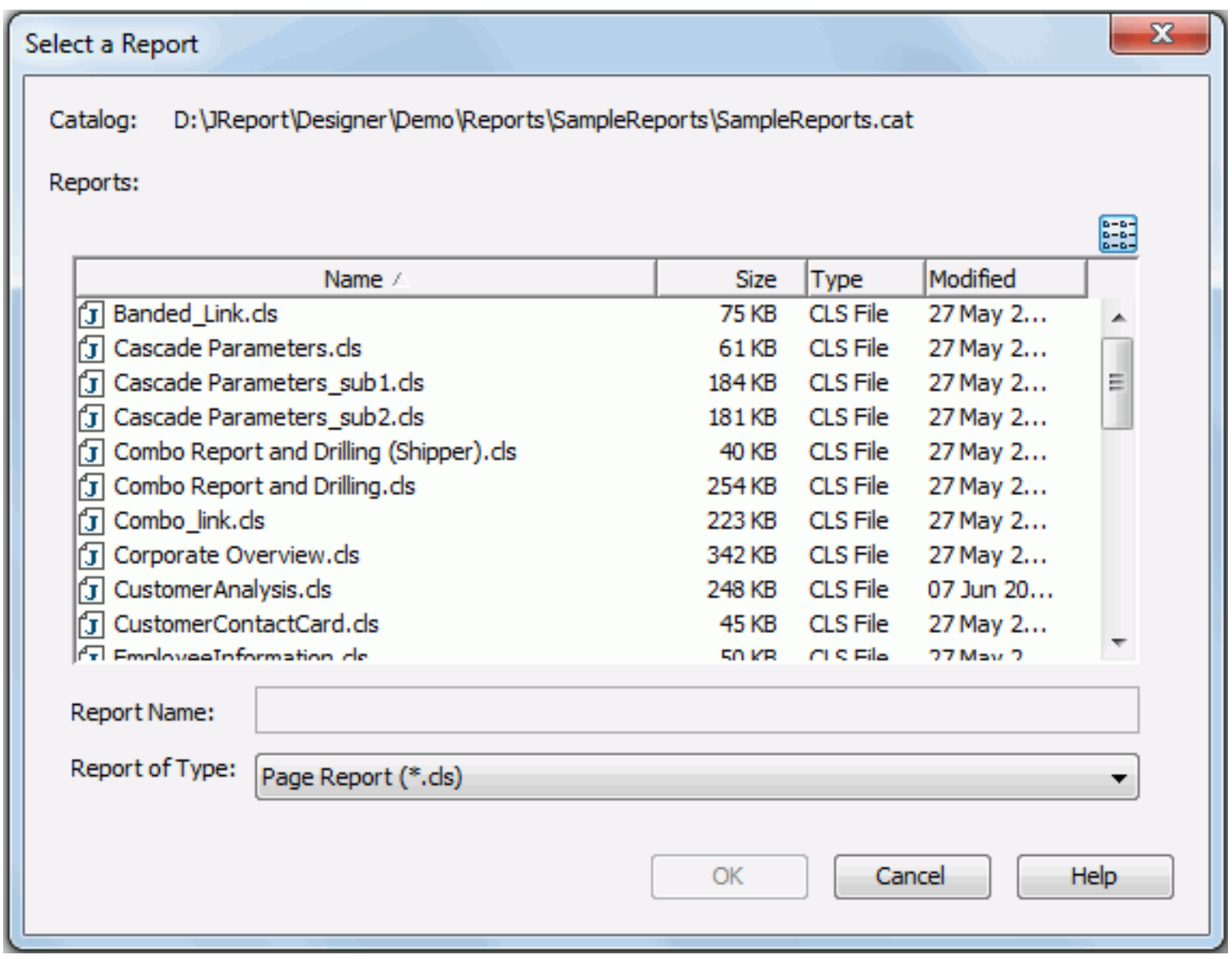
Displays the help document about this feature.

Related topics:

- Setting up a connection-scope security policy

Select a Report dialog for page report

The dialog appears when you click the Browse button next to the Report text field in the Insert Link dialog for page report or Edit Link dialog for page report. It helps you to specify the report with which to link the primary or master report.



The following are details about options in the dialog:

Catalog

Shows the directory of the catalog that is being used now.

Reports

Specifies the report which contains the linked/detail report tab.

- **List**
  Clicks to go to the List mode where only the names of the files are displayed.
- **Details**
  Clicks to go to the Details mode where the files with detailed information such as name, size, type and last modified time are displayed.
- **Name**
  Shows the names of all the reports included in this catalog, based on the type you selected from the Report of Type drop-down list.
- **Size**
  Shows the size of the reports.
- **Type**
  Shows the type of the reports.
- **Modified**
  Shows the time when the reports were modified at the last time.

Report Name

Shows the names of the selected reports.

Report of Type

Specifies the type of the reports.

OK

Accepts the selecting report result and closes the dialog.

Cancel

Cancels the selecting report process and exits the dialog.

Help

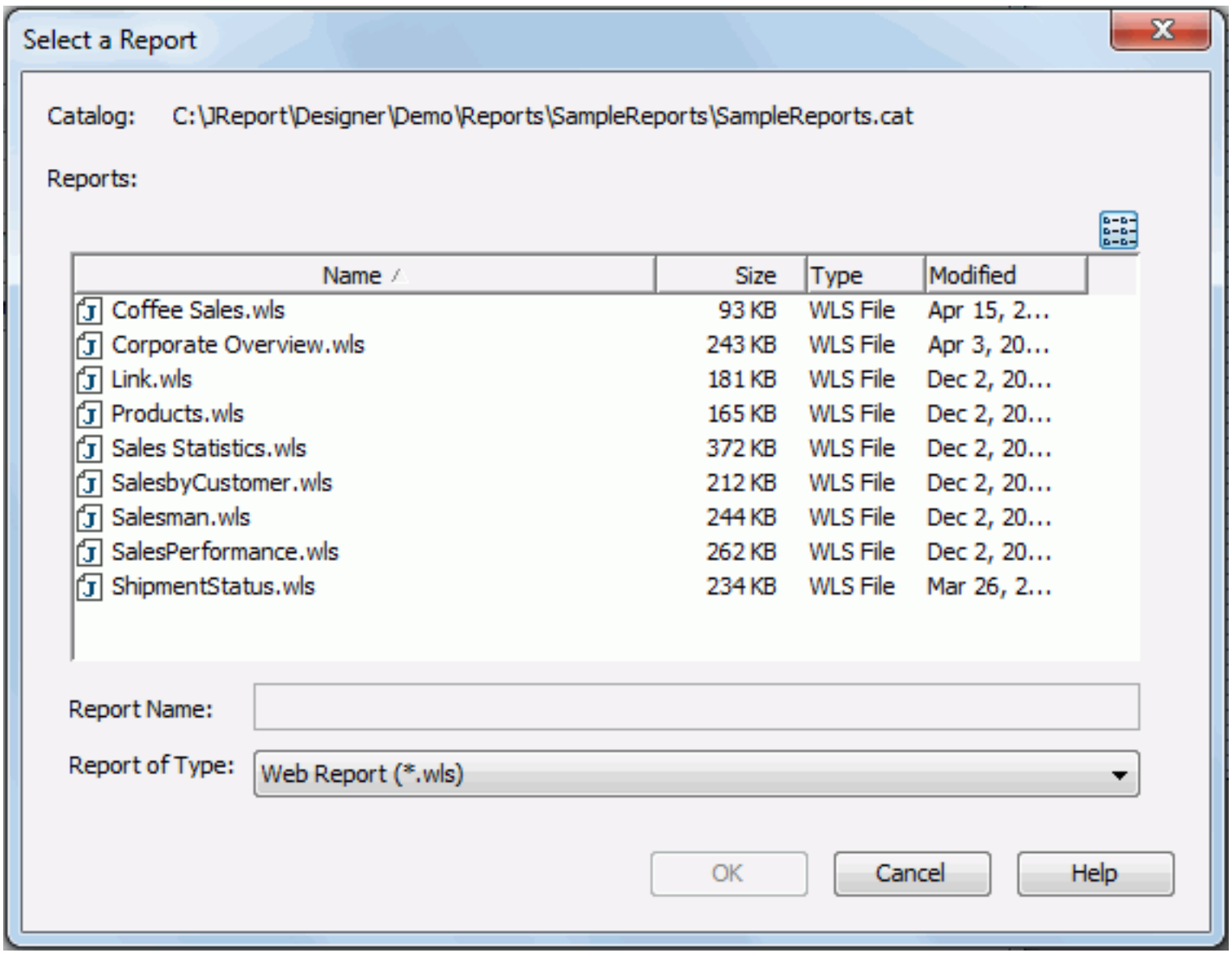
Displays the help document about this feature.

Related topics:

- Linking to another report
- Linking to a detail report

Select a Report dialog for web report/library component

The dialog appears when you click the Browse button next to the Report text field in the Insert Link dialog for web report/library component or Edit Link dialog for web report/library component. It helps you to specify the report with which to link the primary report.



The following are details about options in the dialog:

Catalog

Shows the directory of the catalog that is being used now.

Reports

Specifies the linked target report.

- **List**
  Clicks to go to the List mode where only the names of the files are displayed.
- **Details**
  Clicks to go to the Details mode where the files with detailed information such as name, size, type and last modified time are displayed.
- **Name**
  Shows the names of all the reports included in this catalog.
- **Size**
  Shows the size of the reports.
- **Type**
  Shows the type of the reports.
- **Modified**
  Shows the time when the reports were modified at the last time.

Report Name

Shows the names of the selected reports.

Report of Type

Shows the type of the reports.

OK

Accepts the selecting report result and closes the dialog.

Cancel

Cancels the selecting report process and exits the dialog.

Help

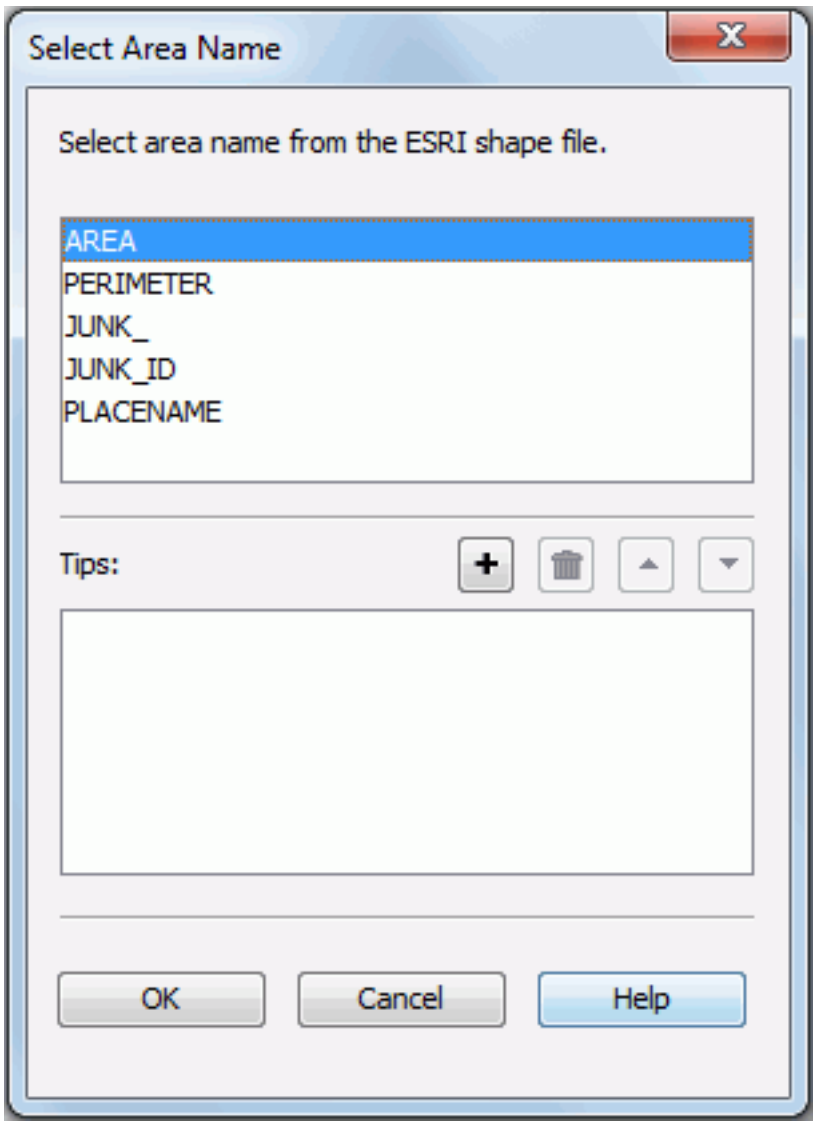
Displays the help document about this feature.

Related topics:

- Linking to another report

Select Area Name dialog

This dialog appears when you specify to import areas from a .shp file. It helps you to define the map area name.



Select area name from the ESRI shape file

Lists all the available fields in the selected ESRI shape file. Select one field in the box to define the area name of the map object.

Tips

Lists the fields from which to get the area tooltips, which will be displayed when you hover the mouse over the areas when the report runs in HTML format or in Page Report Studio. The order of the fields in this box determines the display sequence of the tooltips and the tooltips can be displayed only when Alternate Content Type on the map object is set to customized.

- To add a field to use its value as tooltip, select it from the box above, then click .
- To remove a field from the tooltip, select it in the Tips box, then click .
- To adjust the order of the fields in the Tips box, select a field, then click to move the field one step up, or to move it one step down.

OK

Applies the changes and closes the dialog.

Cancel

Does not retain any changes and closes the dialog.

Help

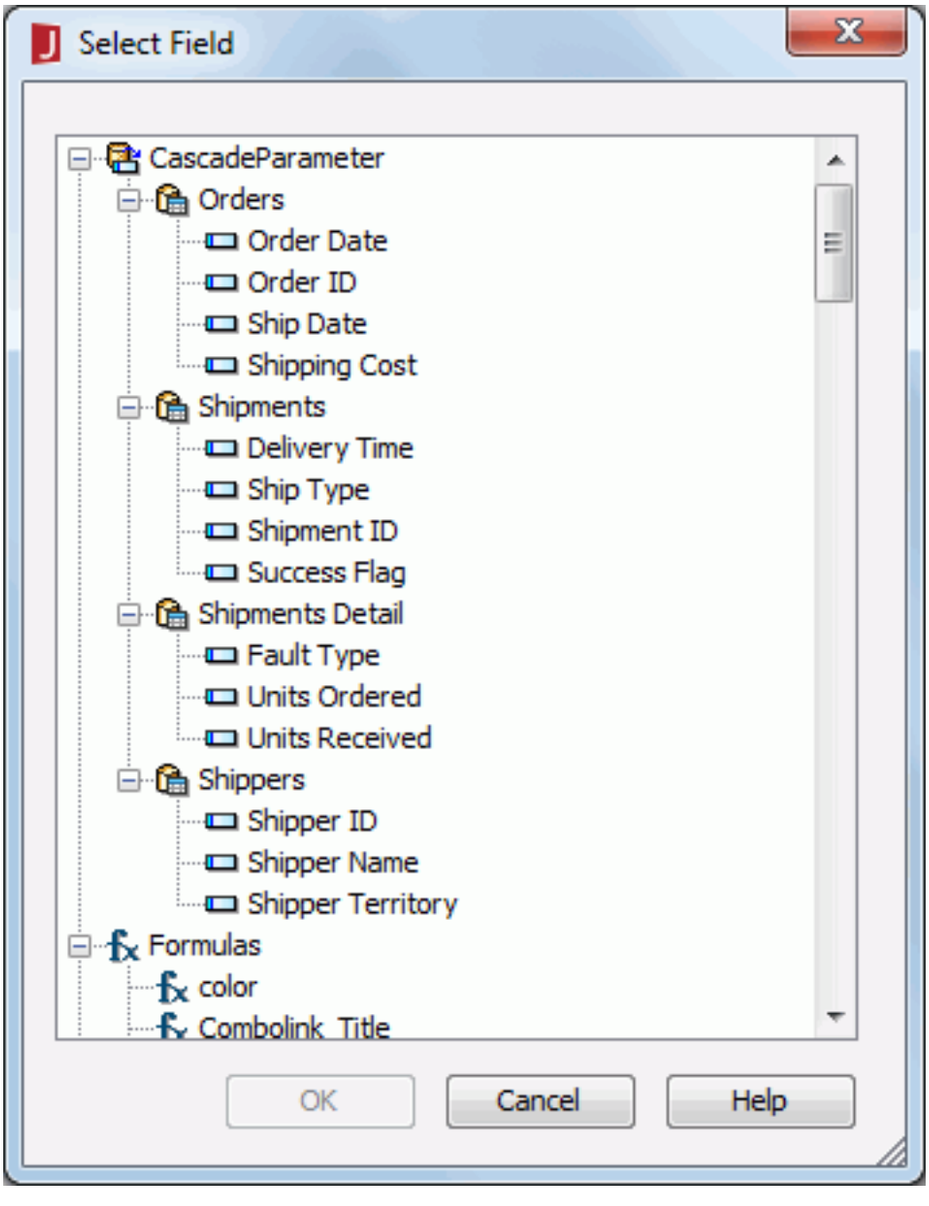
Displays the help document about this feature.

Related topics:

- Inserting a shape map

Select Field dialog

The dialog appears when you click the Add Dynamic Field button in the Insert Link dialog for page report or web report/library component, or the Edit Link dialog for page report or web report/library component when the link type is URL or E-mail. It helps you to insert a field into the URL or e-mail address that an object is linked to.



Resources box

Lists all the available data resources in the currently used dataset. Select a field from the resource tree to insert it into the URL or e-mail address to compose a dynamic hyperlink. You should make sure the field you select contains the information to compose a correct URL or e-mail address.

OK

Applies the selected field and closes the dialog.

Cancel

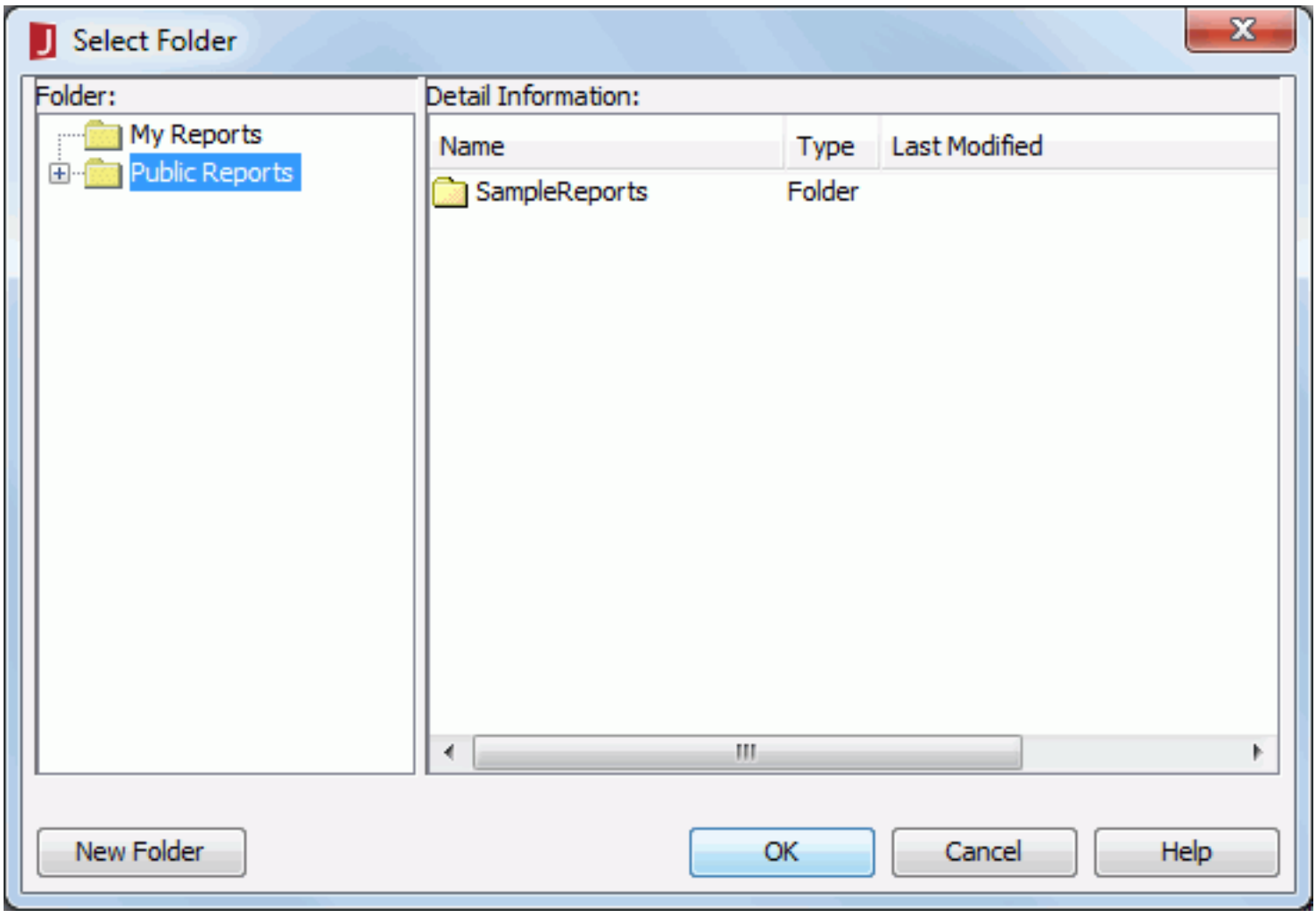
Cancels the selection and exits the dialog.

Help

Displays the help document about this feature.

Select Folder dialog

This dialog appears when you click the Browse button next to the Publish Resource To text field in the Publish to JReport Server dialog or the Browse button next to the Download Resource From text field in the Download from JReport Server dialog. It helps you to specify the folder on the server where you want the resources to be published or from which resources will be downloaded.



The following are details about options in the dialog:

Folder

Lists folders on JReport Server in a tree structure.

Detail Information

Shows the detailed information of sub folders in the selected folder.

- **Name**
  Displays the names of sub folders.
- **Type**
  Displays the file type.
- **Last Modified**
  Displays the last modified time of the sub folders.
- **Description**
  Other necessary descriptions of the sub folders.

New Folder

Creates a new folder in the selected folder. You can rename the folder by double-clicking the folder in the folder tree.

This button is available only when the dialog is opened by clicking the Browse button next to the Publish Resource To text field in the Publish to JReport Server dialog.

OK

Saves the information and closes the dialog.

Cancel

Closes the dialog, leaving any changes unsaved.

Help

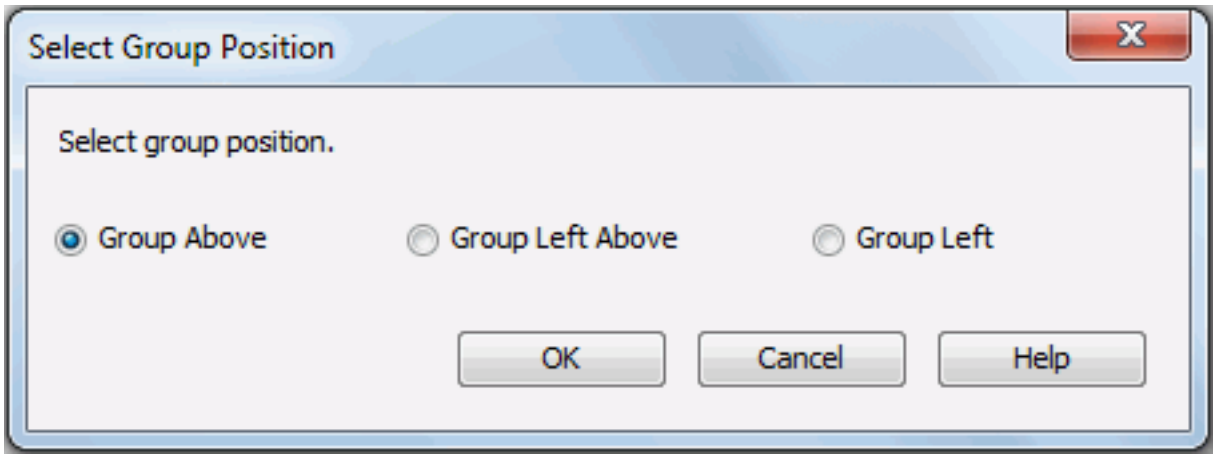
Displays the help document about this feature.

Related topics:

- Publishing resources remotely
- Downloading resources from JReport Server

Select Group Position dialog

The dialog appears when you right-click the detail column of a table and select Convert to Group from the shortcut menu. It helps you to specify the position of the group by field when you convert the detail column to a group column.



The following are details about options in the dialog:

Group Above

Specifies to place the group by field in its own row above the detail columns.

Group Left Above

Specifies to place the group by field in its own row and column above and left of the detail columns.

Group Left

Specifies to place the group by field in its own column left of the detail columns.

OK

Applies the setting to convert the column and closes the dialog.

Cancel

Cancels the conversion and closes the dialog.

Help

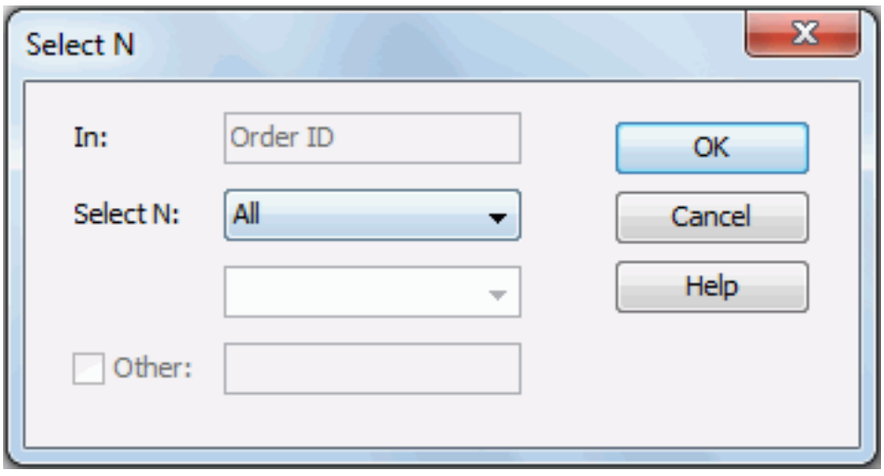
Displays the help document about this feature.

Related topics:

- Managing columns/rows in a table - Converting a column

Select N dialog

This dialog appears when you click the Select N button in the Group screen of the banded/table wizard, or in the Display screen of the map wizard. It helps you to specify a number to filter groups to be displayed in the report, or a parameter which returns an integer to filter group dynamically when previewing the report.



The following are details about options in the dialog:

In

Indicates where the Select N condition will be applied.

Select N

Specifies the Select N condition to define groups that will be displayed while others are hidden.

- **All**
  If selected, all groups in the report will be displayed.
- **Top**
  If selected, specify a number in the field below and the first N groups in the report will be displayed. You can also select a parameter from the drop-down list, which returns an integer, and then specify a value for the parameter when viewing the report to dynamically define the Top N condition.
- **Bottom**
  If selected, specify a number in the field below and the last N groups in the report will be displayed. You can also select a parameter from the drop-down list which returns an integer, and then specify a value for the parameter when viewing the report to dynamically define the Bottom N condition.

Other

If checked, all the other records that don't match the Select N condition will be displayed (which are by default hidden) in another group. Specify a name for this group as required.

This option is available only when the Select N condition is applied to a certain group.

OK

Accepts the changes and closes the dialog.

Cancel

Discards the changes and closes the dialog.

Help

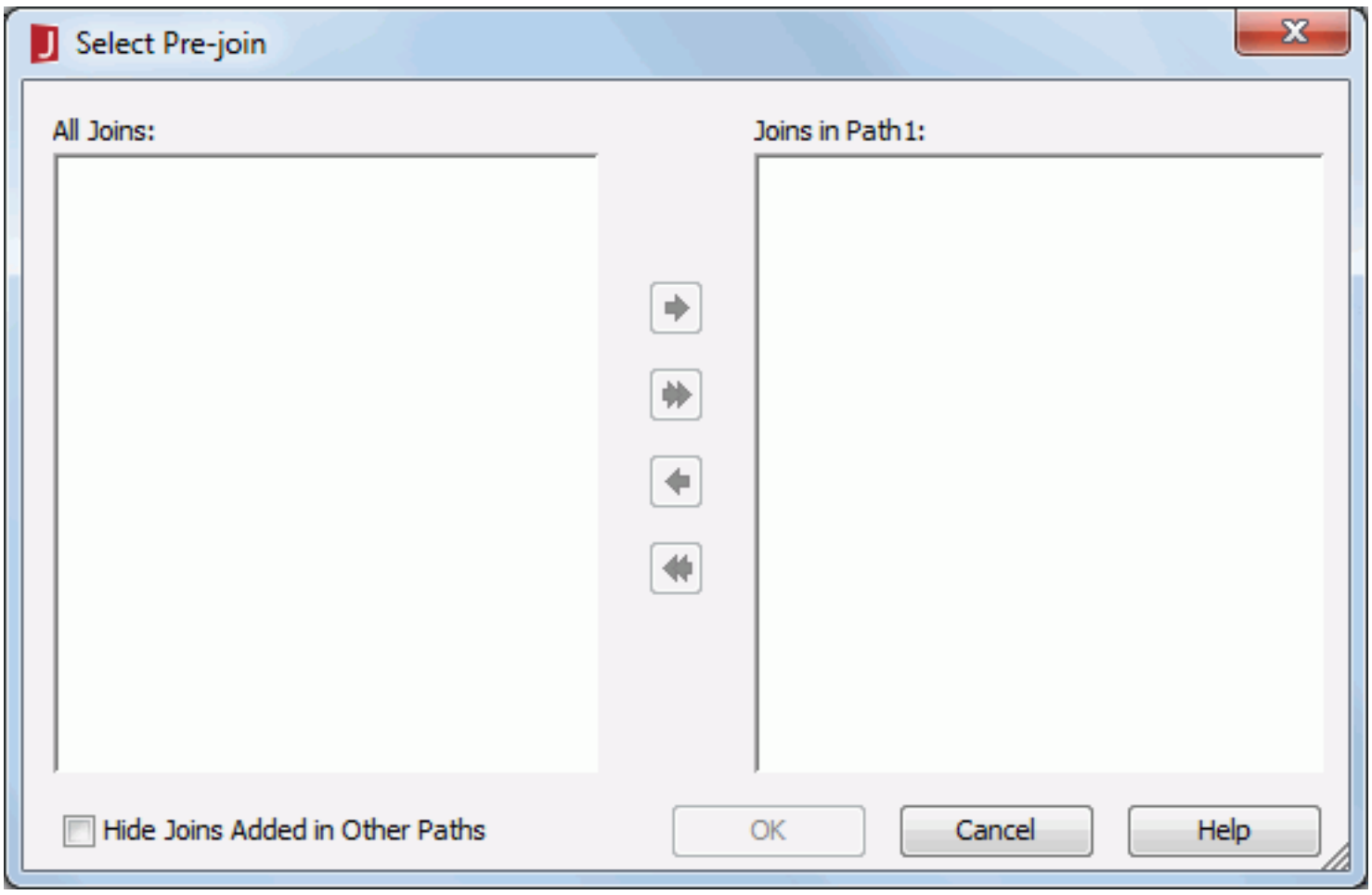
Displays the help document about this feature.

Related topics:

- Specifying the Select N condition

Select Pre-join dialog

This dialog appears when you click the New button in the Save Pre-join Path dialog, input a path name and then click OK. It helps you to define new pre-join paths in the current catalog.



The following are details about options in this dialog:

All Joins

Lists all the pre-joins in the connection.

Joins in <path name>

Lists all joins that are added to the new path.

Hide Joins Added in Other Paths

Specifies whether to hide those joins that have been added in other paths.

Adds a join to the new path.

Adds all listed join from the new path.

Removes a join from the path.

Removes all joins from the path.

OK

Applies the changes and closes the dialog.

Cancel

Does not retain any changes and closes the dialog.

Help

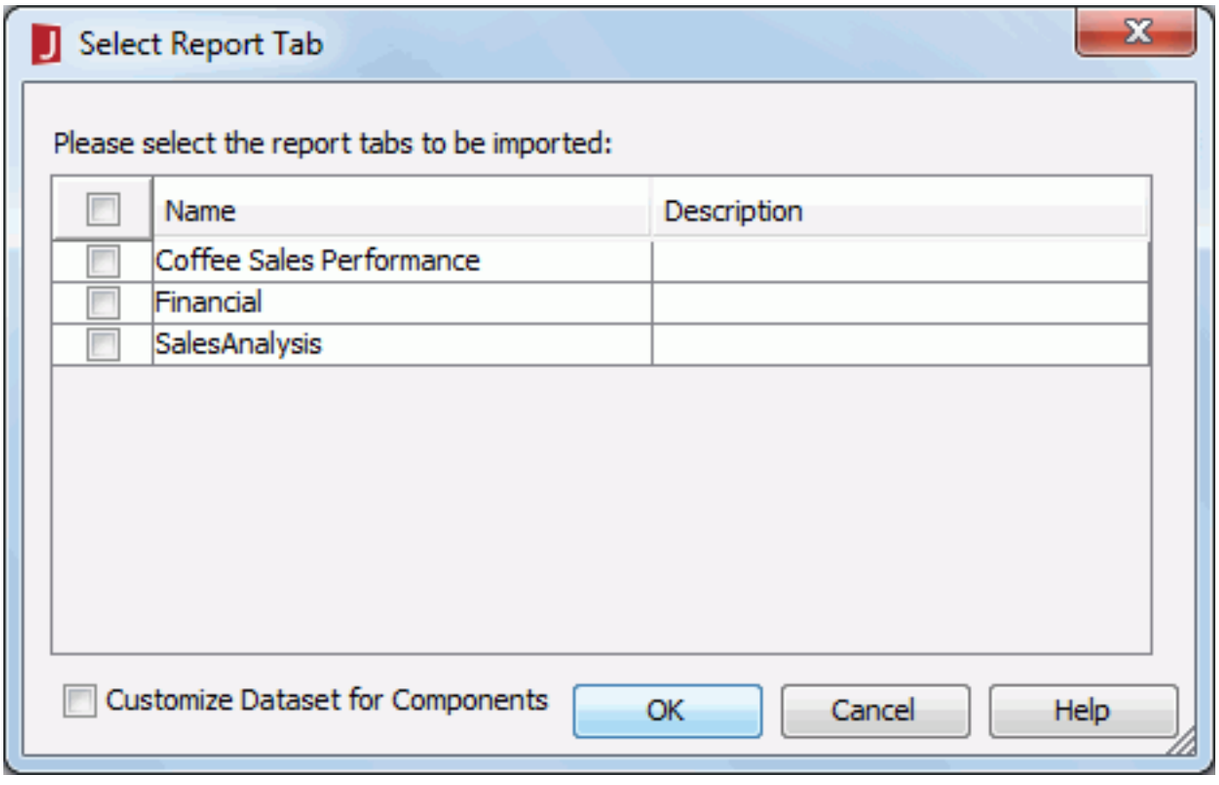
Displays the help document about this feature.

Related topics:

- Defining the join relationships between tables

Select Report Tab dialog

This dialog appears when you click File > Import From > JReport Page Report, specify a report in the Open Report dialog and then click Open. It helps you to select which report tabs in the specified report you want to import.



The following are details about options in this dialog:

Name

Displays the names of the report tabs.

Description

Displays the description of the report tabs.

Customize Dataset for Components

Specifies to customize datasets used by components within the selected report tabs. If this option is checked, when you click OK in the dialog, the Customize Dataset dialog will be displayed.

OK

Applies the changes and closes the dialog.

Cancel

Does not retain any changes and closes the dialog.

Help

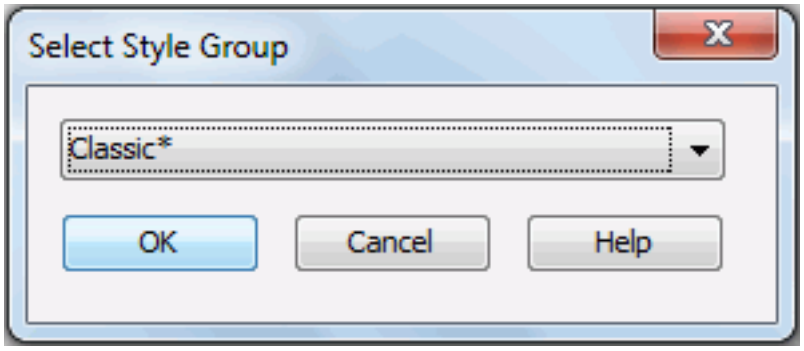
Displays the help document about this feature.

Related topics

- Importing report tabs into a page report

Select Style Group dialog

Once a page report tab or a web report is applied with a style group, when you view, export or publish the report, you will be asked to specify a style group for the report (to apply a style group to a page report tab or a web report, specify its Style Group property in the Others category in the Report Inspector). A style group can hold a large amount of predefined object properties, giving you more controls over the appearance and presentation of your report. By simply selecting a style group at runtime, you can change the look and feel of your report in a dramatic way.



OK

Applies the style group to the report and closes the dialog.

Cancel

Does not accept the style group you've selected and closes the dialog.

Help

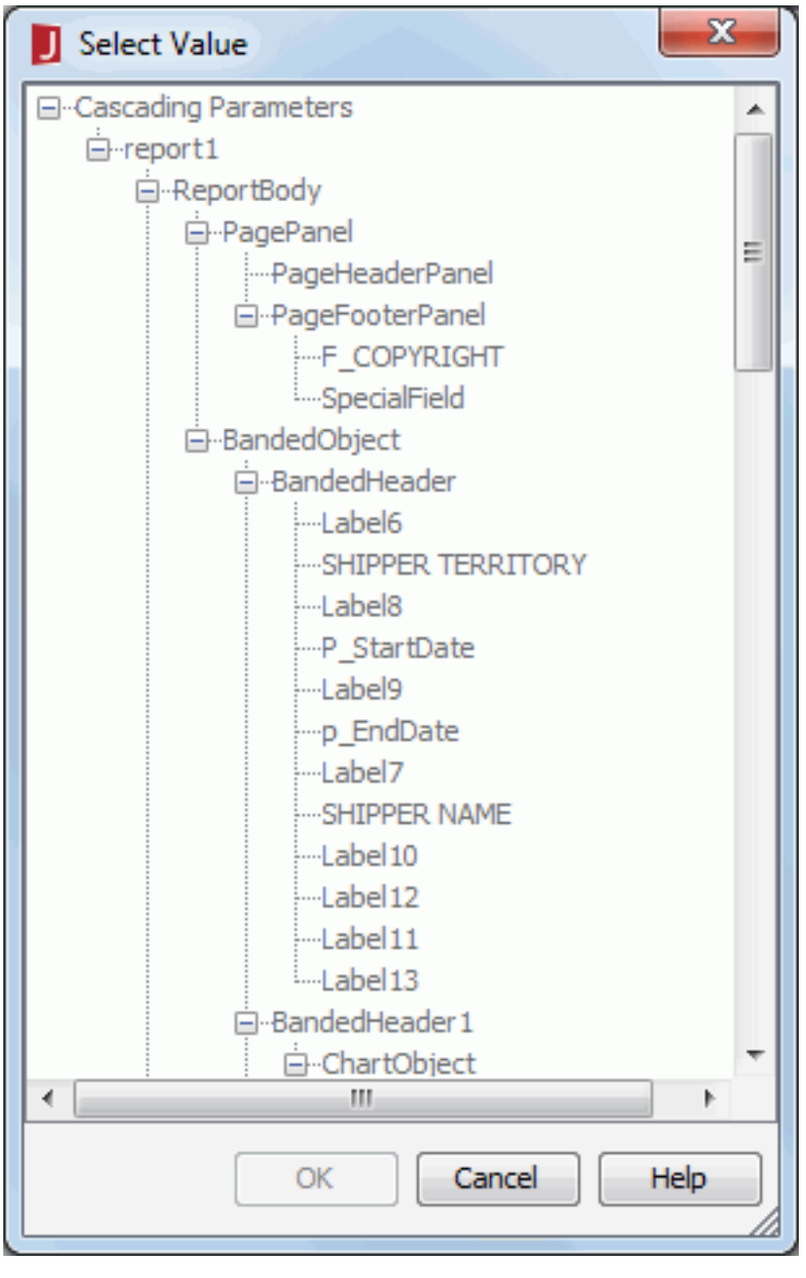
Displays the help document about this feature.

Related topics:

- Applying styles

Select Value dialog

The dialog is displayed when you click the button ... in the Advanced tab for the Report link type in the Insert Link dialog for page report or web report/library component, or in the Edit Link dialog for page report or web report/library component. It helps you to select a filter object such as filter control or slider to pass values between the filter objects in the primary report and the linked report.



The resources panel lists all the objects in the report. Only filter objects are available for selection. Select one to use.

OK

Selects a filter object and closes the dialog.

Cancel

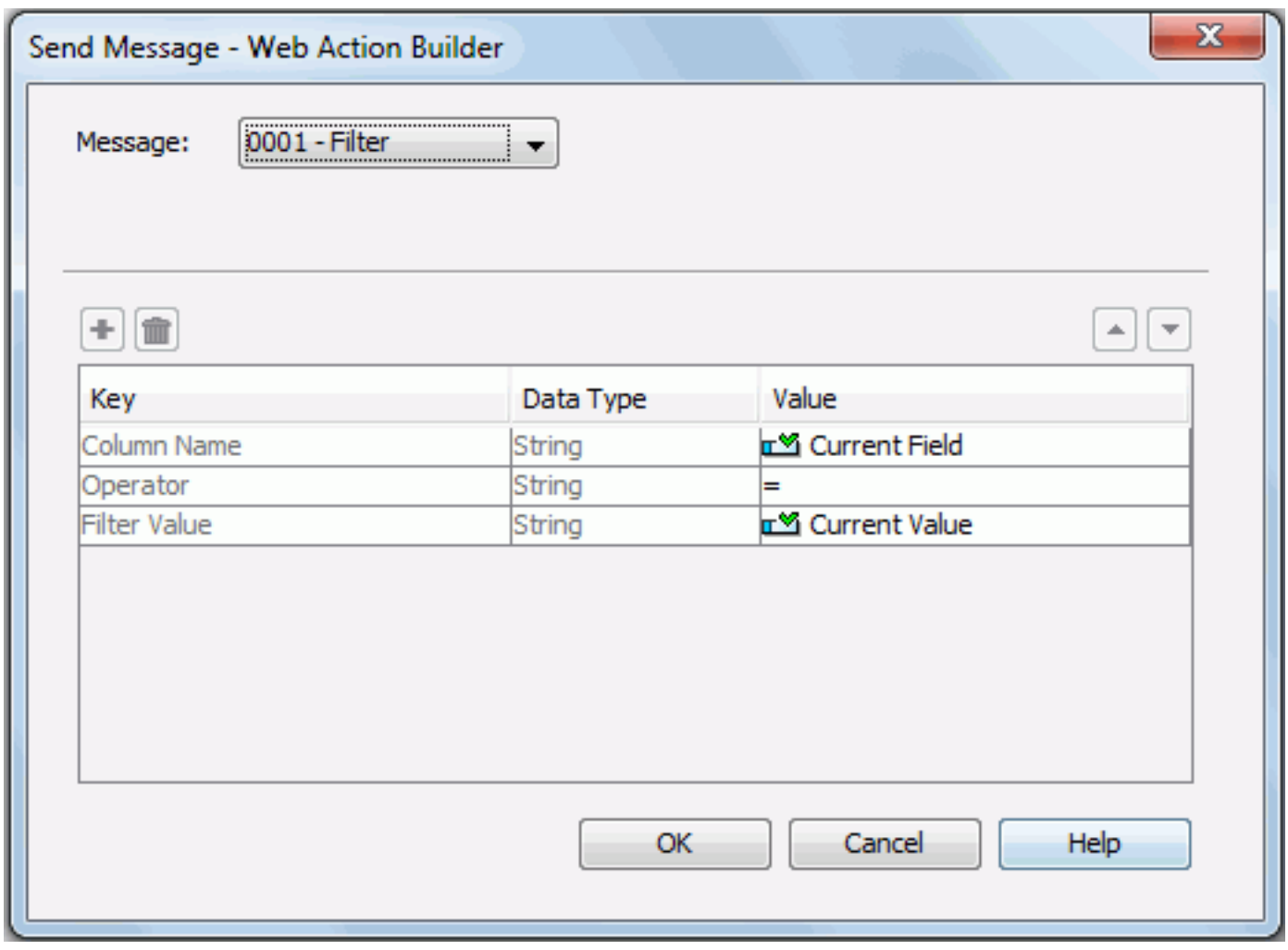
Cancels the selection and closes the dialog.

Help

Displays the help document about this feature.

Send Message - Web Action Builder dialog

This dialog appears when you select *Send Message and then click OK in the Web Action List dialog. It helps you to build a Send Message web action on an object so as to send out a message when the event bound with the action is triggered at runtime.



The following are details about options in the dialog:

Message

Specifies the message that will be sent out.

- **0001 - Filter**
  Specifies to send out the built-in Filter message, which will ask the data component who receives the message to filter itself based on the filter condition defined in the message.
- **0002 - Sort**
  Specifies to send out the built-in Sort message, which will ask the data component who receives the message to sort itself based on the sort condition defined in the message.
- **0003 - Parameter**
  There are two types:
  - **Automatic**
    Specifies to send out the built-in Parameter message with all the parameter values used in the current library component, which will ask the library component in which the data component that receives the message is to re-run itself using the same parameter values.
  - **Customized**
    Specifies to send out the built-in Parameter message, which will ask the library component in which the data component that receives the message is to re-run itself using the parameter value defined in the message. In this case, you need to define a parameter with a value.

  For compatibility with earlier versions, the previous 0003 - Parameter message will be converted to the Customized type and 0004 - Parameters to the Automatic type.
- **User Defined**
  Specifies to send out a user defined message.
  - **ID**
    Specifies the ID of the message.
  - **Name**
    Specifies the name of the message.

Adds a new message key-value line. Not available to built-in messages.

Removes the selected message key-value line. Not available to built-in messages.

Moves the selected message key-value line up a step. Not available to built-in messages.

Moves the selected message key-value line down a step. Not available to built-in messages.

Key

Specifies the key of the message. For a built-in message, the key cannot be changed.

Data Type

Specifies the data type of the value. For a built-in message, the data type cannot be changed.

Value

Specifies the value of the key.

OK

Accepts the changes and closes this dialog.

Cancel

Does not retain any changes and closes the dialog.

Help

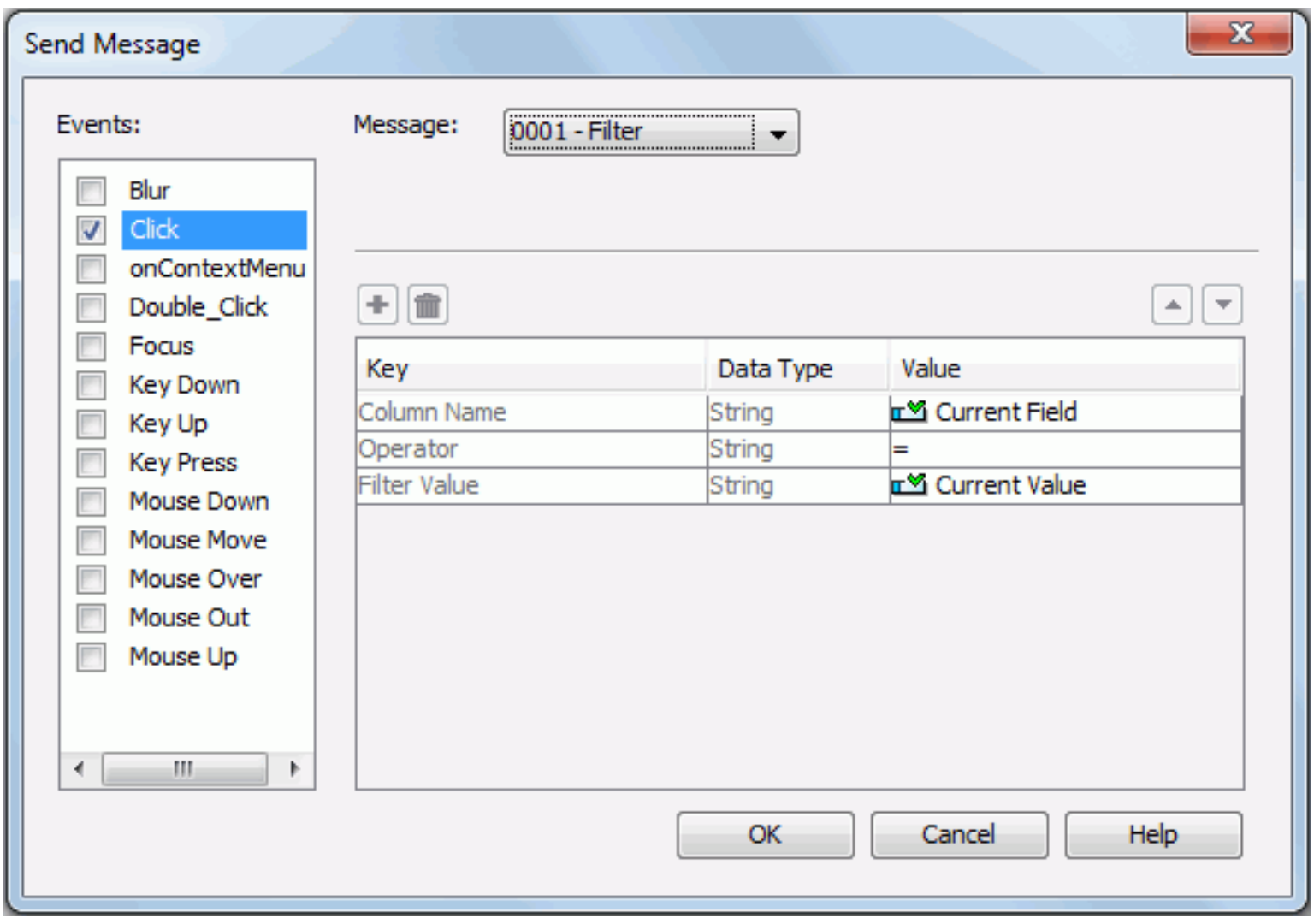
Displays the help document about this feature.

Related topics:

- Sending out messages

Send Message dialog

This dialog appears when you right-click some object in a library component and select Send Message > Customize from the shortcut menu. It helps you to customize the message that will be sent out when a specific event occurs on the object.



The following are details about options in the dialog:

Events

Specifies the trigger event by checking the checkbox ahead of the event. Click on the event to specify the message it is going to trigger.

Message

Specifies the message that will be sent out.

- **0001 - Filter**
  Specifies to send out the built-in Filter message, which will ask the data component who receives the message to filter itself based on the filter condition defined in the message.
- **0002 - Sort**
  Specifies to send out the built-in Sort message, which will ask the data component who receives the message to sort itself based on the sort condition defined in the message.
- **0003 - Parameter**
  There are two types:
  - **Automatic**
    Specifies to send out the built-in Parameter message with all the parameter values used in the current library component, which will ask the library component in which the data component that receives the message is to re-run itself using the same parameter values.
  - **Customized**
    Specifies to send out the built-in Parameter message, which will ask the library component in which the data component that receives the message is to re-run itself using the parameter value defined in the message. In this case, you need to define a parameter with a value.

  For compatibility with earlier versions, the previous 0003 - Parameter message will be converted to the Customized type and 0004 - Parameters to the Automatic type.
- **User Defined**
  Specifies to send out a user defined message.
  - **ID**
    Specifies the ID of the message.
  - **Name**
    Specifies the name of the message.

Adds a new message key-value line. Not available to built-in messages.

Removes the selected message key-value line. Not available to built-in messages.

Moves the selected message key-value line up a step. Not available to built-in messages.

Moves the selected message key-value line down a step. Not available to built-in messages.

Key

Specifies the key of the message. For a built-in message, the key cannot be changed.

Data Type

Specifies the data type of the value. For a built-in message, the data type cannot changed.

Value

Specifies the value of the key.

OK

Accepts the changes and closes this dialog.

Cancel

Does not retain any changes and closes the dialog.

Help

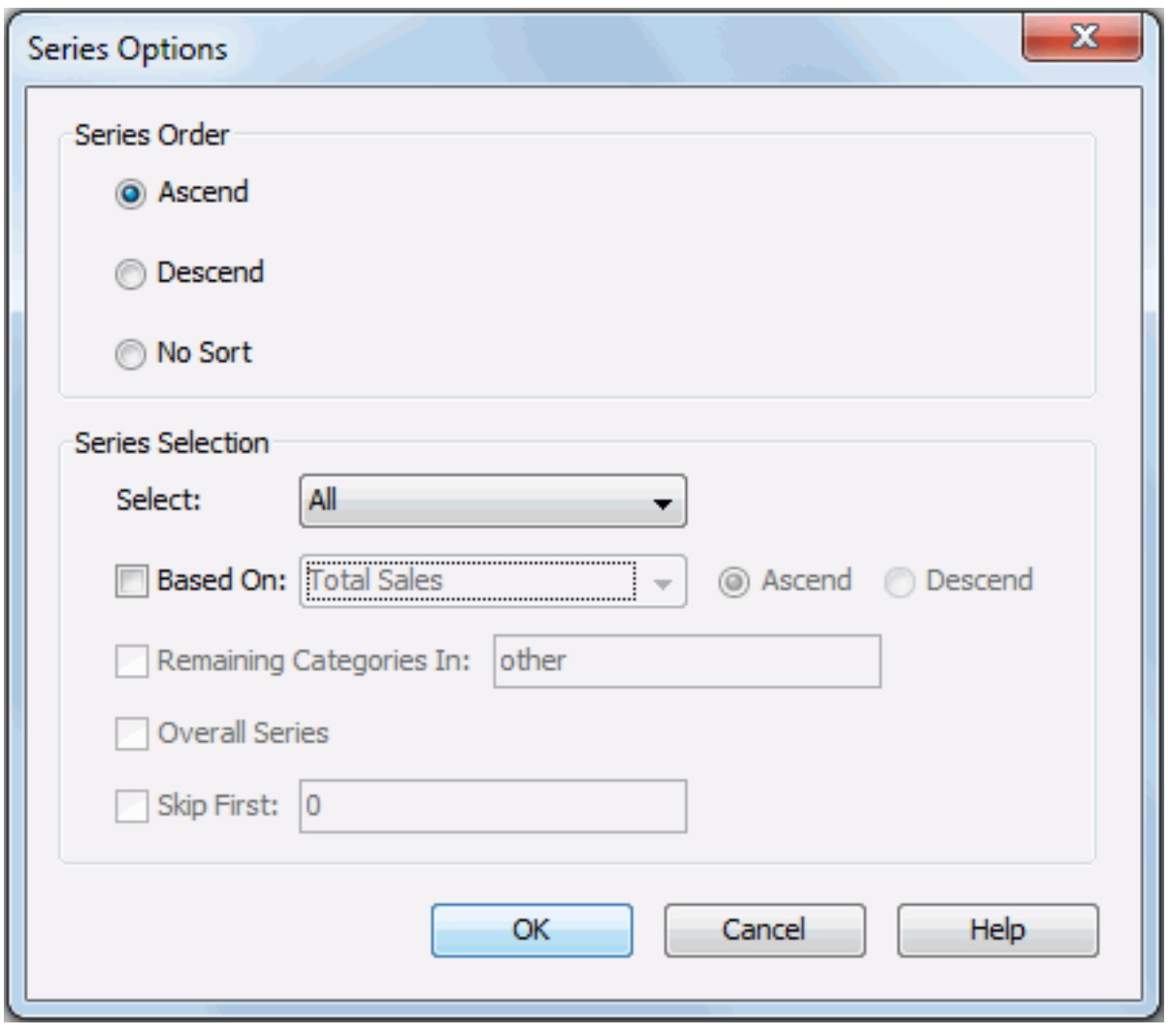
Displays the help document about this feature.

Related topics:

- Sending out messages

Series Options dialog

The dialog appears when you click the Order/Select N button below the Series box in the Display screen of the chart wizard. It helps you to specify the sort order of the series values and define the number of the series values that will be displayed in a chart.



The following are details about options in the dialog:

Series Order

Specifies in which manner to sort the series values.

- **Ascend**
  Values will be sorted in an ascending order (A, B, C).
- **Descend**
  Values will be sorted in a descending order (C, B, A).
- **No Sort**
  Values will be sorted in the original order in database.

Series Selection

Specifies the number of series values that will be displayed in the chart.

- **Select**
  Specifies the Select N condition to define the number of the series values that will be displayed.
  - **All**
    If selected, all series values in the chart will be displayed.
  - **Top N**
    If selected, specify a number in the field to the right and the first N series values in the chart will be displayed. You can also select a parameter from the drop-down list, which returns an integer, and then specify a value for the parameter when viewing the report to dynamically define the Top N condition.
  - **Bottom N**
    If selected, specify a number in the field to the right and the last N series values in the chart will be displayed. You can also select a parameter from the drop-down list which returns an integer, and then specify a value for the parameter when viewing the report to dynamically define the Bottom N condition.
- **Based On**
  If checked, the series values will be sorted by values of the summary selected with the direction specified. If unchecked, the series values will be sorted by the order specified in the Series Order box of the dialog.
- **Remaining Series In**
  Enabled only when Top N or Bottom N is selected from the Select drop-down list. Check this option and then type a character string in the text field to group all the series values beyond the top/bottom N range.
- **Overall Series**
  Not supported on series values.
- **Skip First**
  If you check the Skip First option and input a number M in the text field to the right, then the first M series values in the chart will be skipped and the Select N condition will take effect beginning with M +1. The skipped values will be included in the Remaining Series group together with all the series values beyond the top/bottom N range. Enabled when Top N or Bottom N is selected from the Select drop-down list.

OK

Accepts the changes and closes the dialog.

Cancel

Discards the changes and closes the dialog.

Help

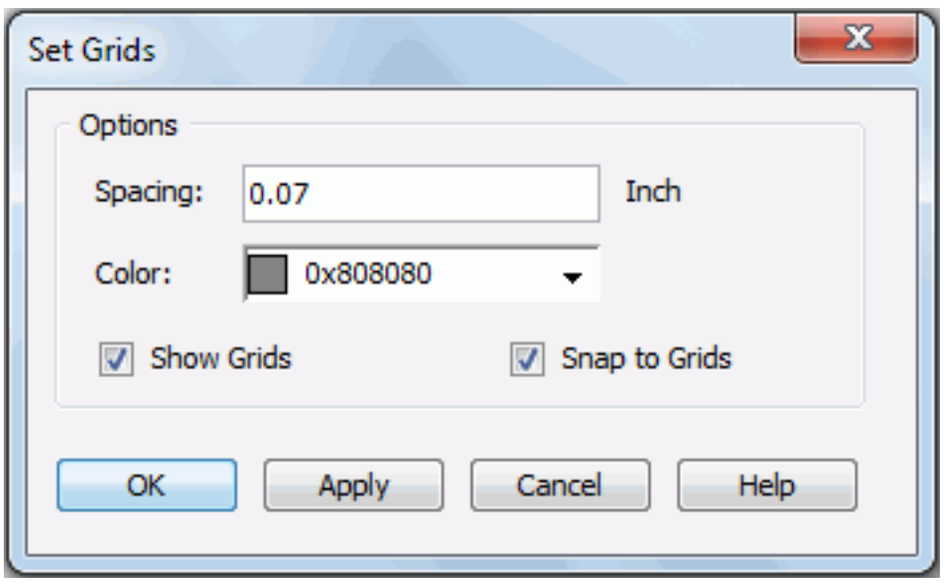
Displays the help document about this feature.

Related topics:

- Setting Order/Select N condition for category/series/group field

Set Grids dialog

This dialog appears when you click View > Grid Size on the menu bar. It helps you to specify the grid settings.



The following are details about options in the dialog:

Spacing

Specifies the size of the grids in the design area, in inches.

Color

Specifies the color of the grids.

To specify the color, select the desired color from the drop-down list or enter a hexadecimal RGB value (for example, 0xff0000) in the text box directly. If the given ones cannot meet your requirement, you can select Custom from the drop-down list to open the Pick a Color dialog, with which you can set a full range of color.

Show Grids

Specifies whether to show grids in the report.

Snap to Grids

Specifies whether to snap and lock objects with the grids when moving and placing objects around.

OK

Accepts changes and closes the dialog.

Apply

Applies the grid settings and closes the dialog.

Cancel

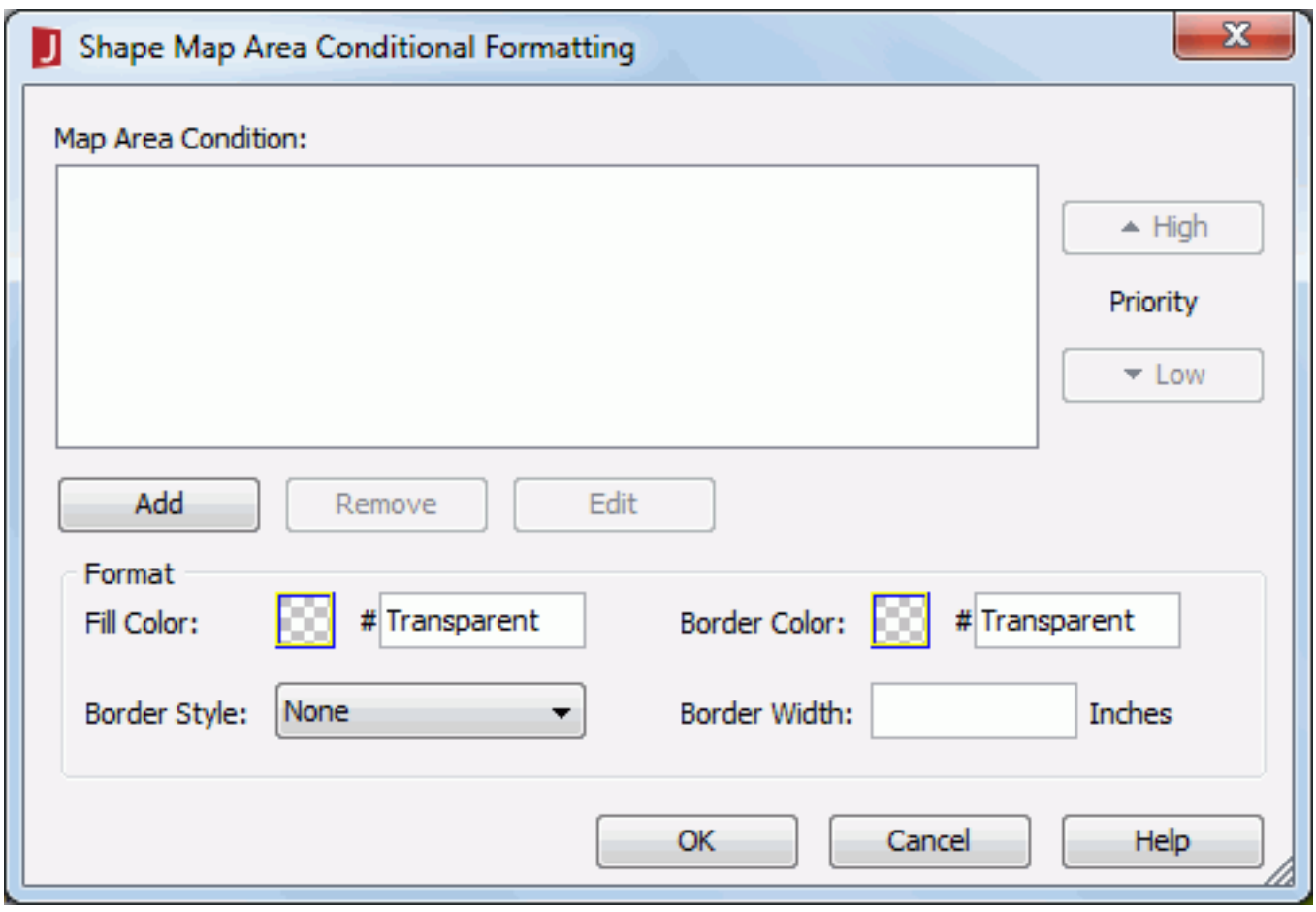
Does not retain the changes and closes the dialog.

Help

Displays the help document about this feature.

Shape Map Area Conditional Formatting dialog

The dialog appears when you click Format > Conditional Formatting on the menu bar, or click the Conditional Formatting button on the toolbar of the Shape Map Editor window. It helps you to set some conditional formats on the shape map areas.



The following are details about options in the dialog:

Map Area Condition

Lists the conditions.

- **Priority**
  Specifies the priority of each condition.
  - **High**
    Moves a condition up for a higher priority.
  - **Low**
    Moves a condition down for a lower priority.
- **Add**
  Adds a new condition in the Edit Conditions dialog.
- **Remove**
  Removes the selected condition.
- **Edit**
  Edits the selected condition.

Format

Specifies the format for the areas defined in the selected condition.

- **Fill Color**
  Specifies the filling color for the areas.

  To specify the color, click the color image button and select a color swatch or type the hexadecimal value of the desired color (for example, 0xff0000) in the text box directly. If the color swatches cannot meet your requirement, you can click More Colors to open the Pick a Color dialog, with which you can set a full range of color.
- **Border Color**
  Specifies the border color for the areas.
- **Border Style**
  Specifies the border line style for the areas.
- **Border Width**
  Specifies the border width for the areas, in inches.

Note: The format set here corresponds with some formulas. After you save the map and the corresponding catalog, and then load the catalog again, you will find some formulas are developed in the catalog. You can edit the formulas in the Formula Editor. However, the change you make to the formulas will not be reflected in the Map Area Conditional Formatting dialog.

OK

Applies the changes and closes the dialog.

Cancel

Cancels the changes and exits the dialog.

Help

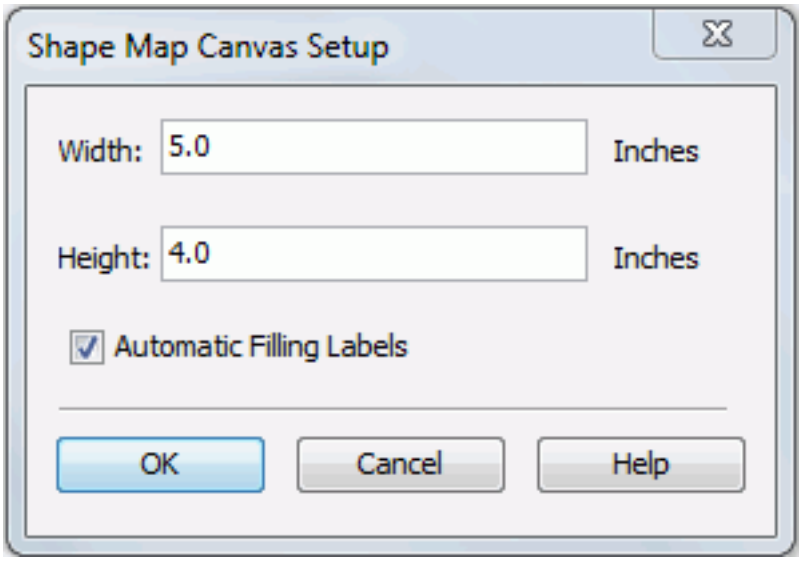
Displays the help document about this feature.

Related topics:

- Adding conditional formats to shape map areas

Shape Map Canvas Setup dialog

The dialog appears when you click File > Canvas Setup on the menu bar, or the Canvas Setup button on the toolbar of the Shape Map Editor window. It helps you to change the width and height of the shape map object.



The following are details about options in the dialog:

Width

Specifies the width of the map object, in inches.

Height

Specifies the height of the map object, in inches.

Automatic Filling Labels

If checked, the label will not be displayed when the size of the area is changed to be smaller than the label.

OK

Applies the changes and closes the dialog.

Cancel

Does not retain any changes and closes the dialog.

Help

Displays the help document about this feature.

Shape Map Data Binding Wizard

The wizard appears when you click Insert > Bind Data on the menu bar, or click the Bind Data button on the toolbar of the Shape Map Editor window. It helps you to bind a dataset to a shape map and make values displayed on the shape map areas according to your requirements.

The wizard consists of the following screens:

- Data screen
- Group screen
- Summary screen
- Filter screen

Back

Goes back to the previous screen.

Next

Goes to the next screen.

Finish

Finishes data binding and closes this wizard.

Cancel

Does not retain changes and closes this wizard.

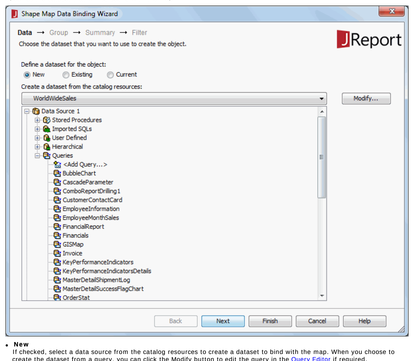
Help

Displays the help document about this feature.

Data screen

Define a dataset for the object

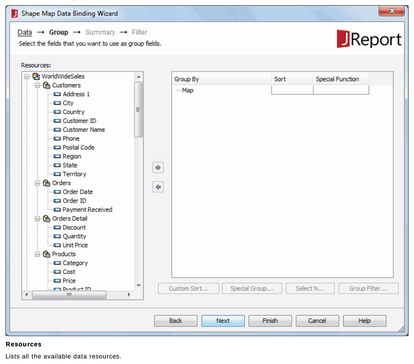
Specifies the dataset that will be bound with the map.



- **New**
  If checked, select a data source from the catalog resources to create a dataset to bind with the map. When you choose to create the dataset from a query, you can click the Modify button to edit the query in the Query Editor if required.
- **Existing**
  If checked, select a dataset from the ones existing in the open report to bind with the map. Click the Modify button to edit the dataset in the Dataset Editor if required.
- **Current**
  If checked, the current dataset used by the parent object will be applied to the map.

Group screen

Specifies the field to group the data.



Resources

Lists all the available data resources.

Adds the selected field as the group by field.

Removes the selected group by field that is not required.

Group By

Lists the field that is used to group data.

Sort

Specifies how groups will be sorted.

- **Ascend**
  Groups will be sorted in an ascending order (A, B, C).
- **Descend**
  Groups will be sorted in a descending order (C, B, A).
- **No Sort**
  Groups will be sorted in the original order in database.
- **Special Group**
  Opens the User Defined Group dialog to define grouping information.
- **Custom Sort**
  Opens the Custom Sort dialog to set how groups will be sorted.

Special Function

If the group by field is of Numeric/String/Date/Time type, you can select a special function for the field in the Special Function column to further specify to which level the data will be grouped by.

If Customize is selected, the Customized Function dialog will be displayed, in which you can set the function by your own.

Custom Sort

Specifies how to sort the groups. Activated only when you have clicked Custom Sort from the Sort column to define the sorting manner of groups for the selected group level.

Special Group

Specifies how to group your information. Activated only when you have clicked Special Group from the Sort column to define a special group.

Select N

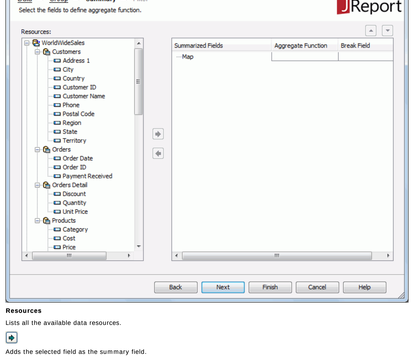
Opens the Select N dialog to specify the Select N condition.

Group Filter

Opens the Group Filter dialog to specify the group filter condition.

Summary screen

Specifies the fields on which to create aggregate functions.



Resources

Lists all the available data resources.

Adds the selected field as the summary field.

Removes the selected summary field that is not required.

Moves the specified summary field one step up.

Moves the specified summary field one step down.

Summarized Fields

Lists all the fields that you want to display and to be summarized.

Aggregate Function

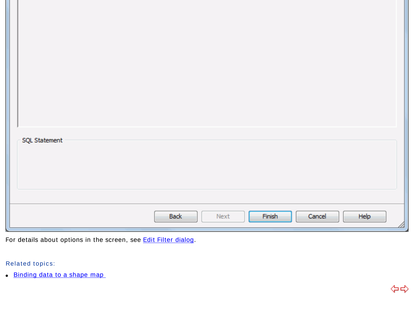
Specifies the function to summarize the selected field. For details about each function, see Math functions.

Break Field

Displays a defined field on which the summary will be calculated.

Filter screen

Specifies to filter the data displayed on the map areas.



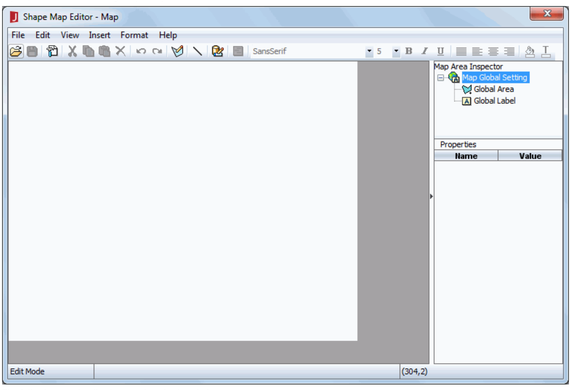
For details about options in the screen, see Edit Filter dialog.

Related topics:

- Binding data to a shape map

Shape Map Editor window

This window appears when you click Insert > Shape Map on the menu bar, or click the Insert Shape Map button on the Insert toolbar, or right-click a shape map and select Format Shape Map from the shortcut menu. It helps you to insert a shape map into a report, or to edit an existing shape map.



The following are details about options in the window:

File menu

- **Open**
  Specifies to import a .shp or .xml file which has been defined with some area information to the map object.
- **Save**
  Accepts and saves the changes on the map object.
- **Save As**
  Saves the map object to an .xml file with the name you specify.
- **Canvas Setup**
  Opens the Shape Map Canvas Setup dialog to change the width and height of the map object.
- **Close**
  Closes the window.

Edit menu

- **Undo**
  Reverses a previous action.
- **Redo**
  The counter-operation of Undo.
- **Cut**
  Erases an object and places it in the clipboard.
- **Copy**
  Copies an object and places it in the clipboard.
- **Paste**
  Takes an object from the clipboard and places it into the report.
- **Delete**
  Deletes a selected object.
- **Insert Mode**
  If checked, the window is in insert mode, and you can add areas into the map manually; if unchecked, the window is in edit mode, and you can edit areas as required.
- **Reset All**
  Opens the Reset All dialog to reset the properties for map areas including the labels and summary fields inside the areas globally.
- **Hide Duplicate Labels**
  If clicked, for map areas with the same name, only the label and summary field for the biggest area will be shown.

View menu

- **Show Area Inspector**
  Specifies whether or not to show the Map Area Inspector panel in the Shape Map Editor window.
- **Show Area Labels**
  Specifies whether or not to show labels and summary fields in map areas in the Shape Map Editor window.

Insert menu

- **Background Image**
  Specifies to select an image as background of the map.
- **Match Background Image**
  Resizes the map object to make it match the size of the background image. Activated only when the size of the background image is not the same as that of the map object.
- **Label**
  Inserts a label to the map object.
- **Line**
  Inserts a line to the map object.
- **Bind Data**
  Opens the Shape Map Data Binding Wizard to bind data to the map object.

Format menu

- **Conditional Formatting**
  Opens the Shape Map Area Conditional Formatting dialog to add some conditional formats to the map areas.
- **Align**
  - **Left**
    Aligns the content in the selected object to the left boundary of the object.
  - **Center**
    Aligns the content in the selected object to the center of the object.
  - **Right**
    Aligns the content in the selected object to the right boundary of the object.
  - **Justify**
    Adjusts horizontal spacing so that the content is aligned evenly along both the left and right margins in the selected object.
- **Text Style**
  - **Bold**
    Specifies whether or not to bold the text in the selected object.
  - **Italic**
    Specifies whether or not to italicize the text in the selected object.
  - **Underline**
    Specifies whether or not to underline the text in the selected object.

Help menu

- **Help**
  Displays the help document about this feature.

Toolbar

The following are commands on the toolbar:

- **Open**
  Specifies to import a .shp or .xml file which has been defined with some area information to the map object.
- **Save**
  Accepts and saves the changes on the map object.
- **Canvas Setup**
  Opens the Map Canvas Setup dialog to change the width and height of the map object.
- **Cut**
  Erases an object and places it on the clipboard.
- **Copy**
  Copies an object and places it on the clipboard.
- **Paste**
  Takes an object from the clipboard and places it into the map object.
- **Delete**
  Deletes a selected object.
- **Undo**
  Reverses a previous action.
- **Redo**
  The counter-operation of Undo.
- **Insert Mode**
  If pressed, the window is in insert mode, and you can insert areas into the map; otherwise, the window is in edit mode, and you can edit the map areas.
- **Line**
  Inserts a line into the map object.
- **Bind Data**
  Opens the Map Data Binding Wizard to bind data to the map areas.
- **Conditional Formatting**
  Opens the Map Area Conditional Formatting dialog to add some conditional formats to the map areas.
- Arial **Font Face Box**
  Specifies the font face of the text in the selected object.
- 9 **Font Size Box**
  Specifies the font size of the text in the selected object.
- **Bold**
  Specifies whether or not to bold the text in the selected object.
- **Italic**
  Specifies whether or not to italicize the text in the selected object.
- **Underline**
  Specifies whether or not to underline the text in the selected object.
- **Justify**
  Adjusts horizontal spacing so that the content is aligned evenly along both the left and right margins in the selected object.
- **Left**
  Aligns the content in the selected object to the left boundary of the object.
- **Center**
  Aligns the content in the selected object to the center of the object.
- **Right**
  Aligns the content in the selected object to the right boundary of the object.
- **Background Color**
  Specifies the background color of the selected object.
- **Foreground Color**
  Specifies the foreground color of the selected object.

Map Area Inspector

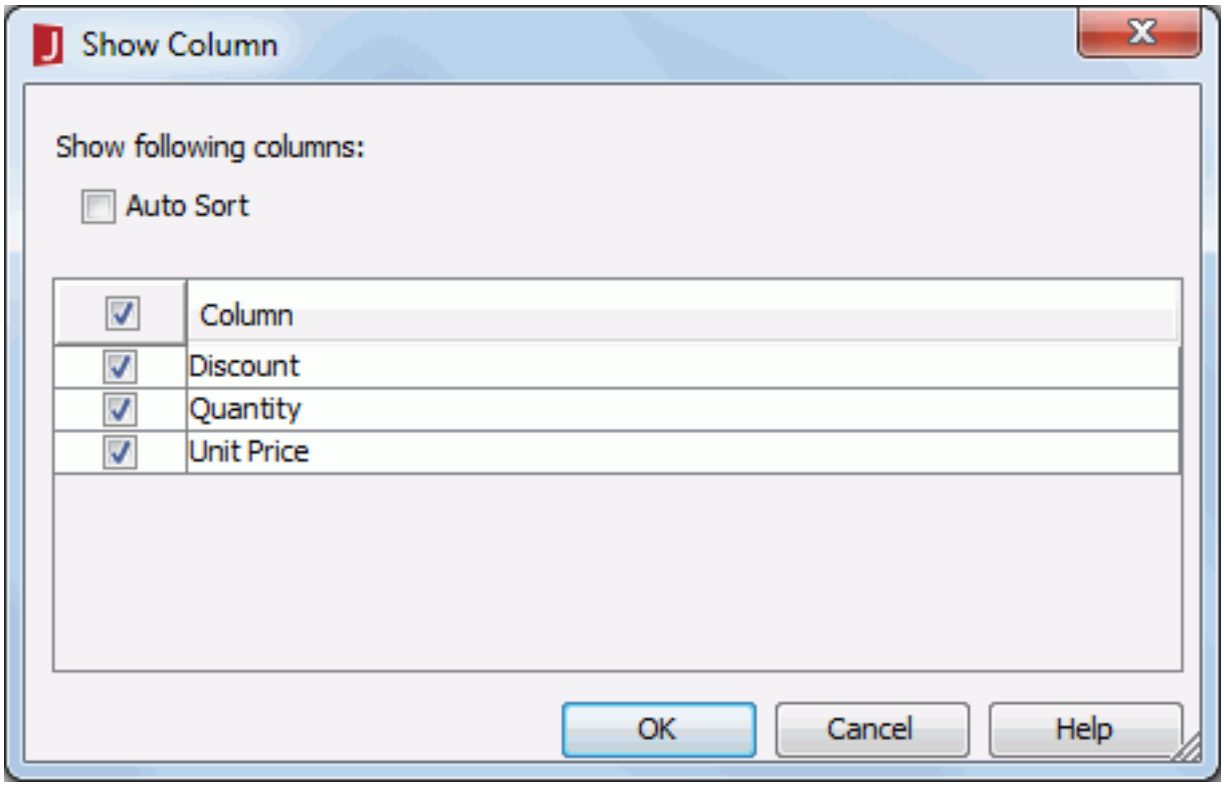
On the right part of the Shape Map Editor window, there is a Map Area Inspector panel, where you can control properties of any area, label, summary field and line in the map object.

Related topics

- Inserting a shape map

Show Column dialog

This dialog appears when you right-click a table and select Show Column from the shortcut menu. It lists all the columns in the table, with text in cells of the table header row representing corresponding columns. You can choose to show or hide some of them by checking or unchecking the checkboxes ahead of the desired column names.



Auto Sort

If checked, the columns will be sorted alphabetically; otherwise, the columns will be listed according to their order in the table.

OK

Applies the changes and closes the dialog.

Cancel

Cancels the changes and exits the dialog.

Help

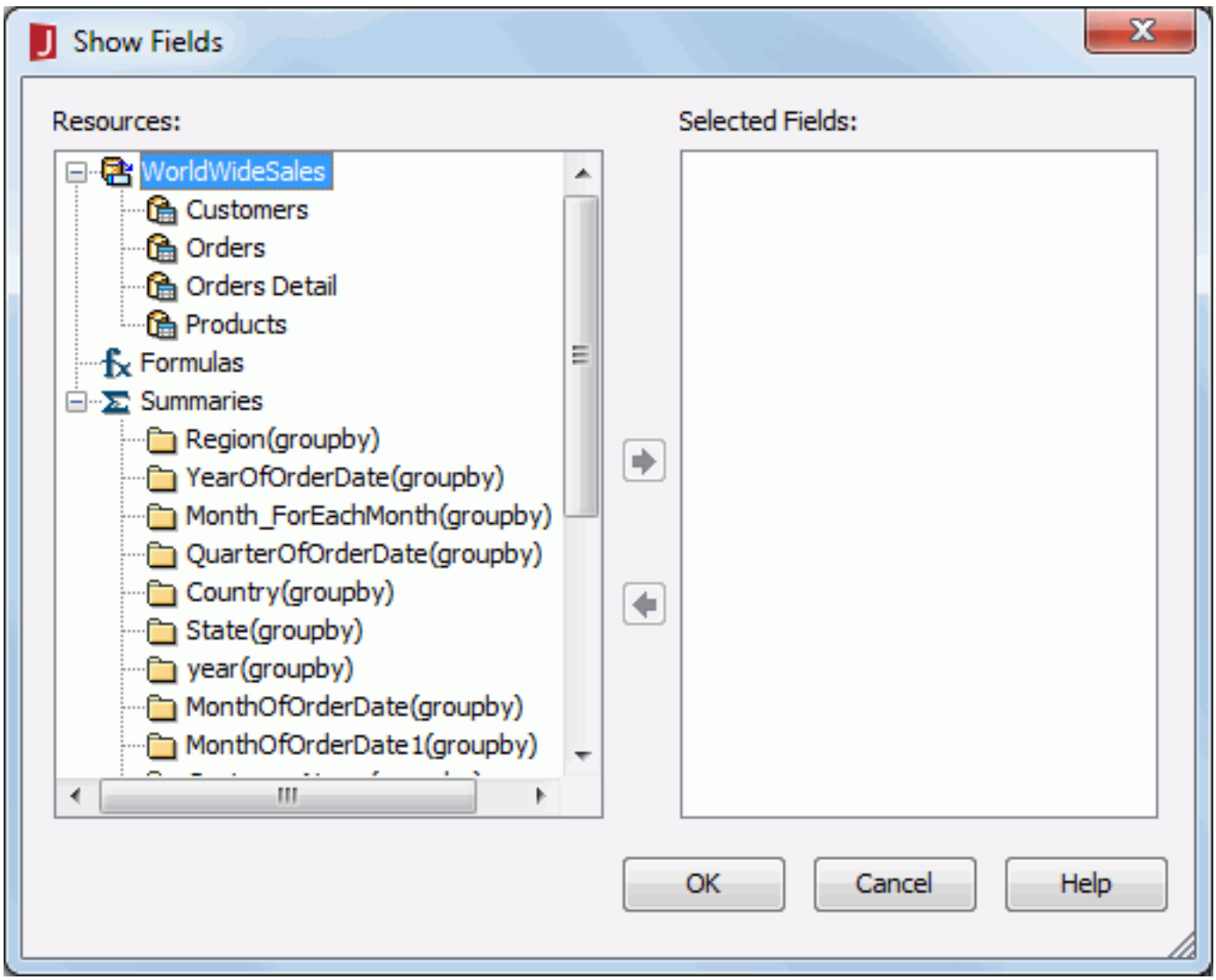
Displays the help document about this feature.

Related topics:

- Managing columns/rows in a table - Showing certain columns

Show Fields dialog

The dialog appears when you select Customized from the drop-down list of the Show Fields property, which is in the Hint category of the chart paper of a heat map in the Report Inspector. It allows you to select fields to show as the heat map hint, which displays when the mouse pointer points at the data marker in Designer view mode, in HTML result, or at runtime on JReport Server.



From the Resources box select a field and click to add it into the right box to make its value shown as the hint. Repeat this to add more fields. The names of the selected fields will be displayed in the value cell of the Show Fields property, separated by ",".

OK

Applies the changes and closes the dialog.

Cancel

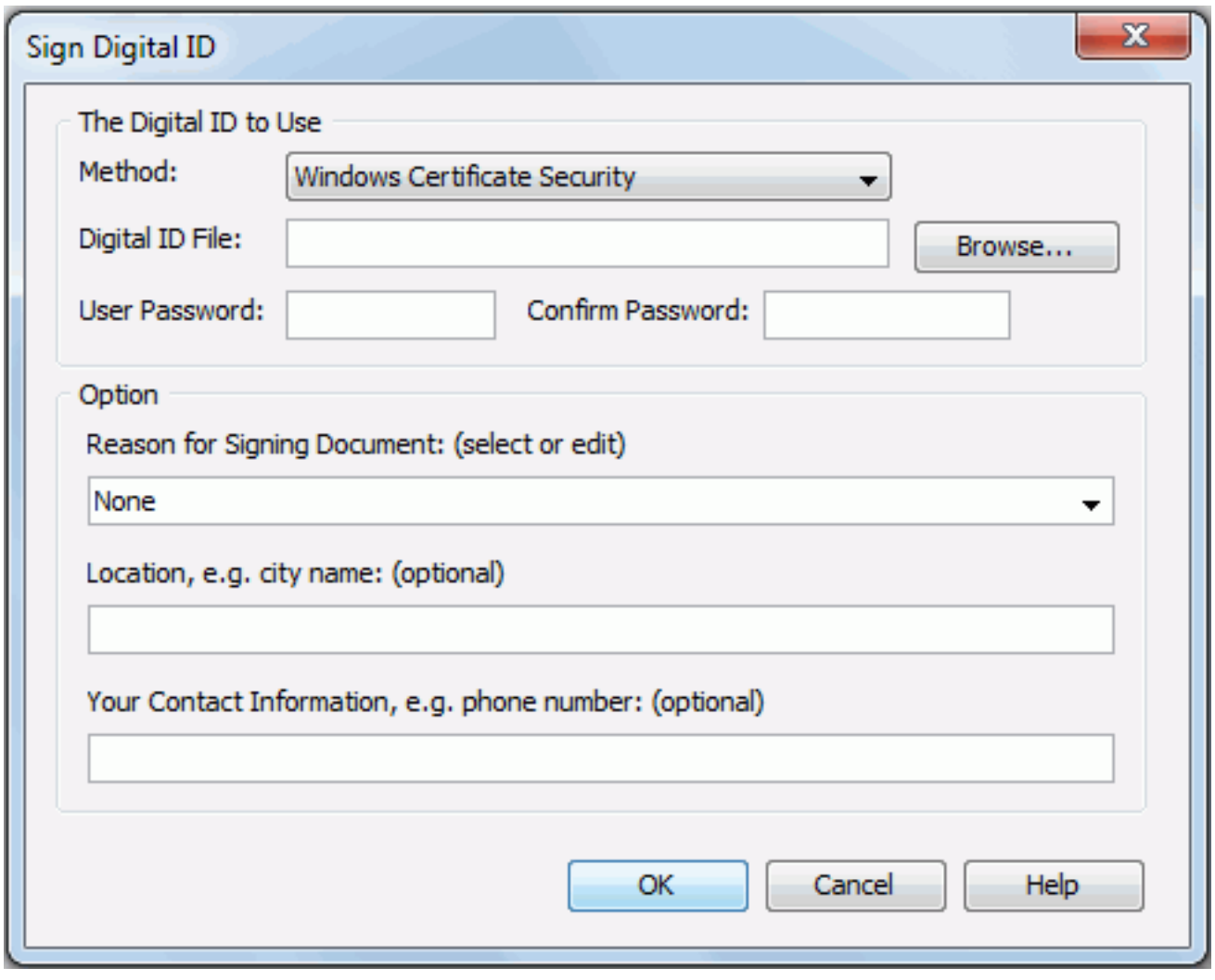
Does not retain any changes and closes the dialog.

Help

Displays the help document about this feature.

Sign Digital ID dialog

This dialog appears when you click the Setting button after checking the Sign checkbox in the Export to PDF dialog. This dialog allows you to configure the signature for the PDF file.



The following are details about options in the dialog:

Method

Specifies the signing digital signature method from the three methods, which are provided by Adobe Acrobat software. It could be Windows Certificate Security, Default Certificate Security and VeriSign Digital Signatures.

Digital ID File

Specifies the digital ID file. Type the file path directly in the text field or click the Browse button beside it to specify the directory.

User Password

Specifies the password for the digital ID file.

Confirm Password

Confirms the password by inputting it again.

Option

- **Reason for Signing Document**
  Specifies the reason for signing the document.
- **Location**
  Specifies the location.
- **Your Contact Information**
  Specifies the contact information.

OK

Applies all changes and closes the dialog.

Cancel

Cancels the changes and exits the dialog.

Help

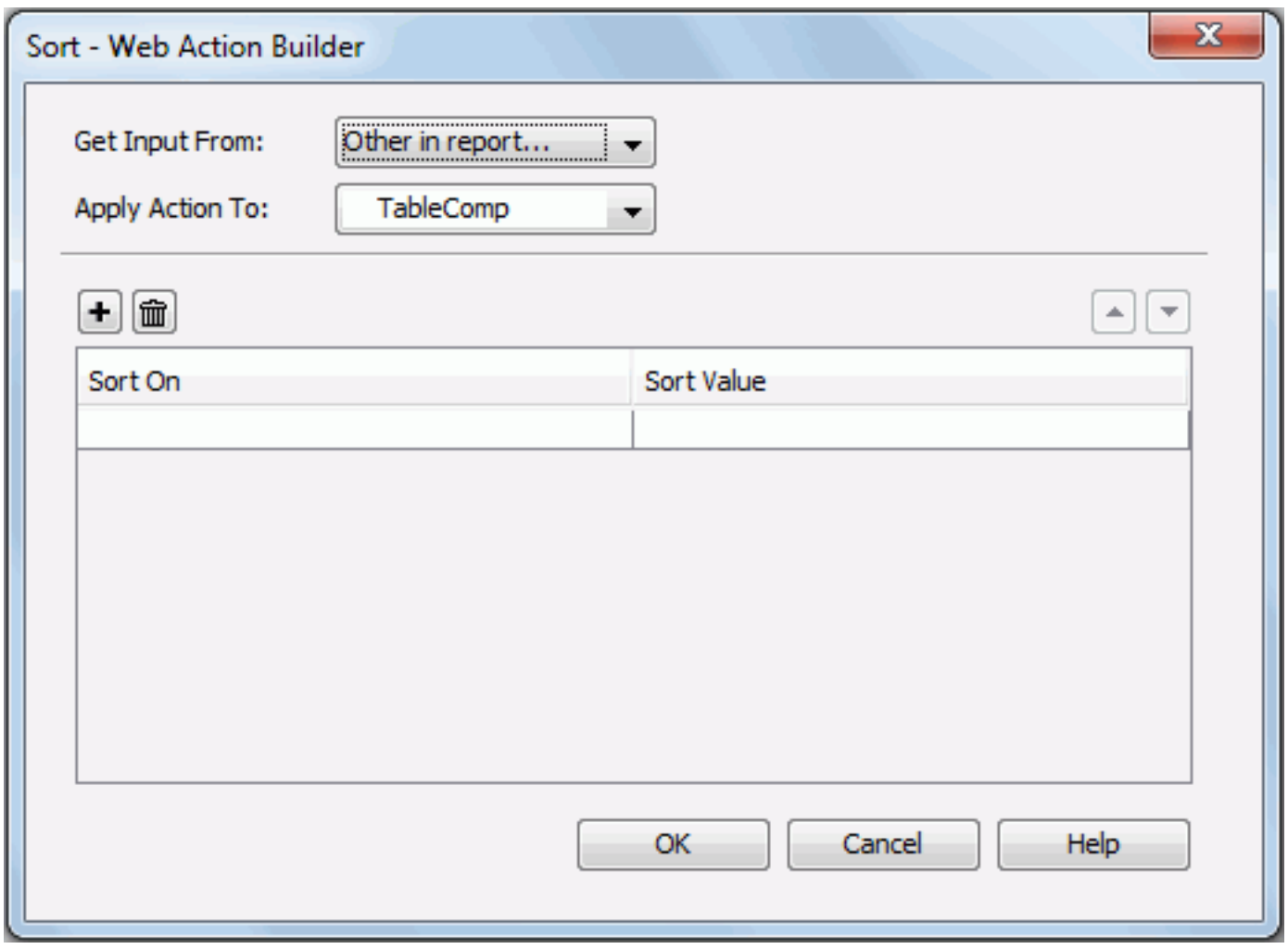
Displays the help document about this feature.

Related topics:

- Exporting report results - To PDF

Sort - Web Action Builder dialog

This dialog appears when you select *Sort and then click OK in the Web Action List dialog. It helps you to build a Sort web action on an object so as to sort the specified data components when the event bound with the action is triggered at runtime.



The following are details about options in the dialog:

Get Input From

Specifies where to get the object. You can select Form if you add some objects to a form. Otherwise, only the item Other in Report you can choose.

Apply Action To

Specifies to which component to apply the sort action.

Adds a new sort criterion for the action.

Removes the selected sort criterion.

Moves the selected sort criterion up a step.

Moves the selected sort criterion down a step.

Sort On

Specifies which field to be sorted on. Which fields are available here depends on which object you choose from the Apply Action To drop-down list.

Sort Value

Specifies the value of how to sort on the field. You can select one from the drop-down list. It could be Ascending or Descending.

OK

Accepts the changes and closes the dialog.

Cancel

Does not retain any changes and closes the dialog.

Help

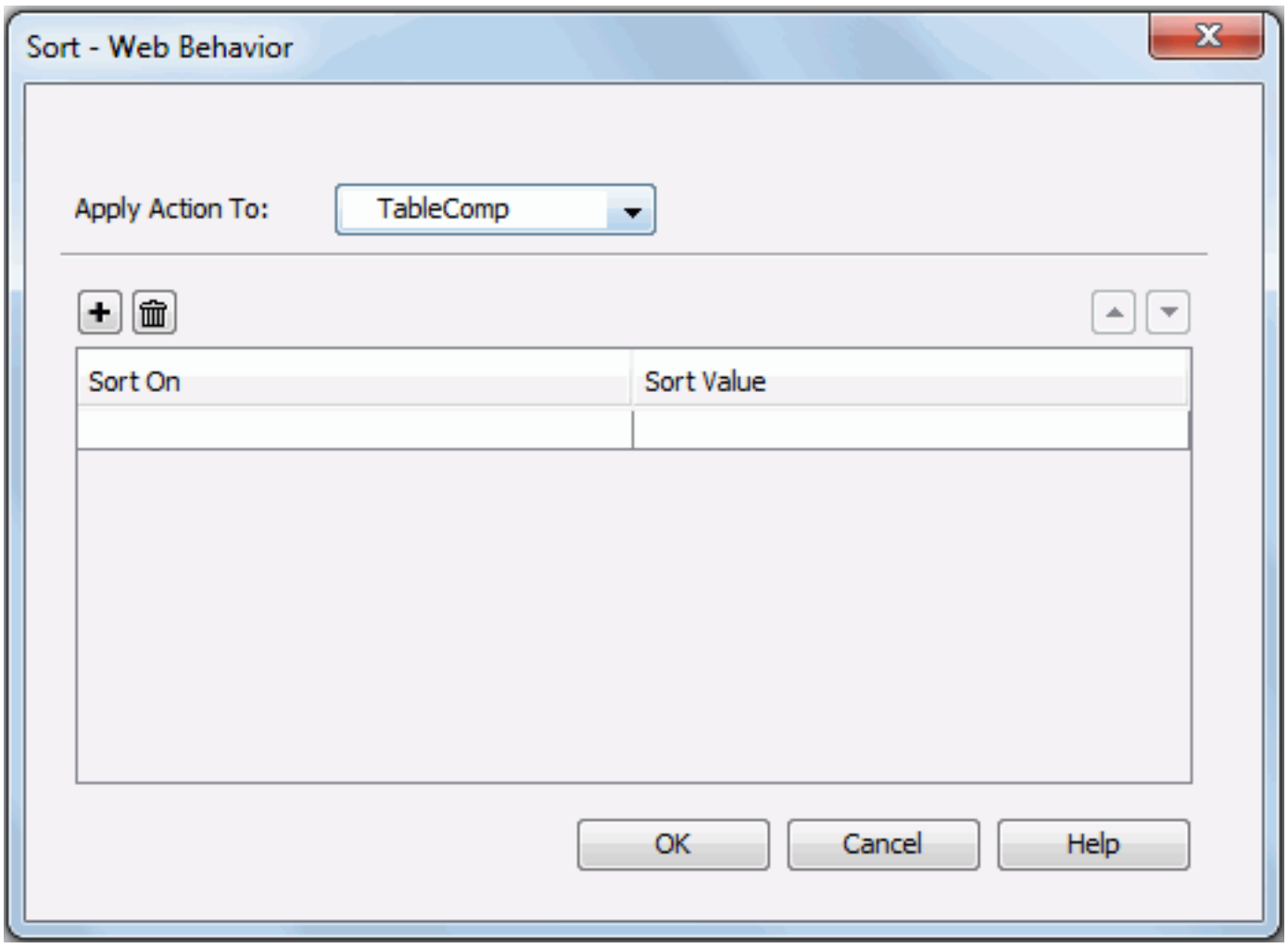
Displays the help document about this feature.

Related topics:

- Applying web actions to a label

Sort - Web Behavior dialog

This dialog appears when you right-click a web control in the configuration panel of a library component and select Web Behaviors > Sort from the shortcut menu. It helps you to build a Sort web behavior to the web control so as to sort data of the library component at runtime.



The following are details about options in the dialog:

Apply Action To

Specifies to which component to apply the sort action. Select the desired component from the drop-down list.

Adds a new sort criterion.

Removes the selected sort criterion.

Moves the selected sort criterion up a step.

Moves the selected sort criterion down a step.

Sort On

Specifies which field to be sorted on. Which fields are available here depends on which object you choose from the Apply Action To drop-down list. Select the field you want to sort on from the drop-down list.

Sort Value

Specifies the value of how to sort on the field. It could be Ascending or Descending, or be specified by the value of a web control.

OK

Accepts the changes and closes the dialog.

Cancel

Does not retain any changes and closes the dialog.

Help

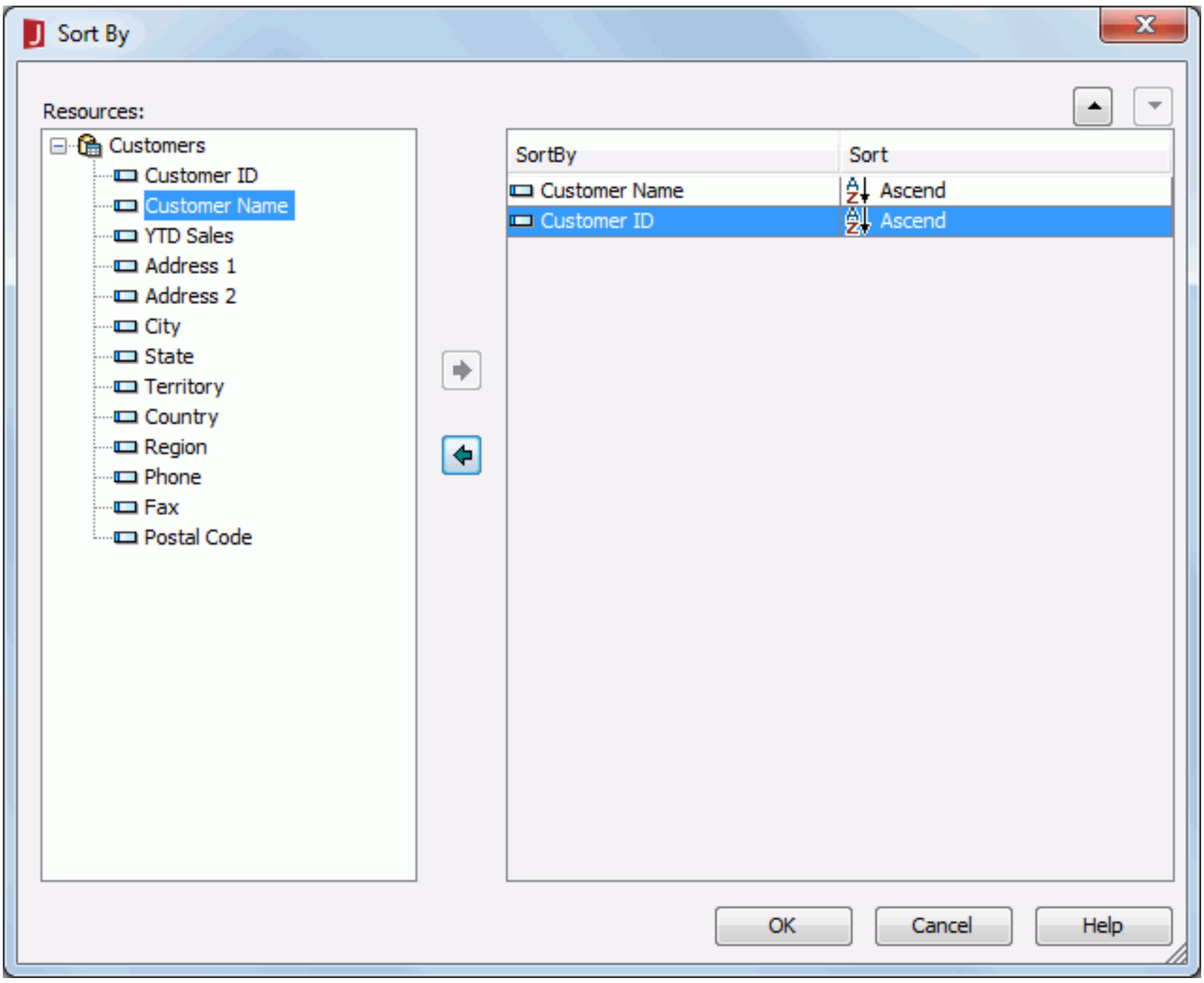
Displays the help document about this feature.

Related topics:

- Applying web behaviors to web controls in the panel

Sort By dialog

The dialog appears when you select Sort By as value of the Sort option in the Options table of the Create Parameter dialog or Edit Parameter dialog. It helps you to sort values of cascading parameters by sorting other fields in the same data source as the cascading group.



Resources

Lists the fields in the same data source as the cascading group.

Adds the selected field to be sorted.

Removes the selected field from being sorted.

Moves the selected field one level up.

Moves the selected field one level down.

Sort By

Lists the added fields to be sorted. The order of the fields determines the order of being sorted. The first field will be sorted first, and the last field will be sorted last.

Sort

Sorts the added fields one by one.

- **Ascend**
  Sorts the current field in an ascending order.
- **Descend**
  Sorts the current field in a descending order.

OK

Applies the sorting and closes the dialog.

Cancel

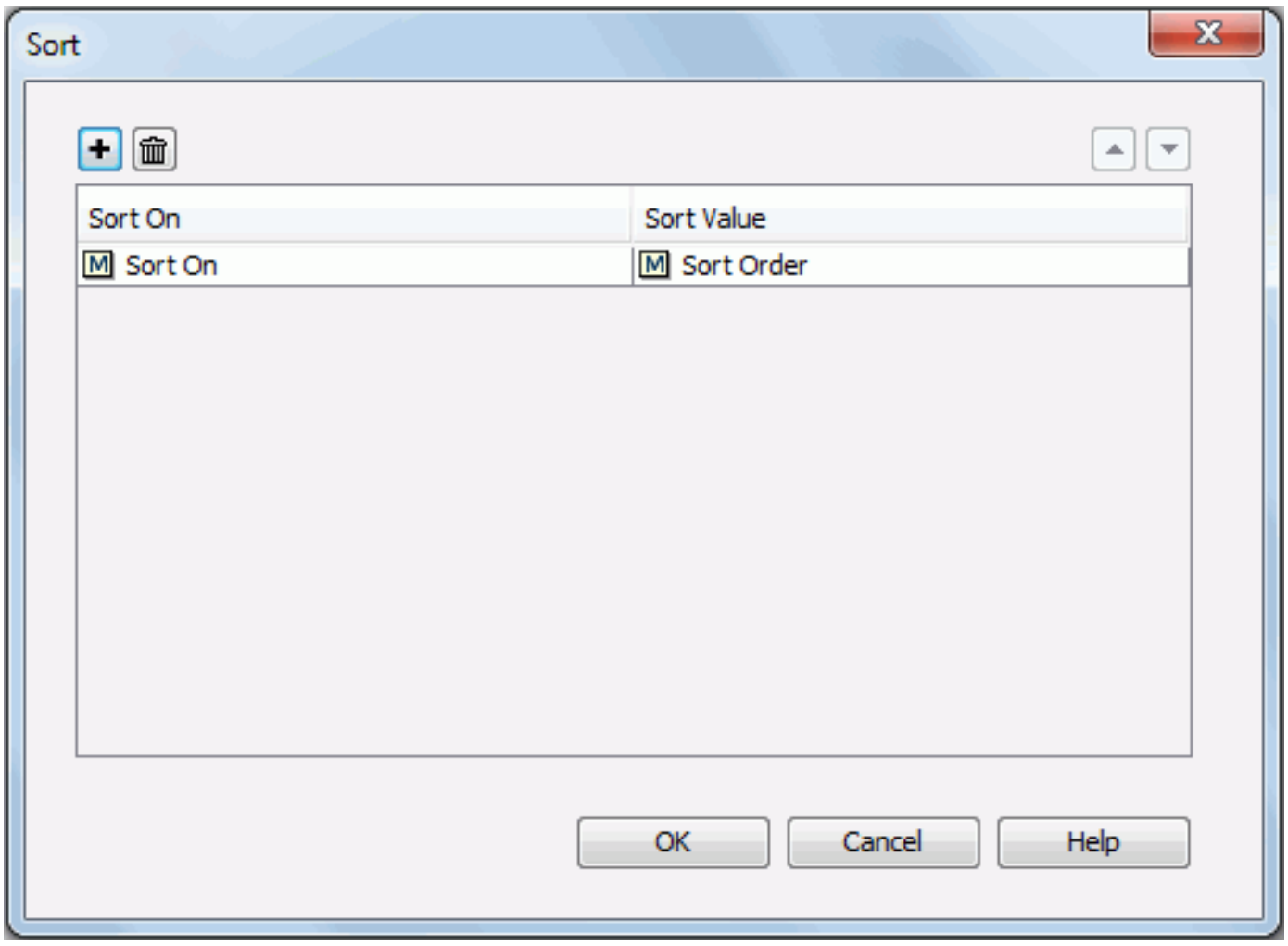
Cancels the sorting and closes the dialog.

Help

Displays the help document about this feature.

Sort dialog

This dialog appears when you select 0002 - Sort from the drop-down list of the Message ID column and click ... in the Actions column of the Receive Message dialog, or when you select *Sort and then click OK in the Web Action List dialog. It helps you to define to sort data of the current library component as a response to the message the library component receives at runtime.



The following are details about options in the dialog:

Adds a new sort criterion.

Removes the selected sort criterion.

Moves the selected sort criterion up a step.

Moves the selected sort criterion down a step.

Sort On

Specifies which column to be sorted on.

Sort Value

Specifies the value of how to sort on the column.

OK

Accepts the changes and closes the dialog.

Cancel

Does not retain any changes and closes the dialog.

Help

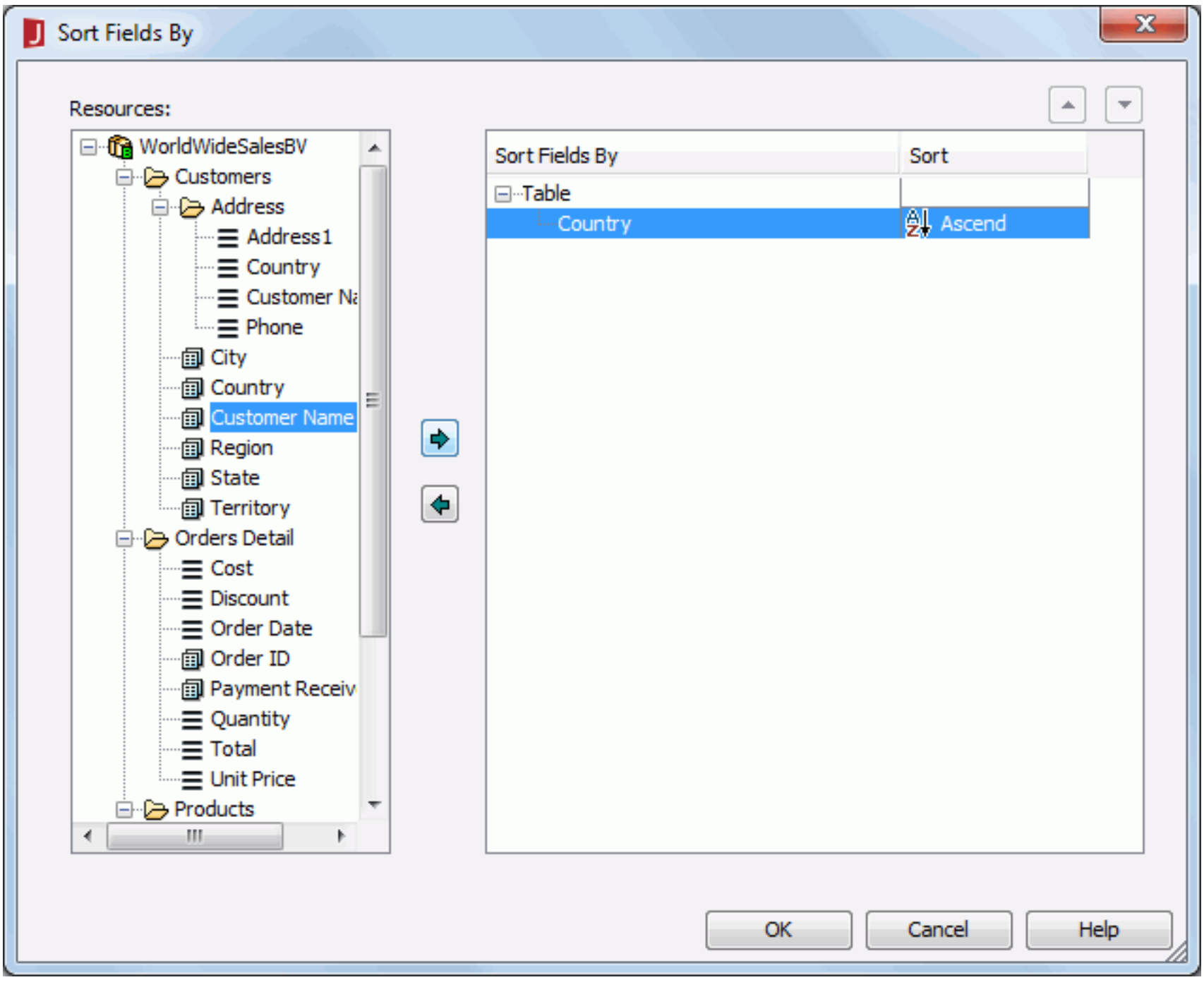
Displays the help document about this feature.

Related topics:

- Receiving messages

Sort Fields By dialog

The dialog appears when you click the Sort Fields By button in the Display screen of the report wizard. It helps you to set the sorting manner of data displayed in the banded object/table.



The following are the details of the dialog:

Resources

Lists all the available data resources.

Sort Fields By

Displays the fields on which the sorting will be based.

Sort

Specifies the sorting manner.

- **Ascend**
  If selected, data in the banded object/table will be sorted based on values of the specified sort by field in ascending order.
- **Descend**
  If selected, data in the banded object/table will be sorted based on values of the specified sort by field in descending order.

Adds the selected field as the sort by field.

Removes the specified sort by field if it is not required.

Moves the selected sort by field one step up.

Moves the selected sort by field one step down.

OK

Applies the changes and closes the dialog.

Cancel

Cancels the changes and exits the dialog.

Help

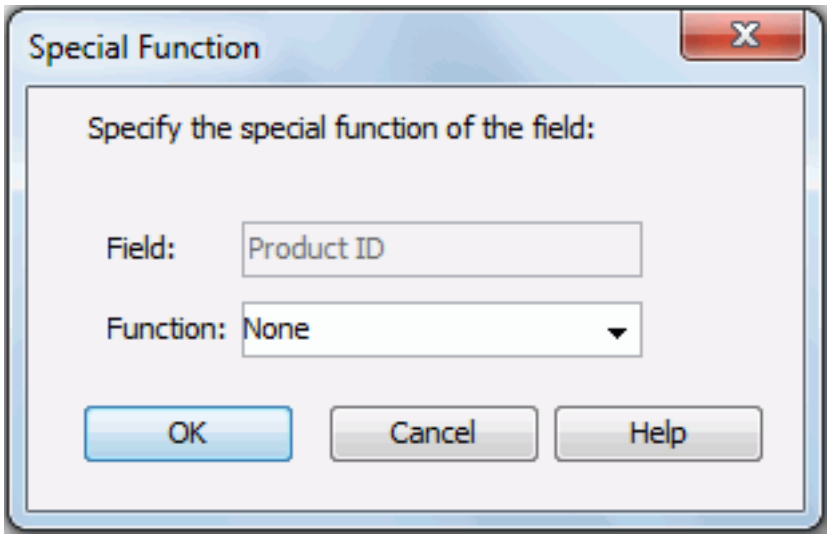
Displays the help document about this feature.

Related topics:

- Sorting the data

Special Function dialog

This dialog appears when you click the Special Function button in the Display screen of the chart wizard. It helps you to specify a special function to the field displayed on the category/series axis of a chart, which is of Numeric/String/Date/Time data type.



Field

Displays on which field the special function will be applied.

Function

Specifies the special function to the field. For details about the special functions, refer to Specifying special function for group by field.

OK

Accepts the changes and closes the dialog.

Cancel

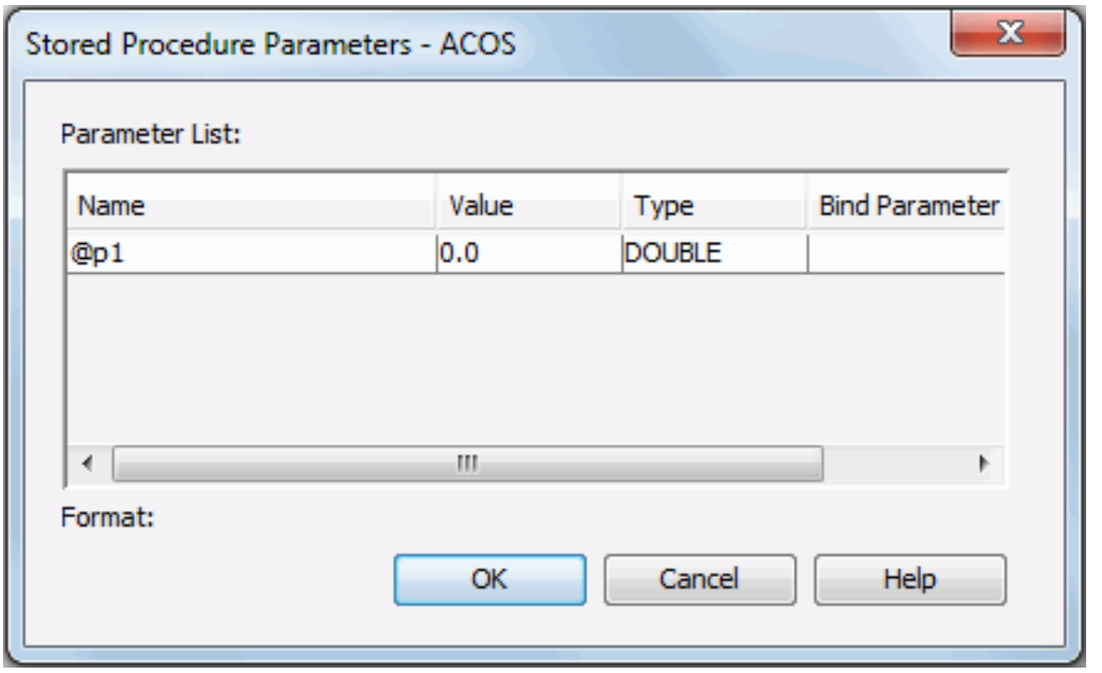
Discards the changes and closes the dialog.

Help

Displays the help document about this feature.

Stored Procedure Parameters dialog

This dialog appears when you right-click a stored procedure in the Catalog Browser and select Parameters from the shortcut menu. It helps you to edit parameter values for the specified stored procedure.



The following are details about options in the dialog:

Parameter List

Lists the IN and INOUT parameters in the stored procedure.

- **Name**
  Displays names of parameters of the stored procedure.
- **Value**
  Specifies the value of each parameter. Double-click the cell to edit the value.
- **Type**
  Displays types of parameters of the stored procedure.
- **Bind Parameter Name**
  Allows you to bind the IN and INOUT parameters in the stored procedure to an existing catalog parameter or constant level formula of the same type or the special field User Name. Double-click the cell to input the parameter name (@parametername), or the formula name (@formulaname), or the special field User Name (username). By default, a bound parameter will have the same name of the IN and INOUT parameter that is automatically created when adding the stored procedure.

Format

Displays the value format of the current parameter.

OK

Applies all changes here and closes the dialog.

Cancel

Cancels the settings you make and closes the dialog.

Help

Displays the help document about this feature.

Related topics:

- Stored procedures

Subreport dialog

This dialog appears when you click Insert > Subreport on the menu bar, the Insert Subreport button on the Insert toolbar, or right-click a subreport and click Format Subreport from the shortcut menu. It helps you to insert a report into another report as its subreport, or to format an exsiting subreport.

The following are details about options in the dialog:

Select Report

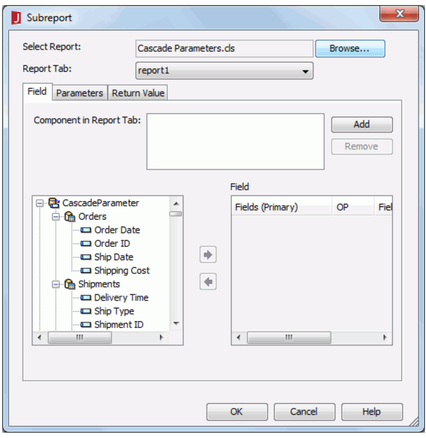
Specifies the report that contains the report tab you want to use as the subreport. Click the Browse button to select the desired report.

Report Tab

Specifies the report tab as the subreport.

Field tab

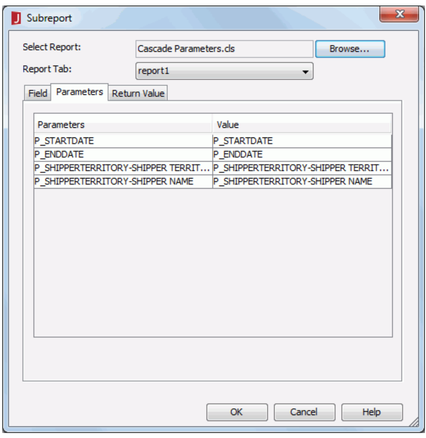
Specifies some links between the primary report and subreport.



- **Component in Report Tab**
  Lists the components in the subreport that will be interlinked with the primary report.
  - **Add**
    Adds a component in the subreport to be interlinked with the primary report.
  - **Remove**
    Removes the selected component from the Component in Report Tab box.
- Adds a field of the primary report to set up links between the primary report and the subreport as required.
- Removes the selected field.
- **Field**
  Lists all the fields that are used in the link conditions.
  - **Fields(Primary)**
    Lists the selected fields of the primary report.
  - **OP**
    Specifies the operator to be the condition for setting up links between the primary report and the subreport.
  - **Fields(Subreport)**
    Lists the DBFields in the datasets of the subreport which are of the same data type as the selected fields in the primary report.

Parameters tab

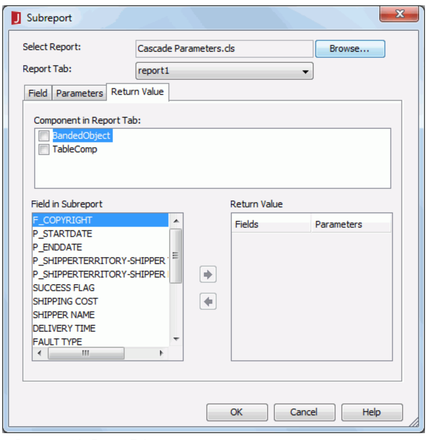
Available only when there is at least a parameter in the subreport. It helps you to specify values for all parameters in the subreport.



- **Parameters**
  Lists all the parameters contained in the datasets of the subreport.
- **Value**
  Lists all the DBFields, formulas, summaries and parameters in the dataset of the primary report which are of the same data type as the parameters of the subreport. Select the field the value of which you want to assign to parameter of the subreport from the drop-down list.

Return Value tab

Specifies values of the subreport fields to be returned as the primary report parameter values.



- **Component in Report Tab**
  Specifies the component in the subreport the fields of which will be used to return value to parameters of the primary report.
- **Fields in Subreport**
  Lists all available fields, formulas, and parameters used by the specified component in the subreport.
- Adds a field that you want to assign to a parameter in the primary report.
- Removes the selected field.
- **Return Value**
  Lists all the fields in the specified component of the subreport that will be used to return value to parameters of the primary report.
  - **Fields**
    Lists all the selected fields from the specified component in the subreport.
  - **Parameters**
    Lists all the parameters in the primary report which are of the same data type as the fields selected in the specified component of the subreport. Select the parameter to which the subreport field value will be returned from the drop-down list.

OK

Applies all changes and closes the dialog.

Cancel

Does not retain any changes and closes the dialog.

Help

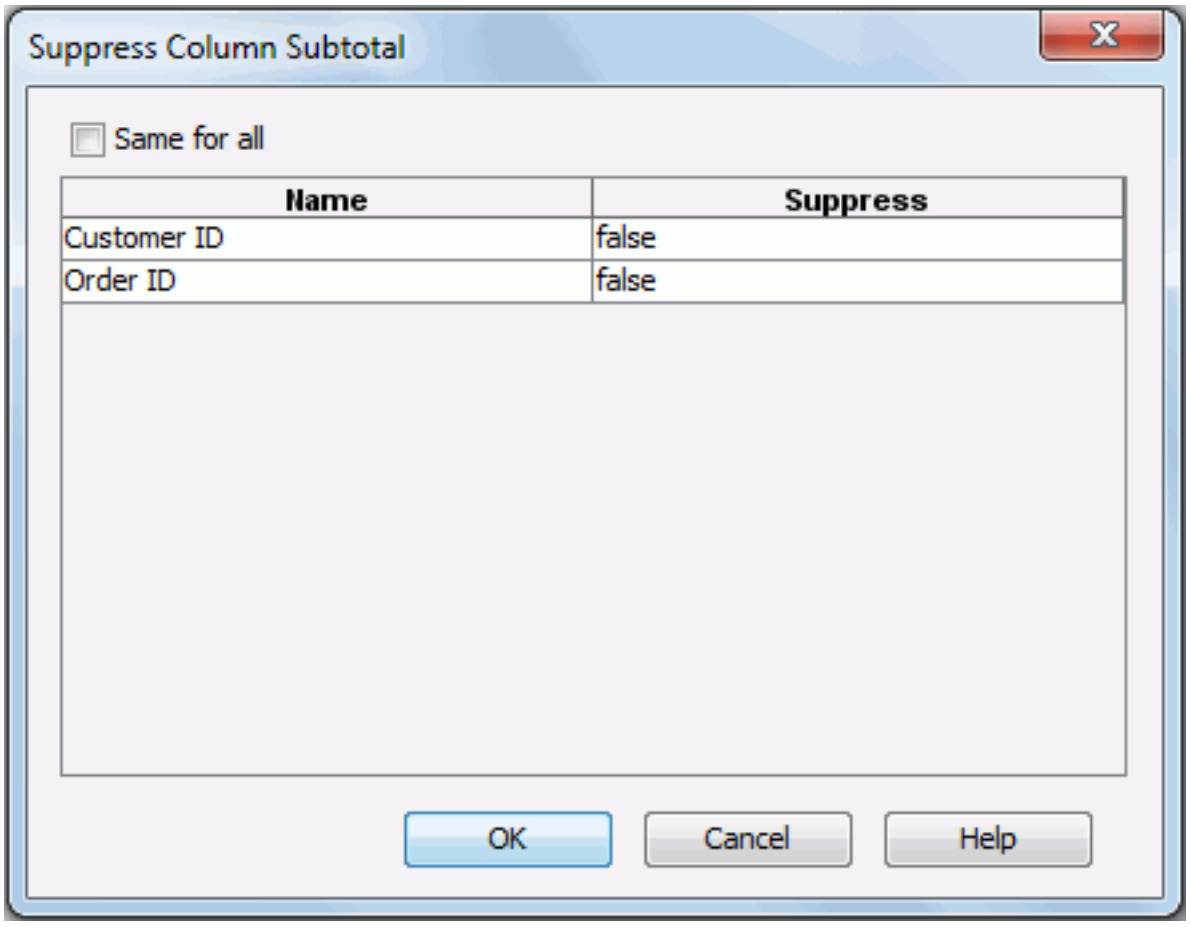
Displays the help document about this feature.

Related topics:

- Subreports

Suppress Column Subtotal dialog

The dialog appears when you click ... next to the Suppress Column Subtotal option in the Layout screen of the crosstab wizard. It helps you to control which subtotals of the columns in the crosstab will be suppressed, and which will be shown.



Same for all

If checked, the setting for the first subtotal will apply to all the subtotals.

Name

Lists the name of all subtotals of columns.

Suppress

Specifies whether to suppress the selected subtotal by selecting true or false from the drop-down list. It could also be controlled by using a constant level formula.

OK

Applies the changes and closes the dialog.

Cancel

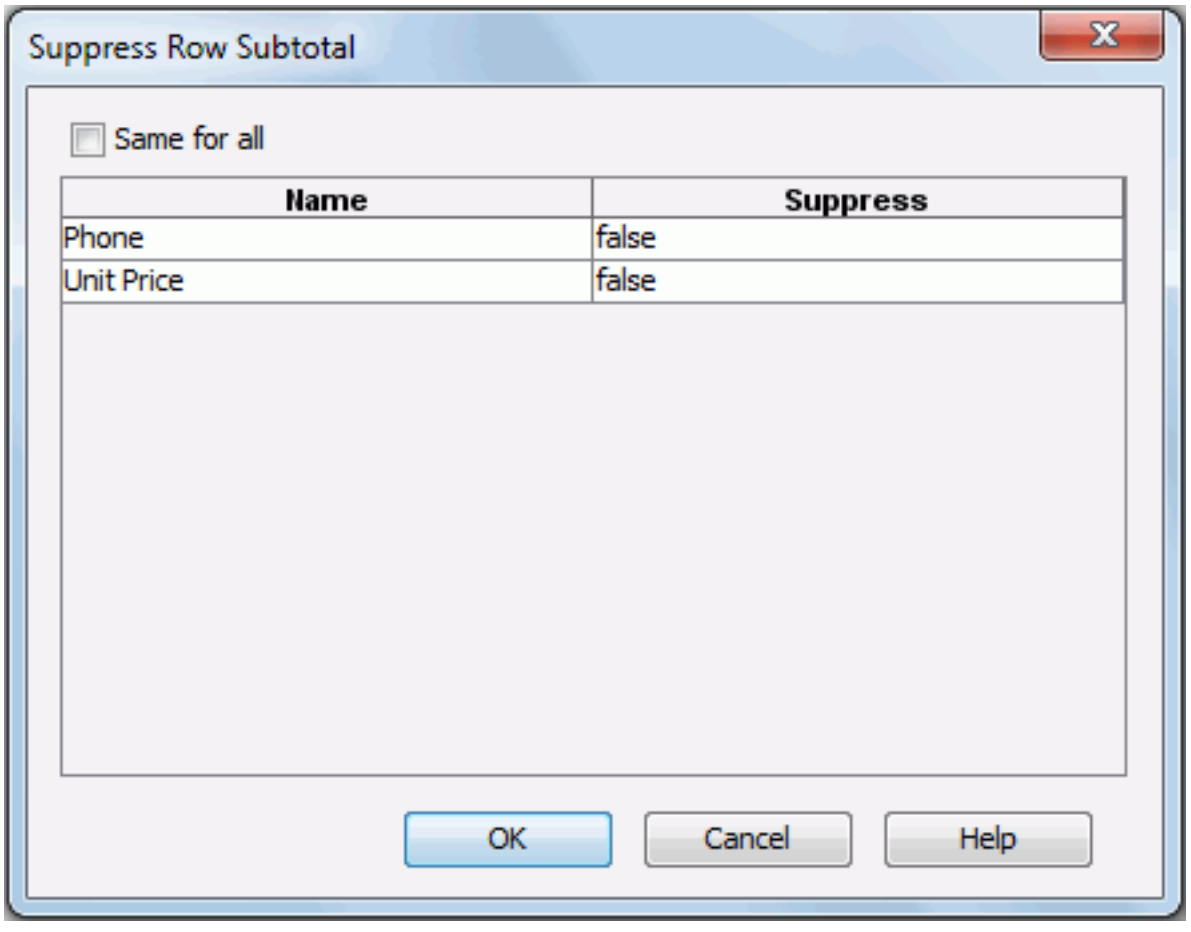
Does not retain any changes and closes the dialog.

Help

Displays the help document about this feature.

Suppress Row Subtotal dialog

The dialog appears when you click ... next to the Suppress Row Subtotal option in the Layout screen of the crosstab wizard. It helps you to control which subtotals of the rows in the crosstab will be suppressed, and which will be shown.



Same for all

If checked, the setting for the first subtotal will apply to all the subtotals.

Name

Lists the name of all subtotals of rows

Suppress

Specifies whether to suppress the selected subtotal by selecting true or false from the drop-down list. It could also be controlled by using a constant level formula.

OK

Applies the changes and closes the dialog.

Cancel

Does not retain any changes and closes the dialog.

Help

Displays the help document about this feature.

Table Type dialog

This dialog appears when you click Insert > Table on the menu bar, click the Insert Table button on the Insert toolbar, or drag Table/Horizontal Table from the Toolbox panel to a report or library component. It helps you to choose a type for the table you want to create.

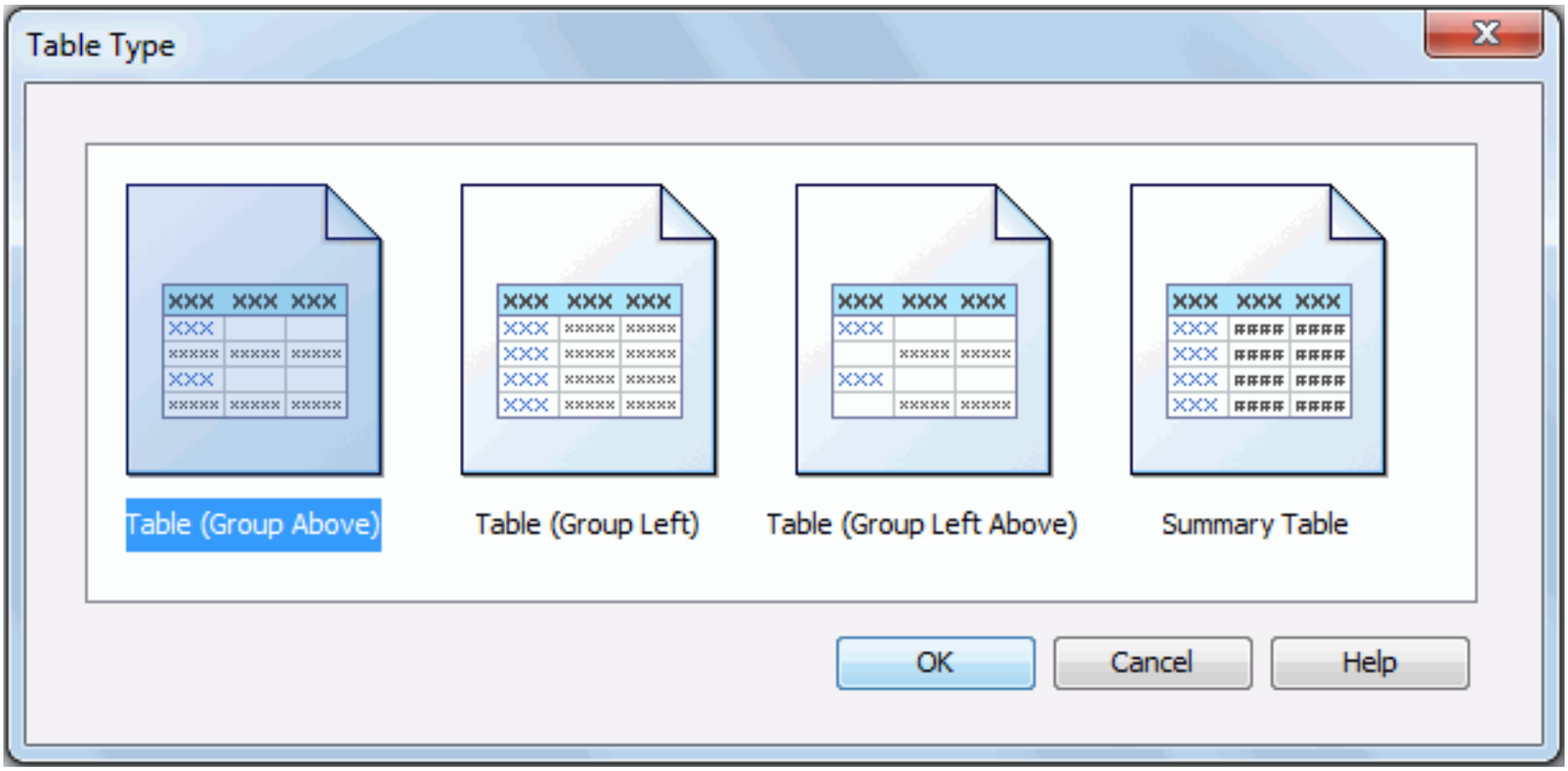


Table (Group Above)

Creates a table with group information above the detail panel.

Table (Group Left)

Creates a table with group information left to the detail panel.

Table (Group Left Above)

Creates a table with group information left above the detail panel.

Summary Table

Creates a table with only group and summary information.

OK

Applies the changes and opens the Create Table wizard for page report or Create Table wizard for web report/library component.

Cancel

Does not retain any changes and closes the dialog.

Help

Displays the help document about this feature.

Table Wizard for page report

This wizard appears when you select a table layout in the New Page Report Tab dialog or New Page Report dialog and then click OK, or right-click a table and click Table Wizard on the shortcut menu. It helps you to create or modify a table in a page report.

The wizard consists of the following screens:

- Data screen
- Display screen
- Group screen
- Summary screen
- Chart screen
- Filter screen
- Style screen

Back

Goes back to the previous screen.

Next

Goes to the next screen.

Finish

Finishes creating or modifying the table and closes this wizard.

Cancel

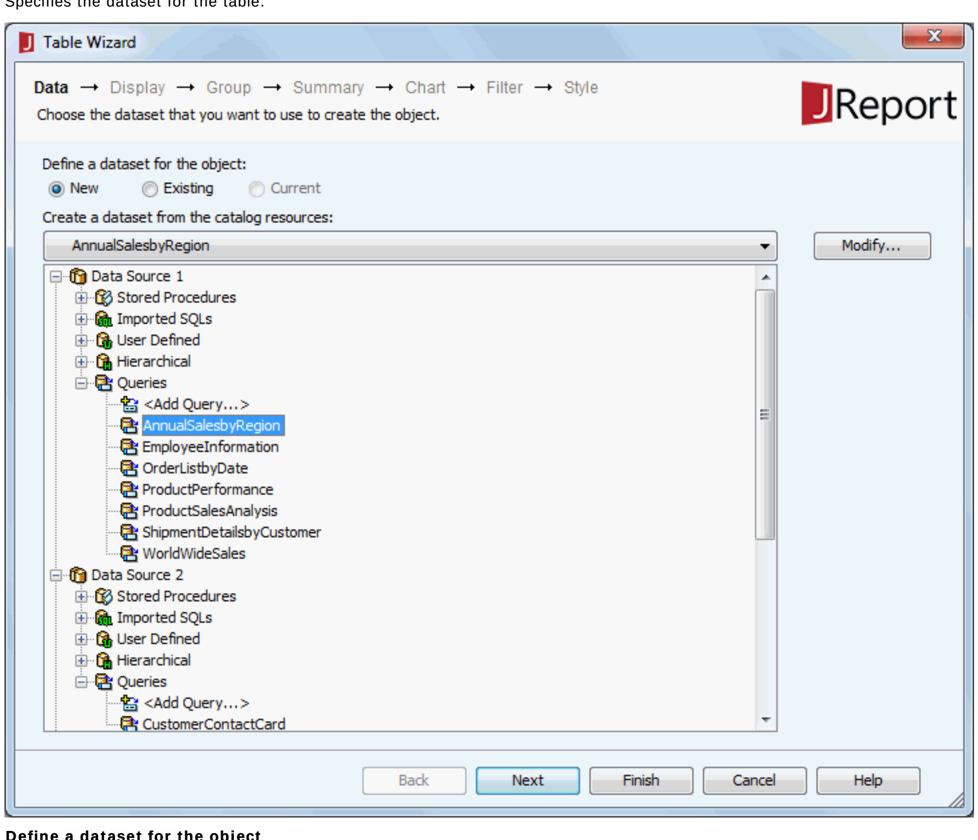
Does not retain changes and closes this wizard.

Help

Displays the help document about this feature.

Data screen

Specifies the dataset for the table.



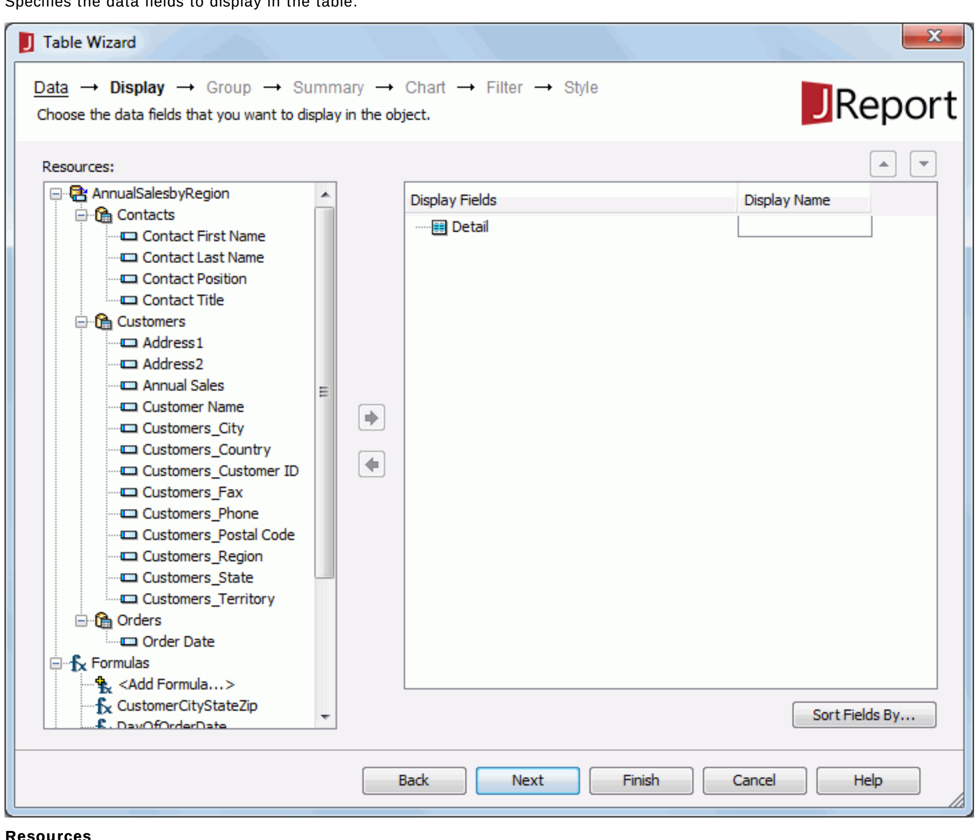
Define a dataset for the object

Specifies the dataset you want to use to create or edit the table.

- **New**
  If checked, select a data source from the catalog resources to create a dataset that will be used to build the table. When you choose to create the dataset from a query, you can click the Modify button to edit the query in the Query Editor if required.
- **Existing**
  If checked, select a dataset from the ones existing in the open report to create the table. Click the Modify button to edit the dataset in the Dataset Editor if required.
- **Current**
  If checked, the current dataset used by the parent object will be applied to the table.

Display screen

Specifies the data fields to display in the table.



Resources

Lists all the available data resources.

Adds the specified field to use in the table.

Removes the specified field that is not required in the table.

Replaces the selected field in the table with the specified field in the Resources box.

Moves the specified field one step up.

Moves the specified field one step down.

Show Table Group Structure

Specifies whether to show the group structure of the table.

Display Fields

Lists the fields that you have selected to display in the table.

Display Name

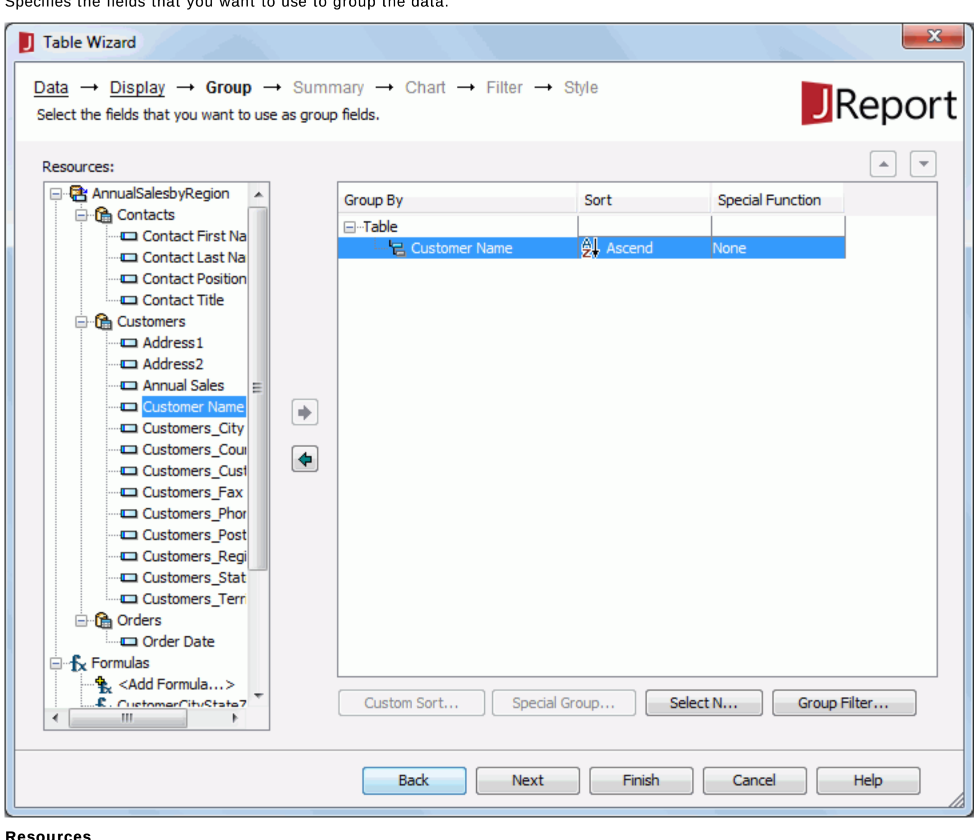
Shows the display names of the selected fields. You can double-click the name cells to edit them if required.

Sort Fields By

Opens the Sort Fields By dialog to specify how to sort data in the table.

Group screen

Specifies the fields that you want to use to group the data.



Resources

Lists all the available data resources.

Adds the selected field as the group by field in the table.

Removes the selected group by field that is not required.

Replaces the selected group by field in the table with the specified field in the Resources box.

Moves the specified group one step up.

Moves the specified group one step down.

Group By

Lists the fields that are used to group data in the table.

Sort

Specifies how groups at the specific group level will be sorted.

- **Ascend**
  Groups will be sorted in an ascending order (A, B, C).
- **Descend**
  Groups will be sorted in a descending order (C, B, A).
- **No Sort**
  Groups will be sorted in the original order in database.
- **Special Group**
  Opens the User Defined Group dialog to define grouping information.

  For example, if you placed a field named Region for grouping, and this field contains all 50 states of the United States; and if you want to see the data between Maryland (MD) and New York (NY), you can define the criteria by selecting the between operator to further define your grouping information.
- **Custom Sort**
  Opens the Custom Sort dialog to set how groups will be sorted.

Special Function

If the group by field is of Numeric/String/Date/Time type, you can select a special function for the field in the Special Function column to further specify to which level the data will be grouped by.

If Customize is selected, the Customized Function dialog will be displayed, in which you can set the function by your own.

Custom Sort

Specifies how to sort the groups. Activated only when you have clicked Custom Sort from the Sort column to define the sorting manner of groups for the selected group level.

Special Group

Specifies how to group your information. Activated only when you have clicked Special Group from the Sort column to define a special group.

Select N

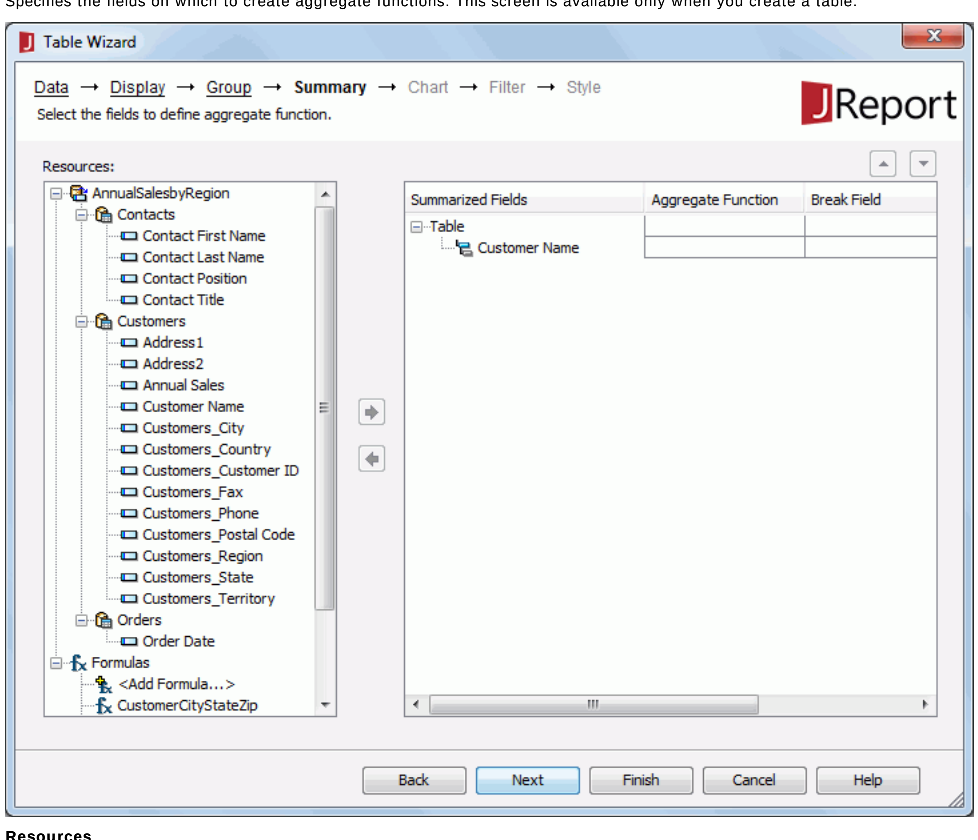
Opens the Select N dialog to specify the Select N condition.

Group Filter

Opens the Group Filter dialog to specify the group filter condition.

Summary screen

Specifies the fields on which to create aggregate functions. This screen is available only when you create a table.



Resources

Lists all the available data resources.

Adds the selected field as the summary field to the table.

Removes the selected summary field that is not required.

Moves the specified summary field one step up.

Moves the specified summary field one step down.

Summarized Fields

Lists all the fields that you want to display and to be summarized in the table.

Aggregate Function

Specifies the function to summarize the selected field.

Break Field

Displays a defined field on which the summary will be calculated. If a summary field is added below the Table node, the break field is null and the summary will be calculated on the whole dataset.

Position

Works together with the Column option to specify the location where the summary field will be put.

- **Header**
  Specifies to put the summary field in the header row. If the summary is calculated on a group by field, it will be put in the group header row of the corresponding group; if the summary is calculated on the whole dataset, it will be put in the table header row.
- **Footer**
  Specifies to put the summary field in the footer row. If the summary is calculated on a group by field, it will be put in the group footer row of the corresponding group; if the summary is calculated on the whole dataset, it will be put in the table footer row.

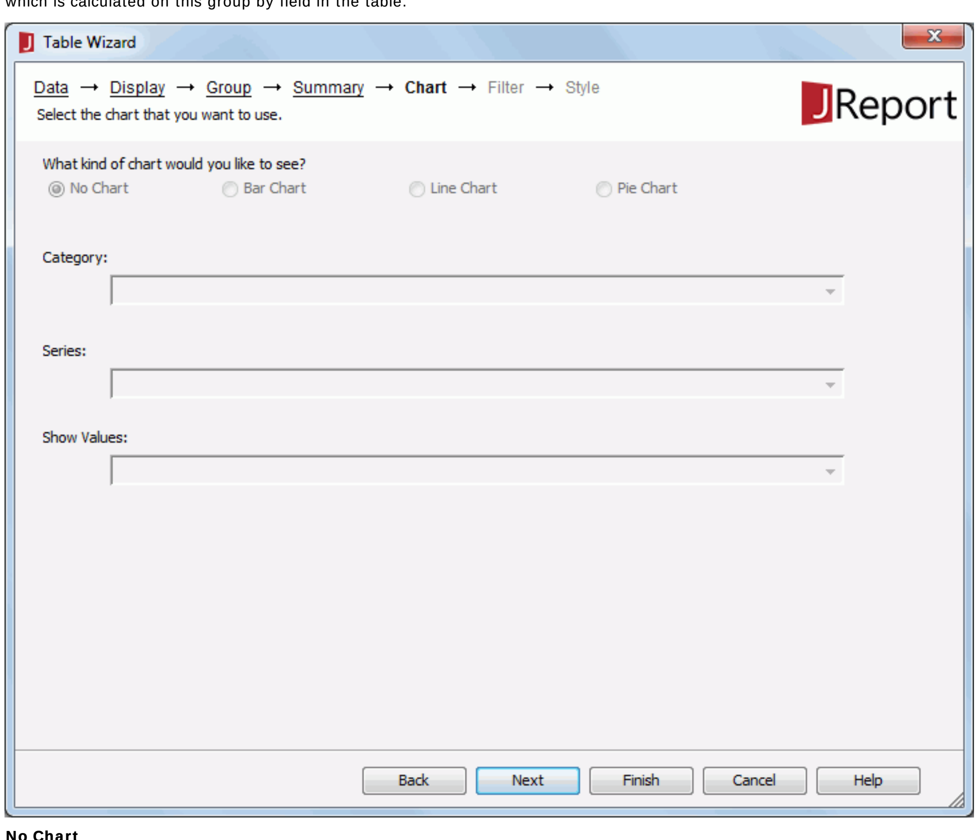
Column

Works together with the Position option to specify the location where the summary field will be put.

- **Detail**
  The summary field will be put in a detail column.
- **Summary**
  The summary field will be displayed in a separate summary column.

Chart screen

Specifies to create a chart together with the table, which will be placed above the table in the report body. This screen is available only when you create a table in a page report, and when there is at least one group by field and one summary field which is calculated on this group by field in the table.



No Chart

Specifies not to create a chart.

Bar Chart

Specifies to create a Clustered Bar 2-D chart together with the table.

Line Chart

Specifies to create a Line 2-D chart together with the table.

Pie Chart

Specifies to create a Clustered Pie chart together with the table.

Category

Lists the group by fields of the table on which the summary fields are calculated. Choose the field you want to display on the category (X) axis of the chart from the drop-down list.

Series

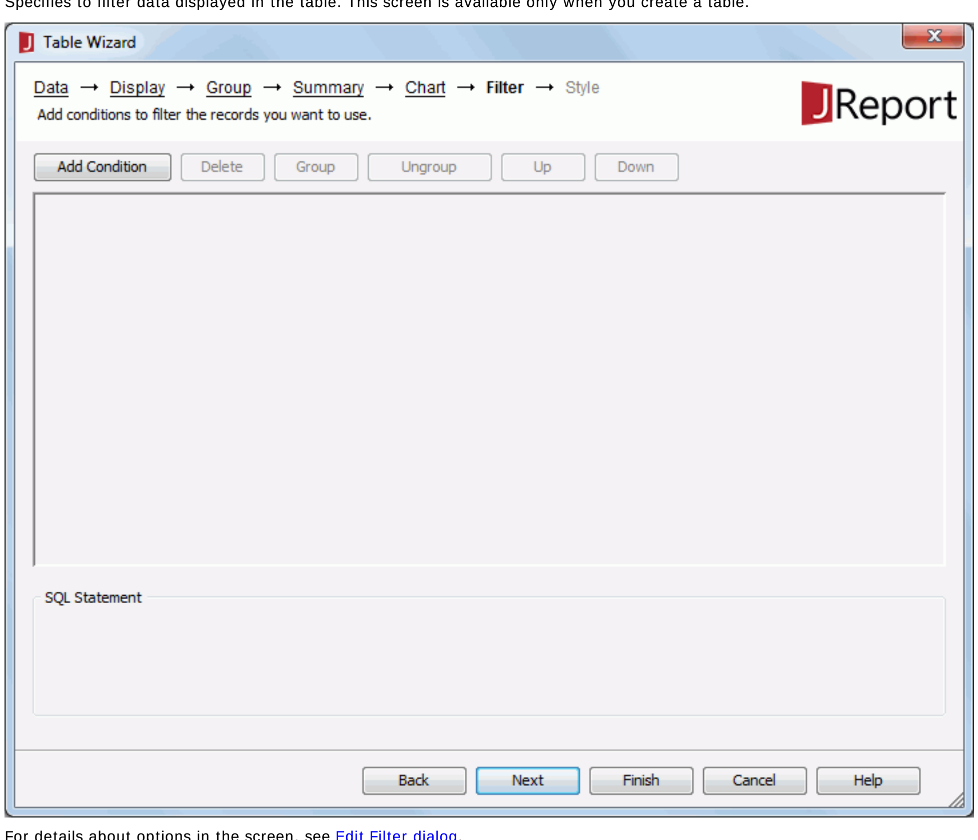
Lists the fields that have been added as the group by fields of the table. Choose the field you want to display on the series (Z) axis of the chart from the drop-down list.

Show Values

Lists the summary fields which are calculated based on the field you choose to display on the category axis of the chart. Choose the value you want to display in the chart from the drop-down list.

Filter screen

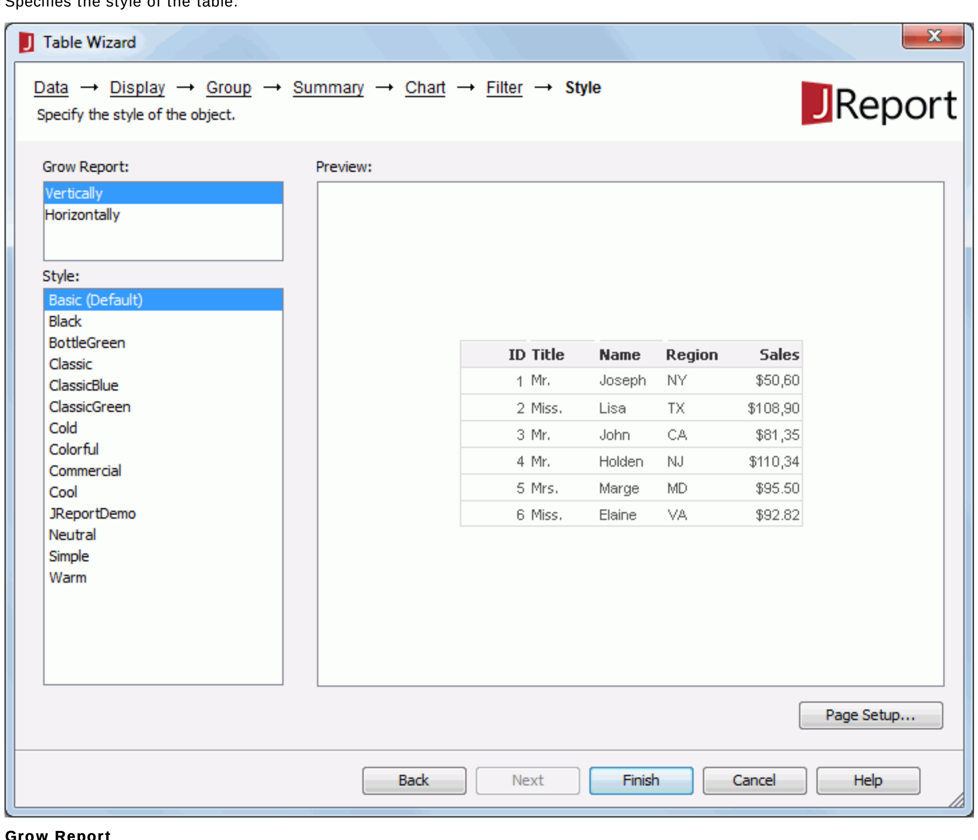
Specifies to filter data displayed in the table. This screen is available only when you create a table.



For details about options in the screen, see Edit Filter dialog.

Style screen

Specifies the style of the table.



Grow Report

Specifies the layout of the table.

- **Vertically**
  Creates a vertical table.
- **Horizontally**
  Creates a horizontal table.

Style

Lists the available styles.

- **<Custom>**
  There is no any style information on it and it is only used to cover previous-version report which did not bind any style or the bound style cannot be found in the style list.

Preview

Displays a diagram illustrating the effect of the selected style on the table.

Inherit Style

Specifies whether to make the table take the style of its parent. This option is available only when you modify a table and the table is inserted into a banded object.

Page Setup

Opens the Page Setup dialog to specify the page settings. Available only when creating a page report.

Related topics:

- Creating a table report
- Modifying a table

Table Wizard for web report/library component

This wizard appears when you select a table type in the component type box and click OK in the New Library Component dialog, or right-click a table and click Table Wizard on the shortcut menu. It helps you to create or modify a table in a web report or library component.

The wizard consists of the following screens:

- Data screen
- Display screen
- Group screen
- Summary screen
- Filter screen
- Style screen

Back
Goes back to the previous screen.

Next
Goes to the next screen.

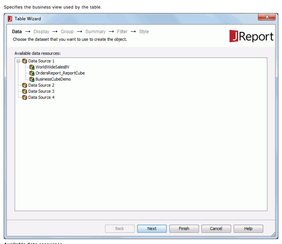
Finish
Finishes creating or modifying the table and closes this wizard.

Cancel
Does not retain changes and closes this wizard.

Help
Displays the help document about this feature.

Data screen

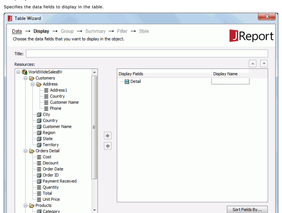
Specifies the business view used by the table.



Available data resources
Lists all the business views in the current catalog. Specify the one you want to use.

Display screen

Specifies the data fields to display in the table.



Title
Specifies the title of the table.

Resources
Lists all the available data resources. You can also create dynamic formulas to use in the table.

Adds the specified field to use in the table.

Removes the specified field that is not required in the table.

Replaces the selected field in the table with the specified field in the Resources box.

Moves the specified field one step up.

Moves the specified field one step down.

Show Table Group Structure
Specifies whether to show the group structure of the table.

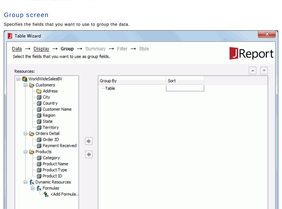
Display Fields
Lists the fields that you have selected to display in the table.

Display Name
Shows the display names of the selected fields. You can double-click the name cells to edit them if required.

Sort Fields By
Opens the Sort Fields By dialog to specify how to sort data in the table.

Group screen

Specifies the fields that you want to use to group the data.



Resources
Lists all the available data resources. You can also create dynamic formulas to use in the table.

Adds the selected field as the group by field in the table.

Removes the selected group by field that is not required.

Replaces the selected group by field in the table with the specified field in the Resources box.

Moves the specified group one step up.

Moves the specified group one step down.

Group By
Lists the fields that are used to group data in the table.

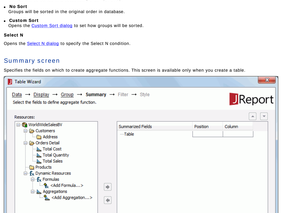
Sort
Specifies how groups at the specific group level will be sorted.

- **Ascend**
  Groups will be sorted in an ascending order (A, B, C).
- **Descend**
  Groups will be sorted in a descending order (C, B, A).
- **No Sort**
  Groups will be sorted in the original order in database.
- **Custom Sort**
  Opens the Custom Sort dialog to set how groups will be sorted.

Select N
Opens the Select N dialog to specify the Select N condition.

Summary screen

Specifies the fields on which to create aggregate functions. This screen is available only when you create a table.



Resources
Lists all the available data resources. You can also create dynamic formulas and aggregations to use in the table.

Adds the selected field as the summary field to the table.

Removes the selected summary field that is not required.

Moves the specified summary field one step up.

Moves the specified summary field one step down.

Summarized Fields
Lists all the fields that you want to display and to be summarized in the table.

Position
Works together with the Column option to specify the location where the summary field will be put.

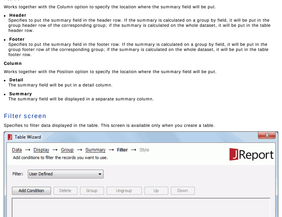
- **Header**
  Specifies to put the summary field in the header row. If the summary is calculated on a group by field, it will be put in the group header row of the corresponding group; if the summary is calculated on the whole dataset, it will be put in the table header row.
- **Footer**
  Specifies to put the summary field in the footer row. If the summary is calculated on a group by field, it will be put in the group footer row of the corresponding group; if the summary is calculated on the whole dataset, it will be put in the table footer row.

Column
Works together with the Position option to specify the location where the summary field will be put.

- **Detail**
  The summary field will be put in a detail column.
- **Summary**
  The summary field will be displayed in a separate summary column.

Filter screen

Specifies to filter data displayed in the table. This screen is available only when you create a table.



Filter
It lists all the predefined filters of the specified business view. Select one from the drop-down list to apply, or select User Defined and specify define a new filter according to your requirements.

For details about options in the screen, see Edit Filter dialog.

Style screen

Specifies the style of the table.

Grow Report
Specifies the layout of the table.

- **Vertically**
  Creates a vertical table.

Style
Lists the available styles.

- **<Custom>**
  There is no any style information on it and it is only used to cover previous version report which did not bind any style or the bound style cannot be found in the style list.

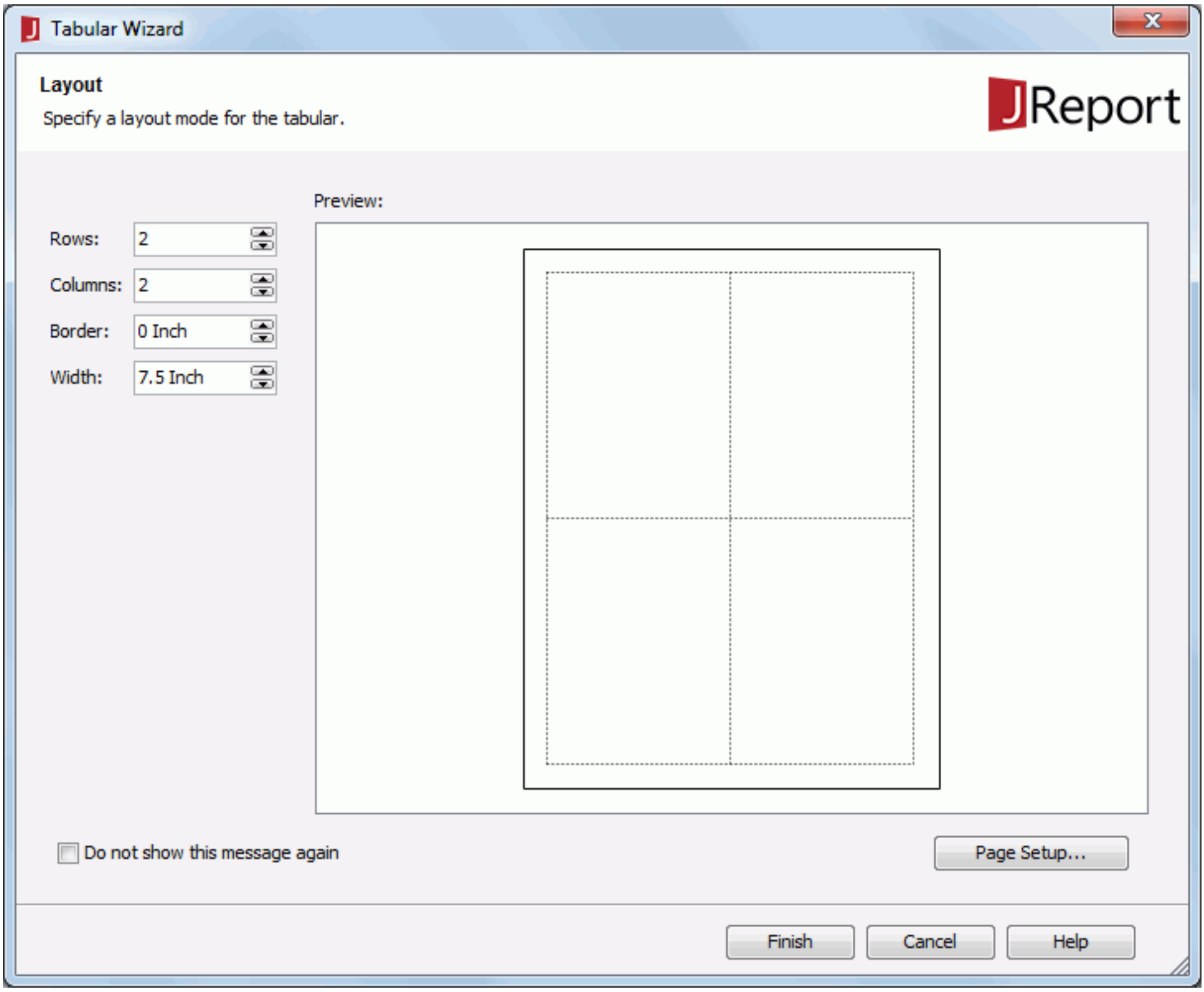
Preview
Displays a diagram illustrating the effect of the selected style on the table.

Related topics:

- Creating a library component
- Modifying a table

Tabular Wizard

This wizard appears when you select the Tabular Report layout in the New Page Report Tab dialog or New Page Report dialog and then click OK. It helps you to create a report with a tabular in it.



Rows

Specifies the number of rows for the tabular.

Columns

Specifies the number of the columns for the tabular.

Width

Specifies the width for the tabular in inches.

Border

Specifies the length of the border for the tabular.

Do not show this message again

Specifies whether or not to show the wizard the next time when you create a tabular report.

Page Setup

Opens the Page Setup dialog to specify the page settings.

Finish

Finishes the setting and closes this wizard.

Cancel

Does not retain changes and closes this wizard.

Help

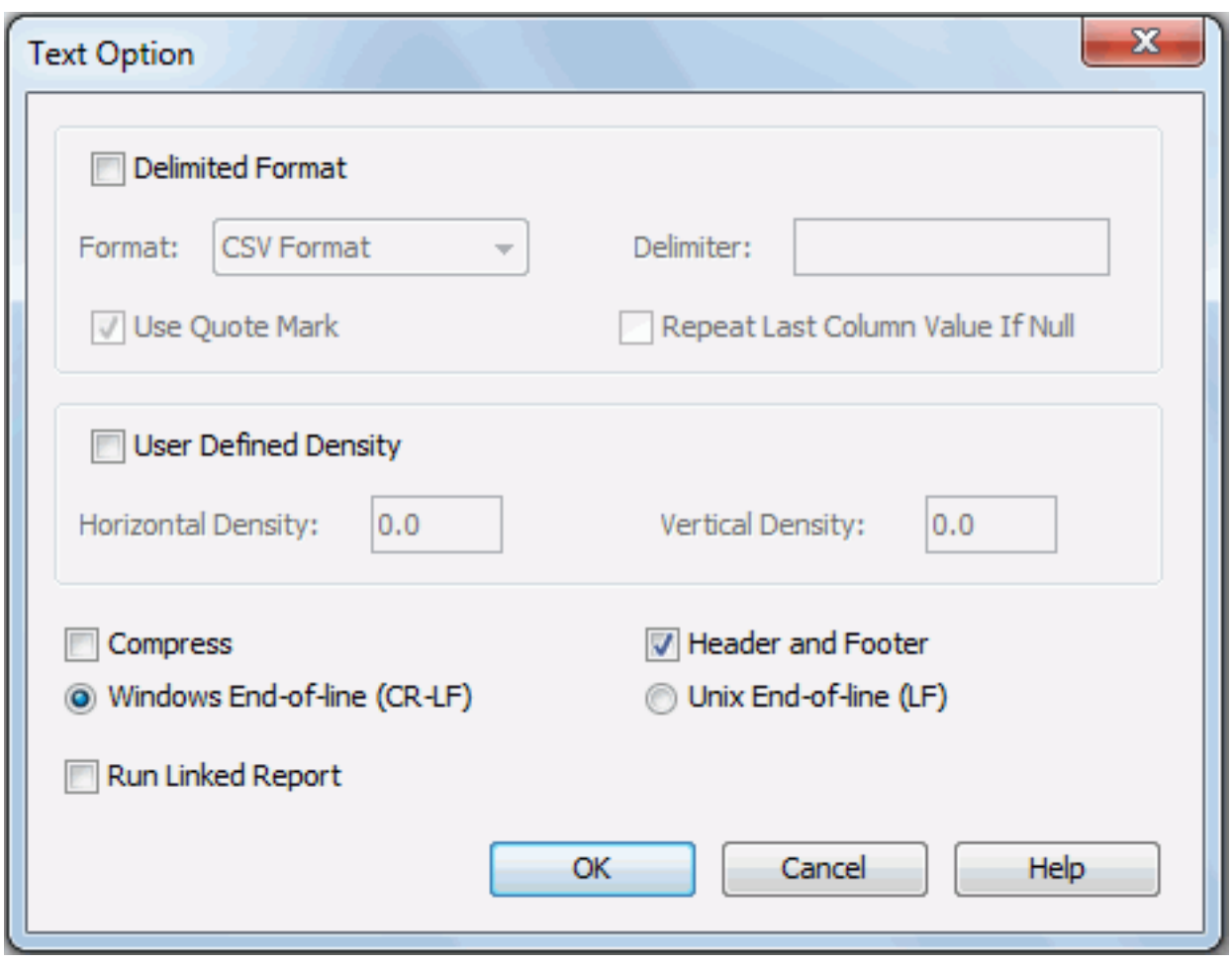
Displays the help document about this feature.

Related topics:

- Creating a tabular report

Text Option dialog

The dialog appears when you set the Default Format for Viewing Report property of a report tab or a web report to Text and then click ... in the value cell in the Report Inspector. It helps you to specify the parameters for running a report in Text format.



The following are details about options in the dialog:

Delimited Format

Specifies whether to use the delimited format for the Text result.

- **Format**
  Specifies the delimiter to separate fields.
  - **CSV Format**
    Fields in the Text result will be separated by a comma.
  - **Tab Delimited**
    Fields in the Text result will be separated by a tab.
  - **Custom Defined**
    Fields in the Text result will be separated by a user defined delimiter. You can type your own delimiter in the Delimiter box (note that the delimiter should be only one character).
- **Use Quote mark**
  Specifies whether the fields in the Text result will be marked with quotation marks.
- **Repeat Last Column Value If Null**
  If checked, when a cell in the Text result has no value, value of the previous cell in the same column will be used.

User Defined Density

Specifies whether to use user defined density for the Text result.

- **Horizontal Density**
  Specifies the value for each unit of the horizontal density between columns. The resulting density is a direct ratio of the value you specify. That is, the greater the value, the smaller the width between columns.
- **Vertical Density**
  Specifies the value for each unit of the vertical density between columns. The resulting density is a direct ratio of the value you specify. That is, the greater the value, the smaller the height between columns.

Notes:

- When using user defined densities, if the densities are not set appropriately, the fields in the report may overlap each other, so you are not recommended to use this way.
- When setting the value of Horizontal/Vertical Density, you need to pay attention to the following:
  - The value of Horizontal/Vertical Density must be greater than the character's width/height of the smallest field in the report (smallest field is the field with the smallest font size), otherwise, the value you set will not be applied.
  - If the value of Vertical Density is greater than 0 and the value of Horizontal Density is less than 0, the value that you specify for the Vertical Density will be applied and the value of Horizontal Density will be specified by JReport.
  - If the value of Vertical Density is less than 0 and the value of Horizontal Density is greater than 0, the value that you specify for the Horizontal Density will be applied and the value of Vertical Density will be specified by JReport.
  - If the values of Vertical Density and Horizontal Density are both greater than 0 and the value of Horizontal Density is less than 11, the specified value of the two densities will be applied. Otherwise, they will be specified by JReport.
  - If the values of Vertical Density and Horizontal Density are both less than 1, these densities will be specified by JReport.

Compress

Specifies whether to generate the Text result in a compressed size, that is to say, there will be no clearance between the columns.

Header and Footer

If checked, the Text result will contain all headers and footers in the report, including Report Header/Footer, Page Header/Footer and Group Header/Footer. Otherwise, the Text result will only contain data in Detail panel.

Windows End-of-line (CR-LF)

Specifies to use Windows end-of-line characters to indicate the start of a new line. If checked, two characters <cr> and <lf> will be used at the end of the line.

Unix End-of-line (LF)

Specifies to use Unix End-of-line characters to indicate the start of a new line. If checked, only the Unix End-of-line character <lf> will be used.

Run Linked Report

If the report is linked with other reports, you can check whether or not to generate the linked reports in the Text result. If you are only interested in the primary report, leave this option unchecked. Generating linked reports at the same time, especially when the linked reports contain a large amount of data, will cause performance issue.

OK

Applies the changes and closes the dialog.

Cancel

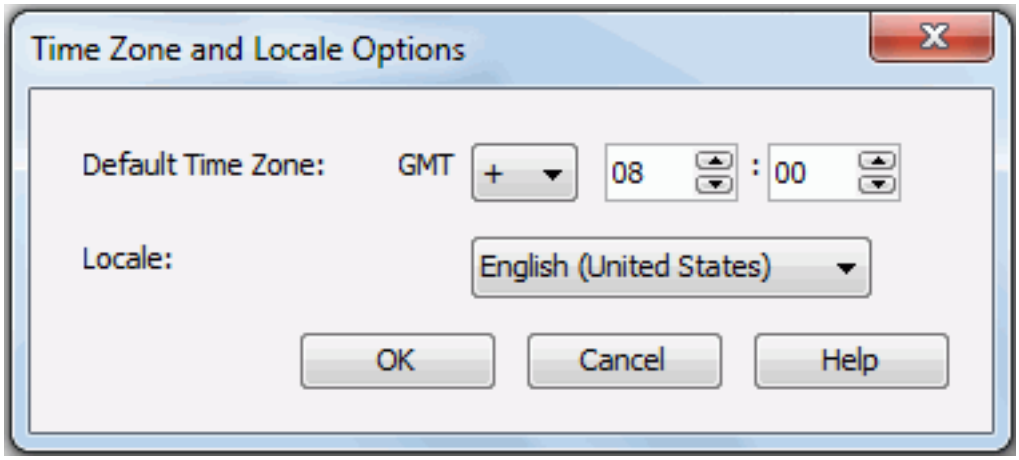
Cancels the changes and exits the dialog.

Help

Displays the help document about this feature.

Time Zone and Locale Options dialog

This dialog appears when you click the Time Zone and Locale button in the XML Connection Wizard. It helps you to specify a default time zone and locale for an XML instance.



The following are details about options in the dialog:

Default Time Zone

Specifies the default time zone for the XML data source, in order to apply it to the Time data type, which is not specified for them in the XML instance. If the time zone has been specified in the XML instance, the default time zone that you set here will be ignored.

Locale

Specifies the locale for the XML instance.

OK

Applies the changes and closes the dialog.

Cancel

Does not retain changes and closes the dialog.

Help

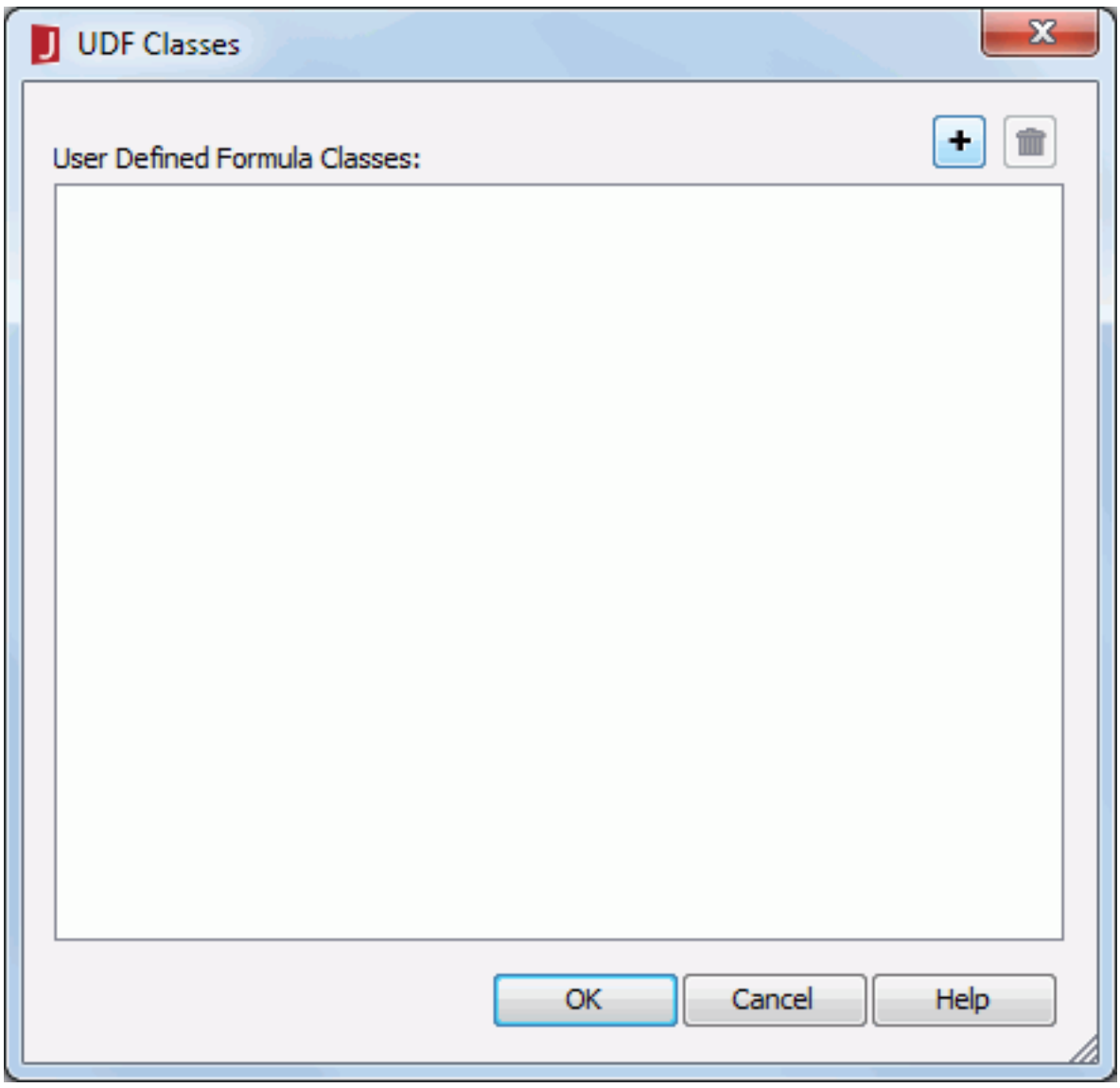
Displays the help document about this feature.

Related topics:

- Setting up an XML connection

UDF Classes dialog

This dialog appears when you click File > Import UDF Classes in the Formula Editor window. It helps you to import user defined formula classes.



The following are details about options in this dialog:

User Defined Formula Classes

Lists the imported user defined formula classes with package path and alias.

By default, each imported UDF class is given an alias which takes the name of the UDF class even though the alias is not shown. You can modify the alias by double-clicking the UDF class and then specify a new alias in the Update UDF Class Alias dialog. The new alias which is different from the UDF class name will be displayed right next to the UDF class and be bracketed by "[]". The format is like this: UDF_Class_Name[Alias_Name].

Opens the Import UDF Class dialog for importing a new UDF class. To import a UDF class, you need first add it to the class path of JReport Designer and then provide the class name with full package path in the Import UDF Class dialog.

Removes the selected UDF class.

OK

Applies the changes and closes the dialog.

Cancel

Does not retain any changes and closes the dialog.

Help

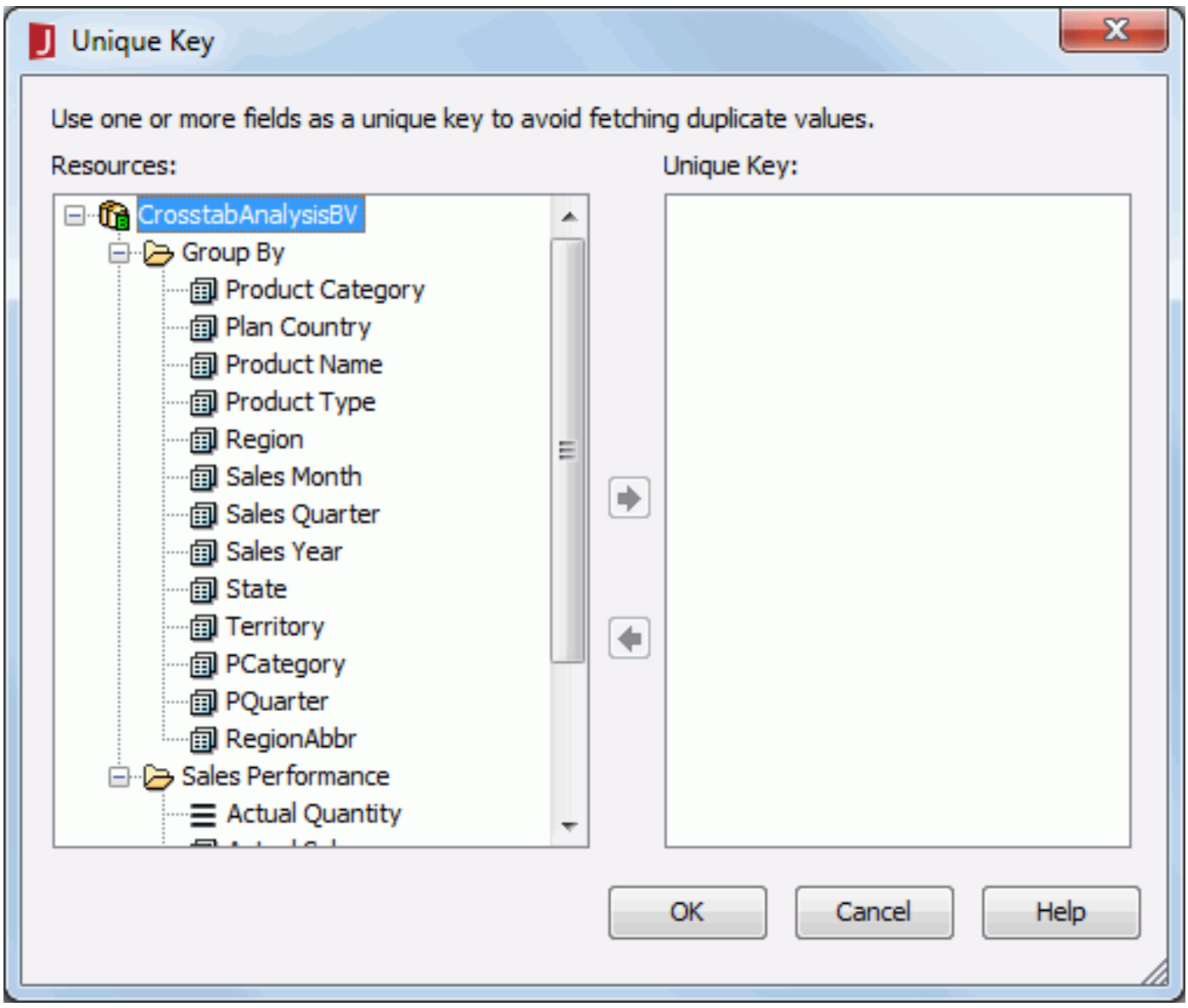
Displays the help document about this feature.

Related topics:

- User defined formula functions

Unique Key dialog

This dialog appears when you click ... in the value cell of the Unique Key property of of a refresh object in a library component in the Report Inspector, or click the Incremental Fetch button in the Display screen of the chart wizard. It helps you to specify a unique key to ensure there are no duplicated records in the library component or real time chart.



The following are details about options in the dialog:

Resources

Lists all the available data resources in the dataset used by the library component or real time chart.

Adds the selected fields to the Unique Key box.

Removes the selected fields from the Unique Key box.

Unique Key

Lists the fields you have added as the unique key, which is used to guarantee there are no two records with the same unique key value in the library component or real time chart.

Take real time chart for example. When a real time chart is specified with a unique key, each time when the chart automatically updates itself, the data that has the same unique key value as the previous loaded data records will be filtered and only data with different unique key value will be added to the chart. For instance, if you add the fields Country and Product ID as the unique key of a real time chart, when a record with the product ID 1 in USA has already been loaded into the chart, no more records of this product ID in USA will be added to the real time chart because they have the same unique key value.

OK

Applies the changes and closes the dialog.

Cancel

Does not retain any changes and closes the dialog.

Help

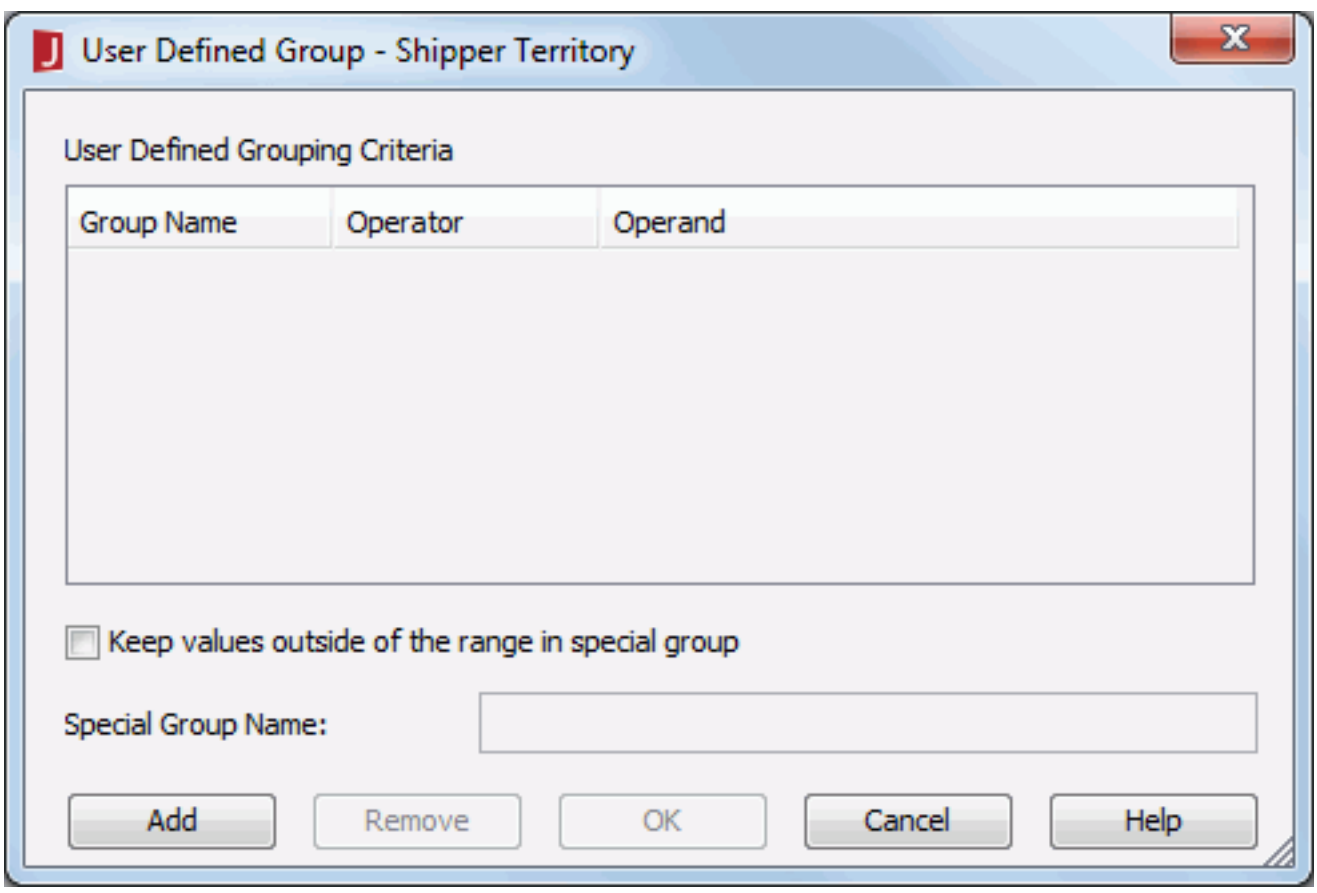
Displays the help document about this feature.

Related topics:

- Creating a real time chart

User Defined Group dialog

This dialog appears when you select Special Group from the Sort column in the Group screen of the banded/table wizard or Map Binding Data Wizard, or click the Special Group button in the Display screen of the chart wizard. It helps you to define how to group your information.



The following are details about options in the dialog:

Group Name

Specifies the name for the grouping criterion.

If you use this dialog to group data in the dataset that is bound with a map, the group name here should match with data in the specified group by field.

Operator

Specifies the operator with which to compose the grouping criterion.

There are many operators with which you can define how your data will be grouped. You can define as: between, not between. > (greater than), < (less than), = (equal to), <= (less than or equal to), >= (greater than or equal to), or != (not equal to).

Operand

Either op1 or both op1 and op2 will be seen in the Operand column. Type the criteria with quotation marks here.

Keep values outside of the range in special group

If checked, JReport will put the values that are not included in the specified criteria in a new special group.

- **Special Group Name**
  Specifies the name for the special group.

Add

Adds a new grouping line.

Remove

Removes the grouping line you specify.

OK

Applies all changes.

Cancel

Does not retain any changes.

Help

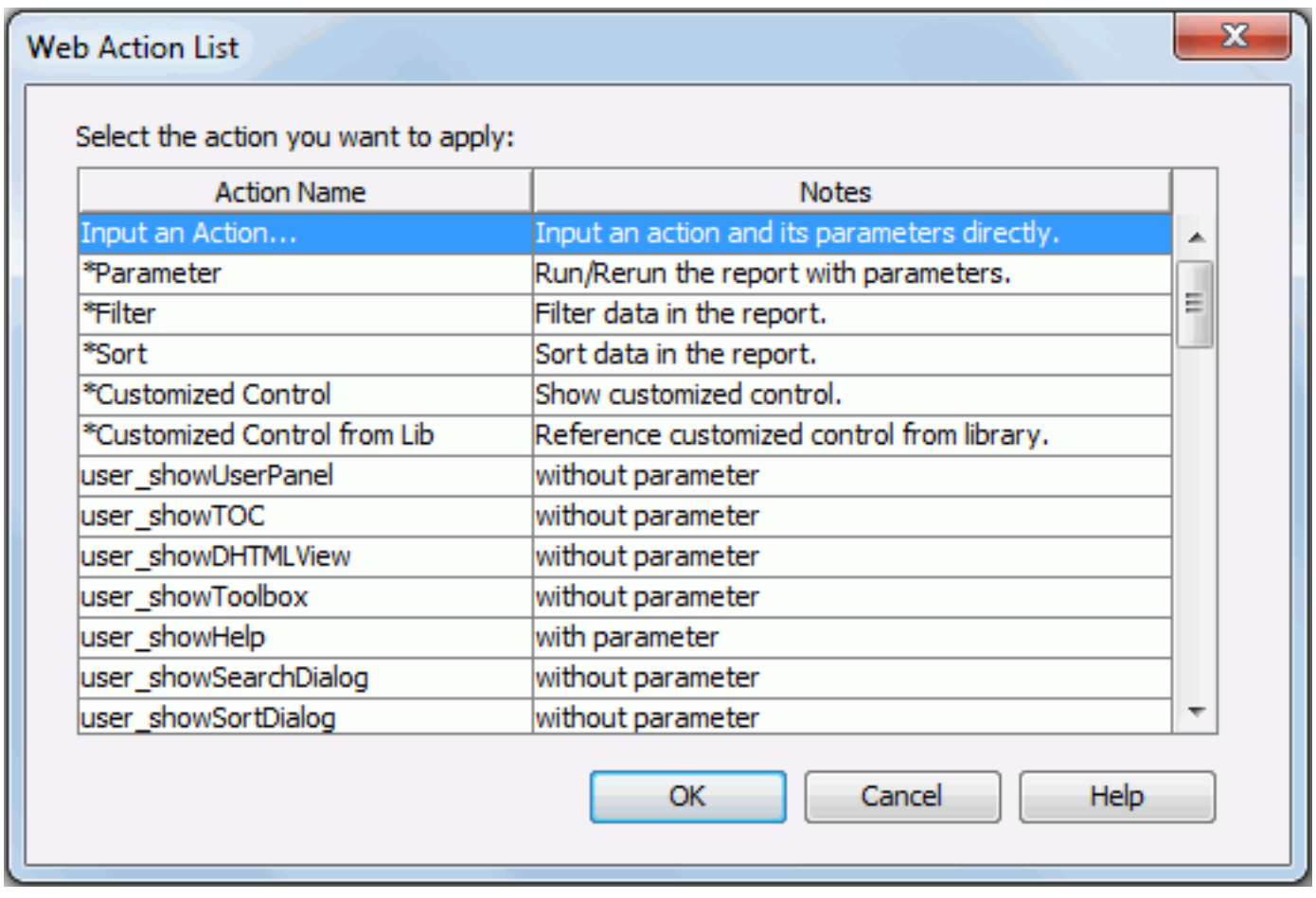
Displays the help document about this feature.

Related topics:

- Setting the sort manner

Web Action List dialog

This dialog appears when you click ... in the Actions column in the Web Behaviors box of the Display Type dialog, in the Web Behaviors dialog, in the Behaviors tab of some chart formatting dialogs, or in the Receive Message dialog when receiving a user defined message. It helps you to bind one or more web actions to an object in a page report or library component.



The following are details about options in the dialog:

Action Name

Lists the names of actions.

- **Input an Action**
  If selected, click OK to open the Input Action dialog to input directly an action and its parameter and bind it to the object. Not supported on library components.
- ***Parameter**
  If selected, click OK to open the Parameter - Web Action Builder dialog or Parameter dialog to build the Parameter web action to the object or to do the Parameter action to respond a received message.
- ***Filter**
  If selected, click OK to open the Filter - Web Action Builder dialog or Filter dialog to build the Filter web action to the object or to do the Filter action to respond a received message.
- ***Sort**
  If selected, click OK to open the Sort - Web Action Builder dialog or Sort dialog to build the Sort web action to the object or to do the Sort action to respond a received message.
- ***Customized Control**
  If selected, click OK to open the Customized Control •Web Action Builder dialog to bind a customized control web action. Available to page report tables only.
- ***Customized Control from Lib**
  If selected, click OK to open the Customized Control Manager dialog to reference a customized control file from the library.
- ***Property**
  If selected, click OK to open the Change Property - Web Action Builder dialog or Change Property dialog to build the Change Property web action to the object or to do the Change Property action to respond a received message. Available to library components only.
- ***SendMessage**
  If selected, click OK to open the Send Message - Web Action Builder dialog to bind the Send Message web action to the object. Available to library components only.
- **User defined web action - with parameter**
  If selected, click OK to open the Parameters dialog to specify values of the parameters of a web action. Not supported on library components.
- **User defined web action - without parameter**
  If selected, click OK to bind the selected action to the web control. Not supported on library components.

Notes

Displays some tips about the specified action.

OK

Accepts the changes and closes the dialog.

Cancel

Discards the changes and closes the dialog.

Help

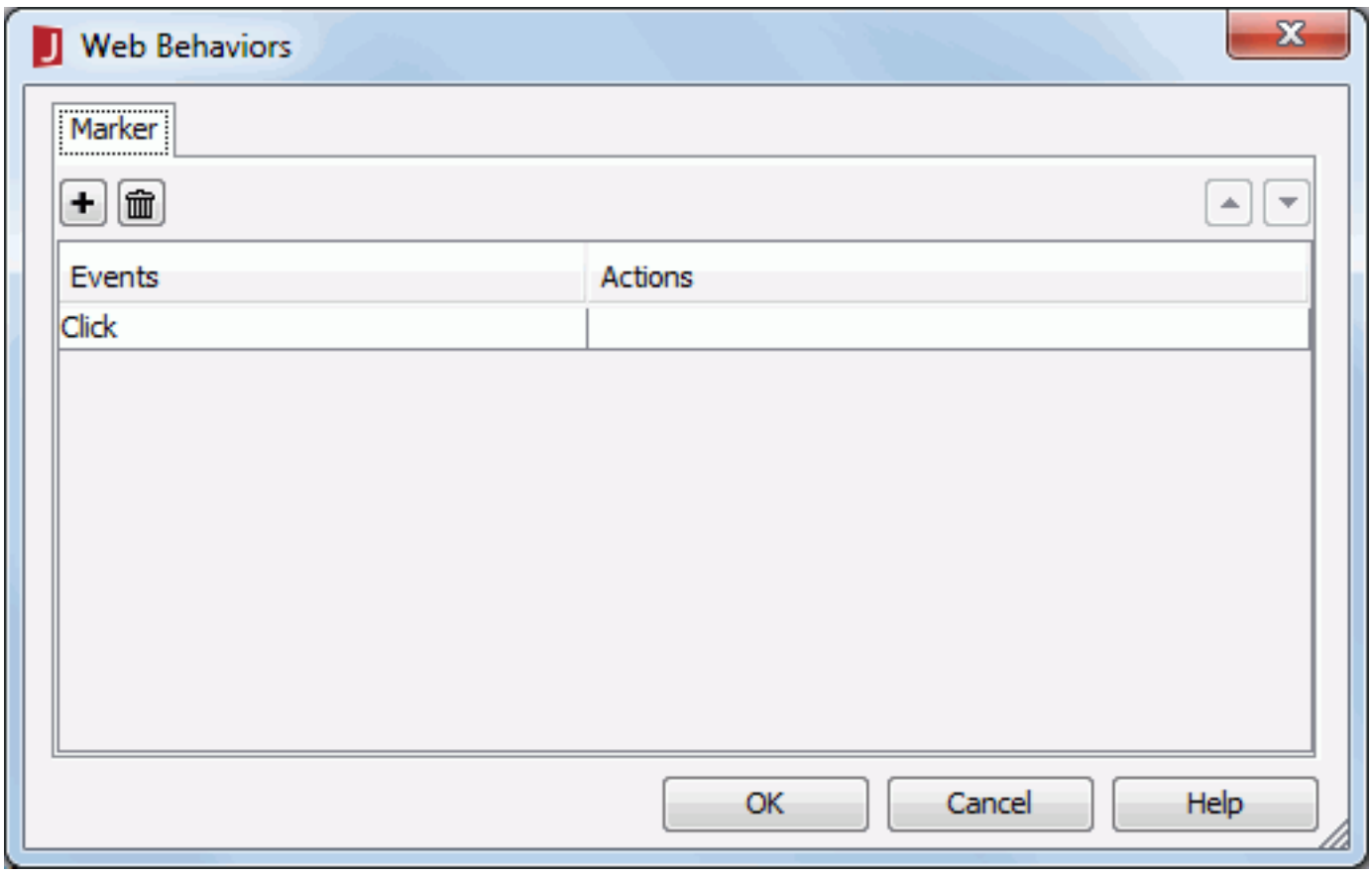
Displays the help document about this feature.

Related topics:

- Applying web actions to a label
- Receiving messages

Web Behaviors dialog

This dialog appears when you click the Web Behaviors button in the Display screen of the map wizard. It helps you to specify some web behaviors to the markers and/or areas of the selected group in the geographic map, and consists of the Marker and/or Area tabs with the same contents.



The following are details about options in this dialog:

Marker tab

Specifies some web behaviors to the markers of the selected group in the geographic map. Available when Enable Marker in the Display screen of the map wizard is checked.

Area tab

Specifies some web behaviors to the areas of the selected group in the geographic map. Available when Enable Area in the Display screen of the map wizard is checked.

Adds a new web behavior line.

Removes the selected web behavior.

Moves the selected web behavior up a step.

Moves the selected web behavior down a step.

Events

Specifies the trigger event from the drop-down list.

Actions

Specifies the web action you want the event to trigger.

- Opens the Web Action List dialog to bind a web action to the event.

OK

Accepts the changes and closes the dialog.

Cancel

Does not retain changes and closes this dialog.

Help

Displays the help document about this feature.

Related topics:

- Inserting a geographic map to a library component

Web Options dialog

This dialog appears when you right-click a web control in the configuration panel of a library component and select Web Options from the shortcut menu. It helps you to define values of the web control and varies with the web control type.

- Text Field
- Checkbox
- Drop-down List
- List

OK

Accepts the changes and closes the dialog.

Cancel

Discards the changes and closes the dialog.

Apply

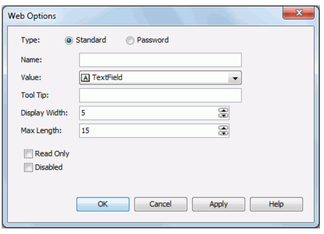
Applies all changes and leaves the dialog open.

Help

Displays the help document about this feature.

Text Field

For a text field web control, options in the dialog are as follows.



Type

Specifies the type of the text field.

- **Standard**
  Specifies that the text field is a standard text field.
- **Password**
  Specifies that the text field is a password text field, which means that the value input in the text field will be hidden using asterisks.

Name

Specifies the name of the text field.

Value

Specifies the value of the text field.

Tool Tip

Specifies the tooltip of the text field, which will be shown when you hover the mouse over the text field at runtime.

Display Width

Specifies the display width of the text field.

Max Length

Specifies the maximum length of the string that is allowed in the text field.

Read Only

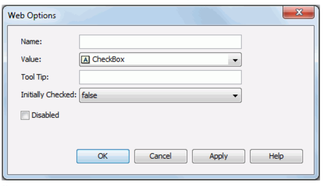
Specifies whether to make the field read-only to others.

Disabled

Specifies whether the text field is disabled or enabled.

Checkbox

For a checkbox web control, options in the dialog are as follows.



Name

Specifies the name of checkbox.

Value

Specifies the value of the checkbox.

Tool Tip

Specifies the tooltip of the text field, which will be shown when you hover the mouse over the text field at runtime.

Initially Checked

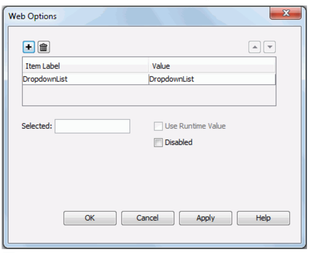
Specifies whether or not the checkbox is selected by default.

Disabled

Specifies whether the checkbox is enabled or disabled.

Drop-down List

For a drop-down list web control, options in the dialog are as follows.



Adds a new item in the drop-down list.

Removes the selected item from the drop-down list.

Moves the selected item in the drop-down list up a step.

Moves the selected item in the drop-down list down a step.

Item Label

Specifies the display text or format of the item value.

Value

Specifies the item value. Type the value directly in the text box or click the button ... next to it to open the Insert Fields dialog where you can add field to control the value.

Selected

Specifies the value that will be selected in the drop-down list by default.

Use Runtime Value

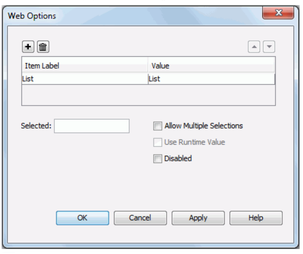
When changing the display type of a parameter to drop-down list, you can check this option to use the runtime value as the selected value of the drop-down list.

Disabled

Specifies whether the drop-down list is disabled or enabled.

List

For a list web control, options in the dialog are as follows.



Adds a new item in the list.

Removes the selected item from the list.

Moves the selected item in the list up a step.

Moves the selected item in the list down a step.

Item Label

Specifies the display text or format of the item value.

Value

Specifies the item value. Type in the value directly in the text box or click the button ... next to it to open the Insert Fields dialog where you can add field to control the value.

Selected

Specifies the value that will be selected in the list by default.

Allow Multiple Selections

Specifies whether to allow selecting multiple items in the list at the same time.

Use Runtime Value

When changing the display type of a parameter to list, you can check this option to use the runtime value as the selected value of the list.

Disabled

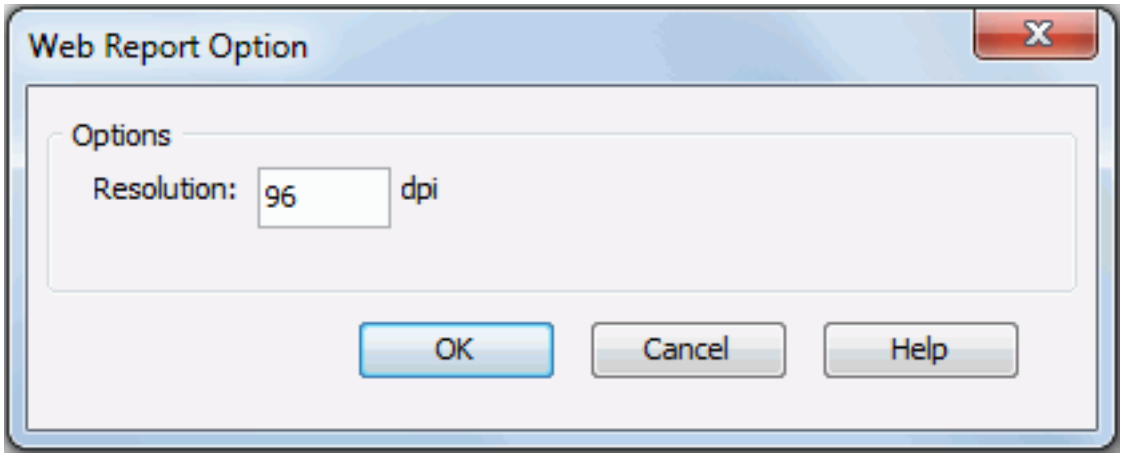
Specifies whether the list is disabled or enabled.

Related topics:

- Inserting objects into the panel

Web Report Option dialog

The dialog appears when you set the Default Format for Viewing Report property of a web report to Web Report and then click ... in the value cell in the Report Inspector.



Resolution

Specifies the resolution of the result to zoom in/out, in DPI. The default value is obtained from the operation system, which is the resolution of your monitor, for example, 72 DPI on Unix or 96 on Windows. You can set higher/lower value to zoom in/out.

OK

Applies the changes and closes the dialog.

Cancel

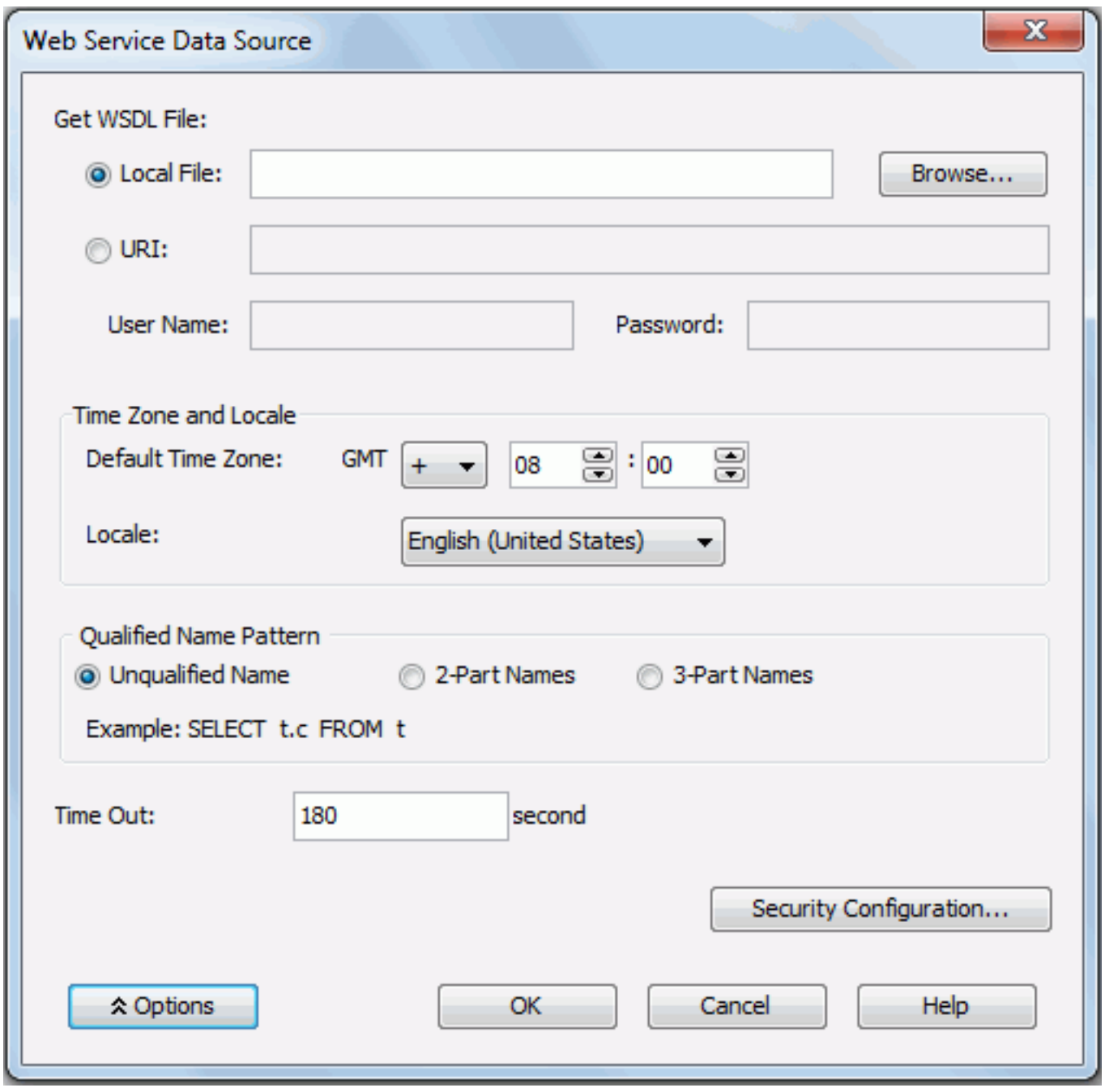
Cancels the changes and exits the dialog.

Help

Displays the help document about this feature.

Web Service Data Source dialog

This dialog appears when you select SOAP Web Service and click OK in the New Data Source dialog, or in the Catalog Browser right-click a data source and select Add SOAP Web Service Connection from the shortcut menu or right-click an existing web service connection and select Edit Connection from the shortcut menu. It helps you to create or edit a web service data source in a JReport catalog.



The following are details about the options of the dialog:

Get WSDL file

Specifies to get a WSDL file to create the web service data source.

- **Local File**
  Specifies to get a WSDL file from local files by clicking the Browse button.
- **URI**
  Specifies the location of the WSDL file by inputting the URI string.
  - **User Name**
    Specifies the user name used for accessing the WSDL file through the URI.
  - **Password**
    Specifies the password used for accessing the WSDL file through the URI.

Time Zone and Locale

Specifies a default time zone and locale for the web service data source.

- **Default Time Zone**
  Specifies the default time zone for the web service data source.
- **Locale**
  Specifies the locale for the web service data source.

Qualified Name Pattern

Specifies whether or not catalog or schema is used in data manipulation.

- **Unqualified Name**
  Neither catalog nor schema will be included in data manipulation. Example: SELECT t.c FROM t
- **2-Part Names**
  Uses schema in data manipulation. Example: SELECT schema.t.c FROM schema.t
- **3-Part Names**
  Uses both catalog and schema in data manipulation. Example: SELECT catalog.schema.t.c from catalog.schema.t

Time Out

Specifies how long to wait to get the WSDL file.

Security Configuration

Opens the Security Configuration Setting dialog to configure security for the web service data source.

Options

Shows the additional settings for creating the web service data source.

OK

Creates the web service data source and closes the dialog.

Cancel

Cancels the creation of the web service data source and exits the dialog.

Help

Displays the help document about this feature.

Related topics:

- Web service data sources

Web Wizard - Export to HTML

This wizard appears when you click File > Export To > HTML on the menu bar. It helps you to export a report to HTML format, and consists of the following tabs:

- Report tab
- Directory tab
- Chart Applet tab
- Note tab

Previous

Retains changes and returns to the previous tab.

Next

Retains changes and goes to the next tab.

Finish

Completes the exporting.

Cancel

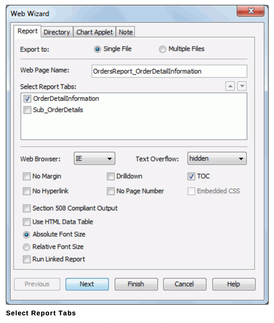
Does not retain any changes and closes the wizard.

Help

Displays the help document about this feature.

Report tab

Specifies the options applied to the report.



Select Report Tabs

When exporting a page report, you can specify the report tabs in the page report you want to export. The selected report tabs will be exported in the list order. If the report has only one report tab, it is selected by default.

Web Page Name

Specifies the name of the exported file. When exporting a page report, you need to specify the name for each report tab.

If you do not type the name, JReport will use the report name as the default HTML file name. JReport automatically attaches the extension *.html, so you do not have to type the extension yourself.

Moves the specified report tab one step up.

Moves the specified report tab one step down.

Web Browser

Specifies the web browser for which the HTML result adapts.

Text Overflow

Specifies whether the text overflow is visible or hidden.

No Margin

Specifies whether to remove the margins originally set when the report was designed.

Drilldown

Specifies whether to include the drilldown file in the exported HTML file.

TOC

Specifies whether to enable the TOC (Table of Contents) in the exported HTML file.

No Hyperlink

If checked, there will be no hyperlinks for navigating previous and next pages on the navigation bar of the exported HTML file. Enabled only when Single File is selected.

No Page Number

If checked, there will be no page number information showing the current page number and total page number on the navigation bar of the exported HTML file. Enabled only when Single File is selected.

Embedded CSS

Specifies whether to embed the cascading style sheet (CSS) in the exported HTML files; otherwise, the .css file will be generated individually. Enabled only when Multiple Files is selected.

Section 508 Compliant Output

If checked, the accessibility attributes defined for the report elements via the Report Inspector will be exported to the HTML result which is Section 508 compliant. See Accessibility for more information.

When Section 508 Compliant Output is checked, the Use HTML Data Table and Relative Font Size options will be checked and disabled. The output will be Section 508 compliant including HTML data table, accessible attributes, and relative font feature.

Use HTML Data Table

Specifies whether the table and crosstab components will be output as table objects in the HTML result.

Absolute Font Size

Specifies whether to generate the report result using an absolute font size, which means the font size is fixed and cannot be adjusted according to the font size settings in the web browser.

Relative Font Size

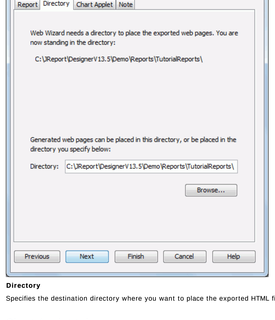
Specifies whether to generate the report result using a relative font size, which means the font size can be adjusted according to the font size settings in the web browser.

Run Linked Report

If the report is linked with other reports, you can check whether or not to generate the linked reports in the exported HTML file. If you are only interested in the primary report, leave this option unchecked. Generating linked reports at the same time, especially when the linked reports contain a large amount of data, will cause performance issue.

Directory tab

Specifies the location in which to place the exported HTML files.

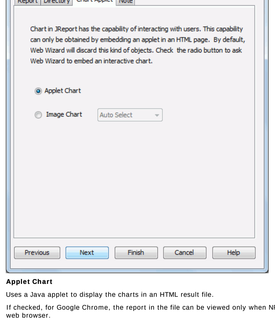


Directory

Specifies the destination directory where you want to place the exported HTML file.

Chart Applet tab

If you have used dynamic charts in your report, they can still be available in the exported HTML file. If Embed chart applet has been checked, dynamic 3-D charts will remain interactive and be available in the outputted HTML file. If Save in image has been checked, the charts will be converted to images. Also, you can specify the format of the images by choosing from the drop-down list (GIF or JPEG. The default format is JPEG).



Applet Chart

Uses a Java applet to display the charts in an HTML result file.

If checked, for Google Chrome, the report in the file can be viewed only when NPAPI is enabled for the web browser.

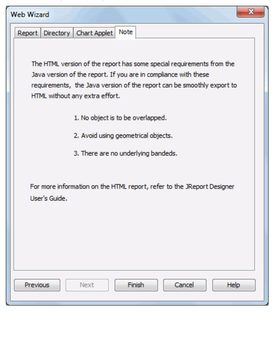
Image Chart

If checked, charts will be converted to images. Also, you can specify the image type from the drop-down list. The options are:

- **Auto Select**
  If selected, the image format will be set to JPG or GIF by the JReport system automatically. If the image colors are less than 256 colors, GIF will be applied; otherwise, it is JPG.
- **GIF**
  If selected, the GIF format will be applied, which is a lossless compression technique and supports only 256 colors. GIF is better than JPG for images with only a few distinct colors, such as line drawings, black and white images and small text that is only a few pixels high.
- **JPG (JPEG)**
  If selected, the JPG (JPEG) format will be applied, which is supported on the Web. JPG is a lossy compression technique that is designed to compress color and grayscale continuous-tone images. JPG images support 16 million colors and are best suited for photographs and complex graphics.
- **PNG**
  If selected, the PNG format will be applied, which provides a portable, legally unencumbered, well-compressed (effectively 100 percent lossless compression), well-specified standard for lossless bitmapped image file. PNG supports indexed-color images of up to 256 colors and shows a more interchangeable, flexible and robust function than GIF.

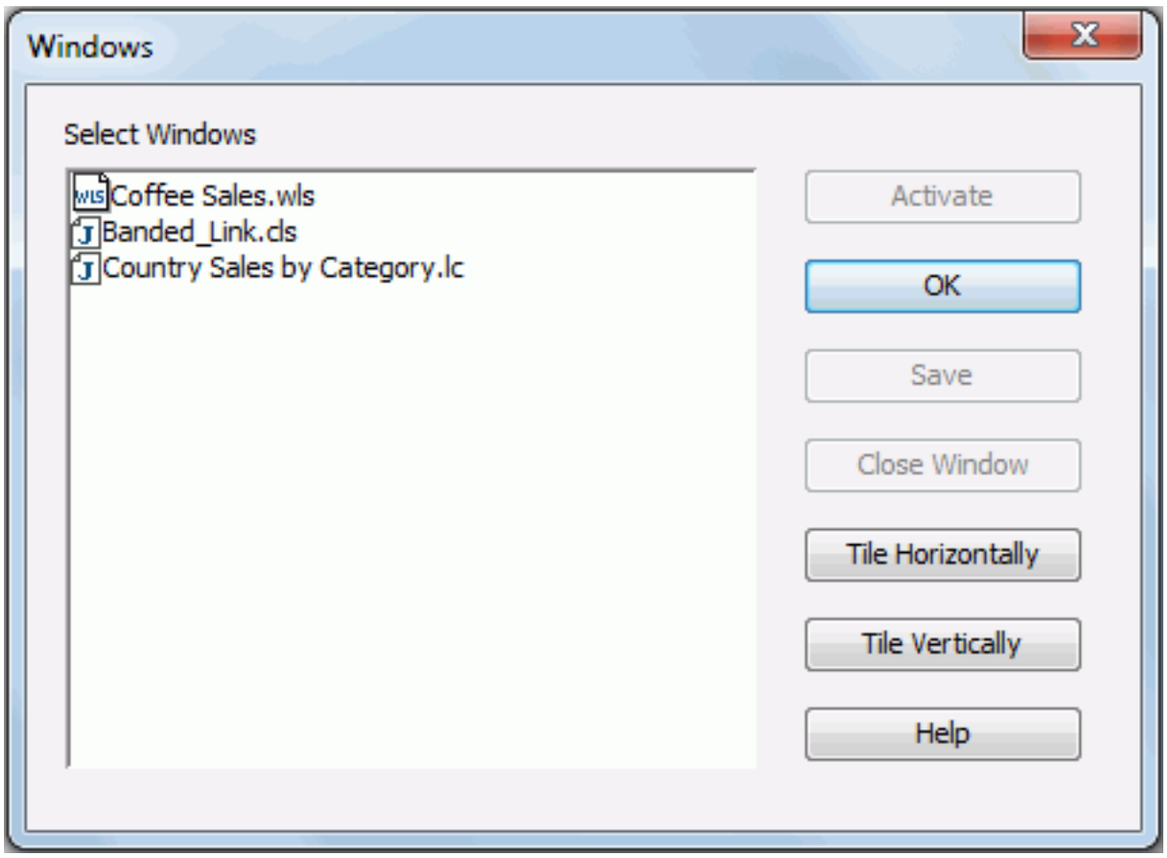
Note tab

This tab includes important information about HTML. We recommend that you read the contents in the Note tab before you convert your report to HTML format.



Windows dialog

This dialog appears when you click Window > Windows on the menu bar. It is used to adjust the multiple reports to suit your requirements.



The following are details about options in the dialog:

Select Windows

Lists all the open reports.

Activate

Activates the report selected in the Select Windows box.

OK

Accepts all changes and closes the dialog.

Save

Saves the reports selected in the Select Windows box.

Close Window

Closes the report selected in the Select Windows box.

Tile Horizontally

Lays out all of the open reports horizontally.

Tile Vertically

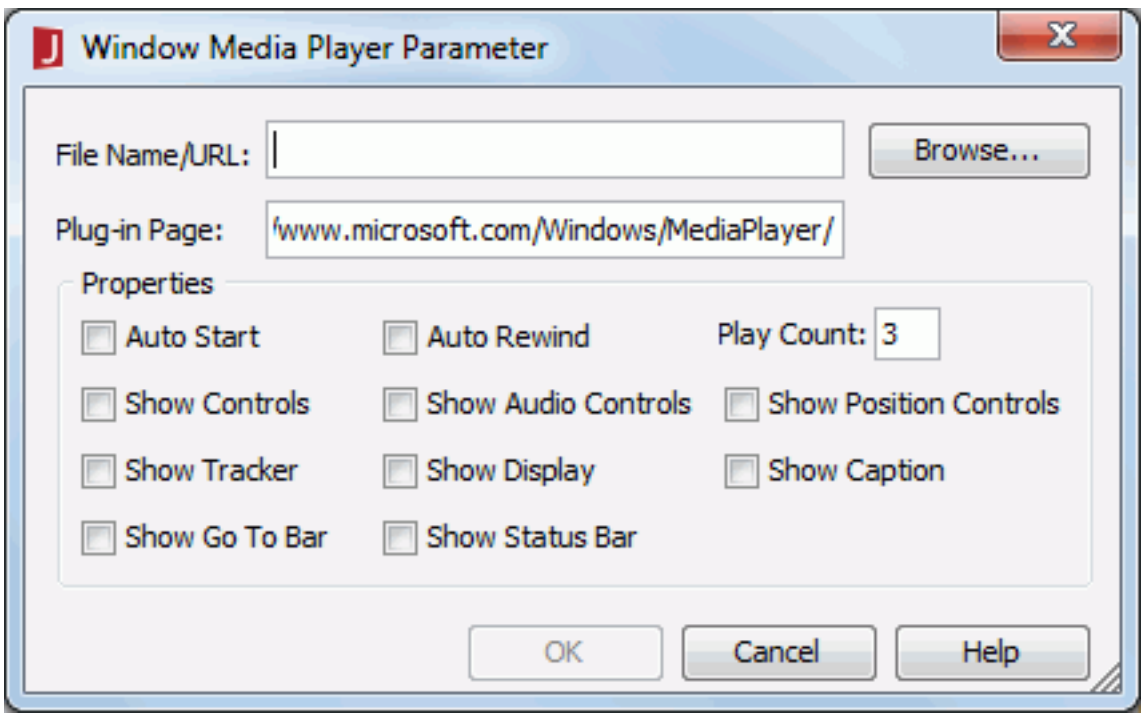
Lays out all of the open reports vertically.

Help

Displays this help document

Windows Media Player Parameter dialog

This dialog appears when you click Insert > Multimedia > Windows Media File on the menu bar, or drag Windows Media File from the Toolbox panel to a report. It helps you to insert a Windows Media file into a report in JReport Designer.



The following are details about options in the dialog:

File Name/URL

Specifies the name or URL of the media file that is to be inserted into the report. You can click the Browse button to specify the path of the name or URL, or type in the path directly in the text field.

Plug-in Page

Specifies the URL of the plug-in page from which to download the player with which to play the media file, if it isn't already installed on your local disk.

Properties

Specifies properties for the media file to control the options when playing the media file.

- **Auto Start**
  Specifies whether or not to start the media file automatically when the report is opened in Page Report Studio or Web Report Studio.
- **Auto Rewind**
  Specifies whether or not to rewind the media file automatically.
- **Play Count**
  Specifies the number of times you want to play this media file.
- **Show Controls**
  Specifies whether or not to show the Windows Media Player controls.
- **Show Audio Controls**
  Specifies whether or not to show the Mute Button and the Volume Slider on the Control Bar.
- **Show Position Controls**
  Specifies whether or not to show the Fast Forward, Fast Rewind, Skip Ahead, Skip Previous, and Preview buttons on the Control Bar.
- **Show Tracker**
  Specifies whether to show the Tracker Bar. It is only enabled when the Control Bar is visible.
- **Show Display**
  Specifies whether to show the Display Panel.
- **Show Caption**
  Specifies whether to show the Closed Captioning Displayed Panel.
- **Show Go To Bar**
  Specifies whether to show the Go To Bar.
- **Show Status Bar**
  Specifies whether to show the Status Bar.

OK

Accepts all changes and closes the dialog.

Cancel

Does not retain any changes and closes the dialog.

Help

Displays the help document about this feature.

Related topics:

- Multimedia objects

XML Connection Wizard

This wizard appears when you select XML and click OK in the New Data Source dialog, or in the Catalog Browser right-click a data source and select Add XML Connection from the shortcut menu or right-click an existing XML connection and select Edit Connection from the shortcut menu. It helps you to import an XML schema and transform it to a relational schema for future use, and consists of the following screens:

- Import XML Schema screen
- Modify Schema Properties screen
- Transform XML Schema screen
- Transformed Relational Schema screen
- Add Table screen

Back

Goes back to the previous screen.

Next

Goes to the next screen.

Finish

Finishes confirming to transform an XML schema to a relational schema.

Cancel

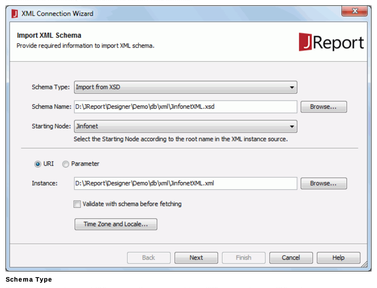
Does not retain changes and closes this wizard.

Help

Displays the help document about this feature.

Import XML Schema screen

Specifies the necessary information to import XML schema.



Schema Type

Specifies the type to import XML schema. You can import from XSD or parse from an XML instance.

Schema Name

Specifies the schema file by clicking the Browse button or inputting the URI string directly in the text box.

Starting Node

Specifies the starting node, according to the root name in the XML source.

Notes:

- If a value of DateTime data type is accurate to millisecond, a message - Invalid Value will still pop-up although the value is valid according to the W3C standard. If that, ignore the error message and continue the transformation steps.
- If you select to create a parameter and then dynamically specify an XML instance at runtime, the XML instance will be validated at runtime rather than in the importing process.

Instance

Specifies the XML instance. It should match with the selected XSD schema.

These two operations below are available only when you select Import from XSD from the Schema Type drop-down list.

- **URI**
  If selected, specify an XML file for the XSD by clicking the Browse button or input the URI string directly in the text box.
- **Parameter**
  If selected, click to open the Create Parameter dialog to create a parameter and then dynamically specify an XML instance that matches with the selected XSD schema at runtime.

Validate with schema before fetching

If selected, JReport Designer will validate whether or not values in the XML instance is valid according to the W3C standard and the specified XSD schema. If not, a message - Invalid Value will pop up.

Time Zone and Locale

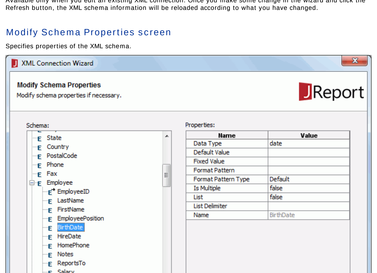
Opens the Time Zone and Locale Options dialog to specify the default time zone and locale for the XML instance.

Refresh

Available only when you edit an existing XML connection. Once you make some change in the wizard and click the Refresh button, the XML schema information will be reloaded according to what you have changed.

Modify Schema Properties screen

Specifies properties of the XML schema.



Schema

Lists the corresponding schema structure of the root, stands for elements, stands for the attributes in the XML schema and stands for the references of elements.

Properties

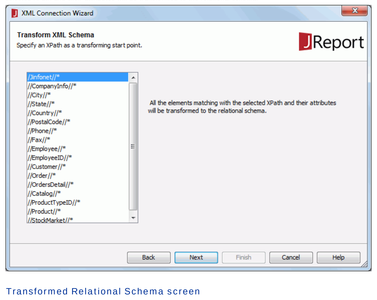
Lists all the properties of the selected elements and attributes in the schema.

- **Name**
  Displays names of properties of the selected elements or attributes.
- **Value**
  Specifies values of properties for the selected elements or attributes.

Note: When the property Data Type is specified as Date, Time or DateTime, you can specify the Format Pattern for them to make values of the data type displayed in the required format. For details about these properties, see the section Supplementing information for the XML schema.

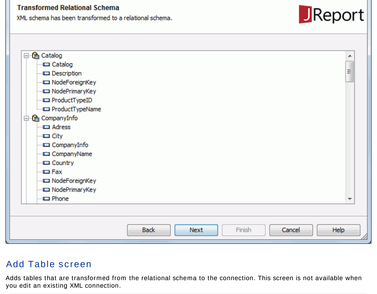
Transform XML Schema screen

This screen allows you to specify an XPath as a transforming start point. All the elements matching with the selected XPath and their attributes will be transformed to the relational schema.



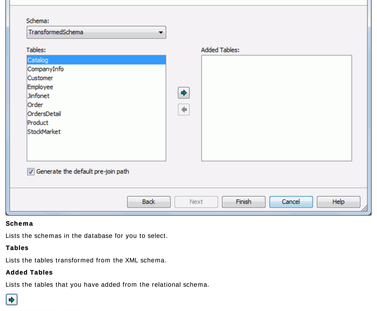
Transformed Relational Schema screen

This screen lists the transformed relational schema structure. It helps you to click Next to confirm the transformation results and go into the Add Table screen.



Add Table screen

Adds tables that are transformed from the relational schema to the connection. This screen is not available when you edit an existing XML connection.



Schema

Lists the schemas in the database for you to select.

Tables

Lists the tables transformed from the XML schema.

Added Tables

Lists the tables that you have added from the relational schema.

Adds the selected tables.

Removes the selected tables from the Added Tables box.

Generate the default pre-join path

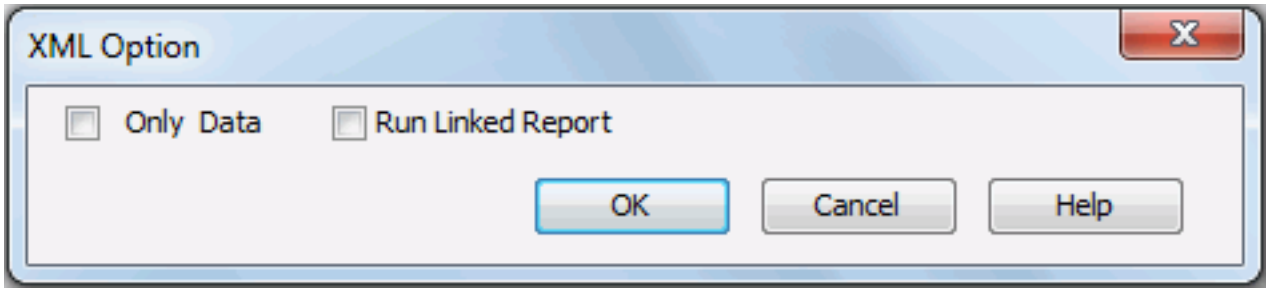
Specifies whether or not to generate the default pre-join path for tables of the XML data source.

Related topics:

- Setting up an XML connection

XML Option dialog

The dialog appears when you set the Default Format for Viewing Report property of a report tab or a web report to XML and then click ... in the value cell in the Report Inspector. It helps you to specify the parameters for running a report in XML format.



The following are details about options in the dialog:

Only Data

- If checked, the exported XML file will only contain the database column information. The exported XML schema file will only contain the structure information of the report.
- If unchecked, the exported XML file will also contain elements controlled by formulas, and the exported XML schema file will contain all the detailed information from the report, including all the property values of each report object.

Run Linked Report

If the report is linked with other reports, you can check whether or not to generate the linked reports in the XML result. If you are only interested in the primary report, leave this option unchecked. Generating linked reports at the same time, especially when the linked reports contain a large amount of data, will cause performance issue.

OK

Applies the changes and closes the dialog.

Cancel

Cancels the changes and exits the dialog.

Help

Displays the help document about this feature.

Formula Reference

A formula is a symbolic statement of the manipulations you want to perform on certain data. JReport provides a number of built-in functions for you to build your formula. Also it provides user defined formula (UDF) functions that allow you to design your own formula functions.

Generally, formulas can be used:

- to calculate information you cannot obtain directly from database data fields,
- to compare data,
- to join text with data,
- to convert data from one form to another,
- to enhance the formatting options with text strings, and
- to do a number of other things to customize your reports.

This chapter contains the following topics:

- Formula syntax
- Data type
- Built-in functions
- Operators

Formula syntax

JReport formula syntax is a syntax with simple flow control and other statements. The Formula Editor is where you write your formula statements. You should follow JReport formula syntax when writing formulas.

In this syntax, sequence and selection control structure are supported. You can use it to control the execution of the statements. In this syntax, a parameter can be used as variable, and its value can be used in computation or making decisions. Since the value of the parameter is input before the report is run, the report can respond to user input and generate different results.

Statements

The statement is the smallest executable unit in the syntax. A SQL procedure is a sequence of statements. Statements are separated by semicolon (;). There are many kinds of statements, such as declare statement, expression statement, assignment statement, if-else statement, and return statement.

Declare statement

The declare statement is used to declare variables.

- **Syntax**: `VariableType VariableName;`

  Example: `String sql;`

  This statement declares a local String type variable called sql.

- **Syntax**: `global VariableType VariableName;`

  Example: `global integer i;`

  This statement declares a global Integer type variable called i.

Expression statement

The expression statement is used to calculate the value of an expression.

Assignment statement

The assignment statement is used to calculate the expression on the right side of assignment operator and assign the value to the variable on the left side.

Syntax: `Variable = Expression`

The type of value of expression must be compatible with the type of the variable.

Example: `total = amount * price`

In this example, where the total, amount and price are declared variable. The statement calculate the "amount * price" and assign the value to "total".

If - else statement

The if-else statement provides conditional control of an action. Normally, the statements are executed one by one sequentially. The if-else statement enables alternative actions that depend on the evaluation result of the expression.

Syntax:

```
if (expression) {
  statement(s);}
or
if (expression) {
  statement(s);}
else {
  statement(s);}
```

Where, the expression must be a logical expression and have a Boolean value. The if clause checks the condition presented by the logical expression. If the condition is true, the statement following the if will be executed. Otherwise, the statement following the else will be executed.

Example:

```
if (score >= 60) {
    return "PASS";}
else {
    return "FAIL";}
```

In this example, if the score is greater than or equal to 60, the result is pass, otherwise the result is fail.

Return statement

The return statement is used to stop the execution of procedure and return a value if necessary. The return statement is the last statement executed.

Syntax: `return or return expression;`

The expression will be calculated before the procedure returns.

Example: `return results;`

For statement

The for statement provides a compact way to iterate over a range of values.

Syntax:

```
for (initialization; termination; increment) {
  statement;
};
```

Or

```
for (initialization; termination; increment) statement;
```

The initialization is an expression that initializes the loop - it's executed once at the beginning of the loop. The termination expression determines when to terminate the loop. This expression is evaluated at the top of each iteration of the loop. When the expression evaluates to false, the loop terminates. Finally, increment is an expression that gets invoked after each iteration through the loop. All these components are optional.

Examples:

```
integer i=0, j=0;
For(i=0;i<100;i++){
  j=j+1;
};
return j;
```

Or

```
integer i=0, j=0;
For(i=0;i<100;i++) j=j+1;
return j;
```

Both examples will return 100 as the result.

```
integer siteint=0;
string site="&_isMultiple_jrs.param$Site+true";

if(@P_int==0) then
  siteint=siteint+"&jrs.param$Site+%07"
else {
  for(siteint=0;siteint<@P_int;siteint=siteint+1)
  siteint= siteint+"&jrs.param$Site+" + @P_int
};
return site;
```

While statement

The while statement is used to continually execute a block of statements while a condition remains true.

Syntax:

```
while (expression) do {
  statement;
};
```

Or

```
while(expression) do statement;
```

First, the while statement evaluates expression, which must return a Boolean value. If the expression returns true, then the while statement executes the statement(s) associated with it. The while statement continues testing the expression and executing its block until the expression returns false.

Example:

```
integer i=0;
While(i<100) do {
  i=i+1;
};
return i;
```

Or

```
integer i=0;
While(i<100) do i=i+1;
return i;
```

Both examples will return 100 as the result.

Note: JReport Designer provides another statement that is similar to the while statement - the do-while statement.

syntax:

```
do { statement(s); } while (expression);
```

Or

```
do statement while(expression);
```

Instead of evaluating the expression at the top of the loop, do-while evaluates the expression at the bottom. Thus the statements associated with a do-while are executed at least once.

Example:

```
integer j=0;
Do {
   j=j+1;}
while(j<100);
return j;
```

Or

```
integer j=0;
Do j=j+1 while(j<100);
return j;
```

Both examples will return 100 as the result.

Select statement

The select statement is usually used in the case that the value of a single variable may determine one of a number of different choices.

A select statement is given a variable and compares its value to all cases in the switch; if there is a case that matches the value, all statements in the matching case are executed. If none match, and a default is given, all statements following the default keyword are executed.

Syntax:

```
select (variable name){
  case expression_1: statement
  case expression_2: statement
  ...
  default: statement
};
```

The expression_1 and expression_2 should be variables or constants. The statement in each should be a single statement or multi-statement (compound statement). When multi-statement is used, they must be enclosed by {}.

Notes:

- The value of the select is only evaluated one time even when a case modifies the tested value.
- No break statement is required, the logic flow does not continue through the following case statements.

Examples:

```
integer i = 0 ; string j = 'a';
i = @"Customers_Customer ID";
select(i) {
    case 1: j='bb'
    case 2: j='cc'
    default: j ='dd'
};
return j;


string state="",result="";
if (IsNull(@Customer_State)) then
  state="Others"
else
  state = @Customer_State;

select (state) {
    case  "BC","ID","MT","WA":result="Northwest"
    case  "CA","AZ","TX","NM":result="Southwest"
    case  "ME","MA","NH","NY":result="Northeast"
    case  "FL","NC","GA","SC":result="Southeast"
    default: result="Rest of Country"
};
return result;
```

Expression

The expression is a combination of values, operators and functions that produce a result. The value in the expression can be a literal value or a variable, while the operator defines the operation between values; the function is used to perform an action and return a value.

Value

The value specified in an expression can be a literal value or a variable value. JReport formula syntax supports seven data types of value. See the section Data type for details.

Literal value

The Literal Value is a value that is used exactly as it is displayed. A literal value represents a value such as a Number, String, or Date. For example, "Name", 98.6...

Variables

A variable is a named storage unit used to store a value. The identifier (name) of variable can be used to refer to the value of the variable or refer to the storage space of the variable. In an expression, the identifier is used to refer to the value, and in assignment statement, the identifier on the left is used to refer to the storage space.

- **Undeclared variables**

  The undeclared variables include parameter, DBField, special field and summary defined in the catalog. They can be used as a variable in a formula.

- **Declared variables**

  Another type of variable which must be declared in a declare statement before use. The declared variable is a real variable and can be assigned in the assignment statement. There are two kinds of declared variables:

  1. **Local Variable**: Variable which is only valid in the formula where it is declared.
  2. **Global Variable**: Variable which is valid in all the formulas once it is declared.

Report level global variable

To control the formula more flexibly, JReport offers you one more global variable scope to operate besides component level global variable. That is report level global variable. You can use "report" as the keyword to define the global variable in a formula.

For the two levels of global variable, the following definitions will help you distinguish them:

- Component level: Its lifetime is the same as the component's lifetime. In its lifetime, whenever a formula is executed, the variable will be executed if it is used in the formula.
- Report level: Its lifetime is the same as the report's (primary report) lifetime. Even in different components (using same data source), the same report level global variable will have only one instance. The report level variable can be passed to different components (using same data source). The components include banded objects, tables, charts, tabulars, crosstabs, and subreports (including many layers of subreports).

When using the keyword "global" to define a global variable in a formula, if you add no words before the keyword, the global variable is on the component level; if you add "report" before the keyword, the global variable will be a report level global variable.

Example

This example uses the report global variable to do a calculation in the report. First, create three formulas based on the report global variable as follows:

Formula1: report global number a =0;

Formula2: a = a+1;

Formula3: return a;

Then use them in one report including two BandedObjects(BandedObject1 and BandedObject2) in the same dataset as follows:

1. Insert Formula1, Formula2, Formula3 to the BandedObject1(15 records) as follows:

   BandedHeader:Formula1

   DetailPanel:Formula2

   BandedFooter:Formula3 - returns 15.

2. Then insert Formula2 and Formula3 to the BandedObject2(25 records) as follows:

   DetailPanel:Formula2

   BandedFooter:Formula3 - returns 40.

Notes:

- For nested data containers, report level global formulas may lead to an unexpected result where the formula is not calculated until the end of the group or report. If you add PageNumber in the formula's definition, the calculation will be in sequential order as viewed in the template and the generated result will be the one expected. For more information, see Formula levels.
- Although one report can have two or more data sources, report level global variable cannot span across data sources, otherwise the formula will not run correctly, thus, even the report global variables with the same variable name in different data sources are different in one report.
- If there is a report level global variable report global integer g, and it is used both in the primary report and subreport, then this variable g can communicate with each other.
- If you want to calculate the record number of the subreport and primary report separately, you can define a different report level variable from the primary report, but if one subreport is inserted into one primary report more than once you will be unable to correctly use the variable separately, in such circumstance, you can change the report level variable into component level variable to solve the problem.

Operator

An operator is a symbol that indicates an operation to be performed on values. The JReport procedure language has several classes of operators, including math, comparison and logical operators. For detailed information about operator, see the section Operators in this chapter.

Function

Functions return a value as its result. JReport formula syntax provides many built-in functions. You can refer to the section Built-in functions for detailed information about function.

Data type

Type				Description
Numeric	**Integer**	**int**	32-bit two's complement	Defines a field containing a whole number between -2,147,483,648 and 2,147,483,647. Fractional values inserted into an Integer field are truncated. If specified as NOT NULL WITH DEFAULT, null values will be replaced by zero.
		long	64-bit two's complement	A data type that holds large integers.
		BigInt		Immutable arbitrary-precision integers.
	Real Number	**float**	32-bit IEEE 754	This data type defines an 32bit-IEEE standard 754 floating-point field. Floating-point fields can be defined as single or double (default) precision based on the value of Integer. If the value of integer is between 1 and 21, the type is single precision floating-point, which requires 4 bytes of storage. If the value of integer is between 22 and 53 inclusive, the type is double precision floating-point, which requires 8 bytes of storage. If a field is specified as NOT NULL WITH DEFAULT, null values will be replaced by zero.
		double	64-bit IEEE 754	This data type defines an 64-bit IEEE standard 754 floating-point field. Floating-point fields can be defined as single or double (default) precision based on the value of integer. If the value of integer is between 22 and 53 inclusive, the type is double precision floating-point, which requires 8 bytes of storage. If a field is specified as NOT NULL WITH DEFAULT, null values will be replaced by zero.
		currency		This data type contains a dollar amount between; -10^{12} and 10^{12} and is displayed in a user-defined format.

Type			Description
DateTime	**Date**	MM/dd/yy MM/dd/yyyy	Dates designate a point in time according to the Gregorian calendar. Historical dates may not follow this calendar. The standard input format for this data type is MM/dd/yy or MM/dd/yyyy (automatic conversion is performed between these two formats).
	Time	HH:mm:ss hh:mm:ss	The Time field data type is defined by the standard input format of HH:mm:ss (using hour in day 0--23) or hh:mm:ss (using in am/pm 1--12).
	TimeStamp	yyyy-MM-dd hh:mm:ss	This combination Date and Time data type is displayed in the format yyyy-MM-dd-HH: mm.ss.nnnnnnn.
Other Types	**Char**	16-bit Unicode character	A single character.
	String		A data type that holds character information.
	Boolean	true or false	True or false.

Notes:

Format	Description
yyyy	An Integer from 0001 to 9999, representing a year.
MM	An Integer from 1 to 12 representing a month
dd	An Integer from 1 to 31 (maximum depends on the month and year) representing the day of the month.
HH	An Integer from 0 to 23, representing the hour in day.
hh	An Integer from 1 to 12, representing the hour in am/pm.
mm	An Integer from 0 to 59, representing minutes.
ss	An Integer from 0 to 59 representing seconds (default 0).
nnnnnn	An Integer (up to 6 digits) representing microseconds. If any trailing digit is omitted, 0 is assumed.

Built-in functions

The Functions panel in the Formula Editor lists all the built-in functions JReport provides. They are powerful and commonly used. If you are still puzzled in finding your own required function, you can write user defined formula functions.

The following are types of the built-in functions:

- Array
- Date Range
- Date/Time
- Financial
- Math
- String
- Others

Array

The Array functions are all used to summarize field data in several different ways.

- Any()
- Average()
- Count()
- DistinctCount()
- Every()
- Maximum()
- Minimum()
- PopulationStdDev()
- PopulationVariance()
- StdDev()
- Sum()
- UBound()
- Variance()

Any()

Function

- Any(array_variable);
- Any(field_variable);

Description

This function is used to check whether there is at least one element in the argument is true. If there is, return true; if there isn't, return false.

Arguments

- **field_variable**
  field_variable is a special variable in formula. It refers to a group of values of a database field or of a formula field.
- **array_variable**
  array_variable refers to an array.

Return type

Returns a Boolean value.

Examples

- `Boolean x = Any([false, false, false, true, false, false])`
- `Boolean x = Any(@isExcellent)`
- `Boolean x = Any(@formula1)` (@formula is a formula which returns a Boolean array.)
- `Boolean x = Any(@@ctf1)` (@@ctf1 is a crosstab formula which returns a Boolean array.)

Average()

Function

- Average(array_variable);
- Average(field_variable);
- Average(field_variable, groupby);

Description

This function is used to return the average value of a group of values referred to by the first argument. The second argument, if there is one, returns the group by field name.

Arguments

- **field_variable**
  field_variable is a special variable in Formula. It refers to a group of values of a database field or of a formula field. Whatever source they come from, database or formula, they should be of Number or Currency data type.
- **array_variable**
  array_variable refers to an array whose element is of the Number or Currency value.
- **groupby**
  groupby can be a constant String which indicates the groupby field name or a field variable on which grouping occurs, or a parameter variable that can accept a different value before each execution of a report.

Return type

The result type relies on the first argument. It can be of the Number or Currency data type, if the data type of each element in the first argument is also of the Number or Currency data type respectively.

Examples

- `Number x = Average([1.0, 2.0, 3.0, 4.5, 45.0, 67.0])`
- `Number x[6] = [1.0, 2.0, 3.0, 4.5, 45.0, 67.0]; Number y = Average(x)`
- `Currency x = Average([$1, $2, $3.0, $4.5, $45, $67])`
- `Currency x[] = [$1, $2, $3.0, $4.5, $45, $67]; Currency y = Average(x)`
- `Number x = Average(@dbfield)`
- `Number x = Average(@formula)`
- `Number x = Average(@dbfield, "group_field")`
- `Number x = Average(@dbfield, @"group_field")`
- `Number x = Average(@dbfield, @parameter)`
- `Currency x = Average(@dbfield)`
- `Currency x = Average(@formula)`
- `Currency x = Average(@formula, "group_field")`
- `Currency x = Average(@formula, @"group_field")`
- `Currency x = Average(@formula, @parameter)`

Note: If a field_variable is assigned to a variable, for example X, the variable X will lose the characteristic of representing a group of values. The following formula will then be treated as incorrect.

```
Number x = @dbfield;
Number y = Average(x); // system will prompt you that there is no such kind of function.
```

Count()

Function

- Count(array_variable);
- Count(field_variable);
- Count(field_variable, groupby);

Description

This function is used to count the number of a group of values referred to by the first argument. The second argument, if there is one, returns the group by field name.

Arguments

- **field_variable**
  field_variable is a special variable in Formula. It refers to a group of values of a database field or of a formula field.
- **array_variable**
  array_variable refers to an array.
- **groupby**
  groupby can be a constant String to indicate the groupby field name, or a field variable on which the grouping occurs, or a parameter variable that can accept a different value before each execution of a report.

Return type

Returns a Number value.

Examples

- `Number x = Count([1.0, 2.0, 3.0, 4.5, 45.0, 67.0]).`
- `Number x[6] = [1.0, 2.0, 3.0, 4.5, 45.0, 67.0];`
  `Number y = Count(x).`
- `Number x = Count(["str1","str2","str3","str4","str5","str6"]).`
- `Number x = Count(@dbfield).`
- `Number x = Count(@formula).`
- `Number x = Count(@dbfield, "group_field").`
- `Number x = Count(@formula, @"group_field").`
- `Number x = Count(@dbfield, @parameter).`

Note: If a field_variable is assigned to a variable, for example X, the variable X will lose the characteristic of representing a group of values. The following formula will then be treated as in correct.

```
Number x = @dbfield;
Number y = Count(x); // system will prompt you that there is no such kind of function.
```

DistinctCount()

Function

- DistinctCount(array_variable);
- DistinctCount(field_variable);
- DistinctCount(field_variable, groupby);

Description

This function is used to count the number of distinct values referred to by the first argument together. The second argument, if there is one, gives the group by field name.

Arguments

- **field_variable**
  field_variable is a special variable in Formula, and refers to a group of values from a database or formula field.
- **array_variable**
  array_variable refers to an array whose element should be a Number.
- **groupby**
  groupby can be a constant String to indicate the groupby field name, or a field variable on which grouping occurs, or a parameter variable that can accept a different value before each execution of a report.

Return type

Returns a Number value.

Examples

- `Number x = DistinctCount([1.0, 1.0, 3.0, 4.5, 4.5, 67.0]) // result is 4`
- `Number x[6] = [1.0, 1.0, 3.0, 4.5, 4.5, 67.0];`
  `Number y = DistinctCount(x)`
- `Number x = DistinctCount(["str1","str1","str3","str4","str4","str6"])}`
- `Number x = DistinctCount(@dbfield)`
- `Number x = DistinctCount(@formula)`
- `Number x = DistinctCount(@dbfield, "group_field")`
- `Number x = DistinctCount(@formula, @"group_field")`
- `Number x = DistinctCount(@dbfield, @parameter)`

Note: If a field_variable is assigned to a variable, for example X, the variable X will lose the characteristic of representing a group of values. The following formula will then be treated as incorrect.

```
Number x = @dbfield;
Number y = DistinctCount(x); // system will prompt you that there is no such kind of function.
```

Every()

Function

- Every(array_variable);
- Every(field_variable);

Description

This function is used to check whether each element in the argument is true. If yes, return true; if not, return false.

Arguments

- **field_variable**
  field_variable is a special variable in formula. It refers to a group of values of a database field or of a formula field.
- **array_variable**
  array_variable refers to an array.

Return type

Returns a Boolean value.

Examples

- `Boolean x = Every([false, false, false, true, false, false])`
- `Boolean x = Every(@isExcellent)`
- `Boolean x = Every(@formula1)` (@formula is a formula which returns a Boolean array.)
- `Boolean x = Every(@@ctf1)` (@@ctf1 is a crosstab formula which returns a Boolean array.)

Maximum()

Function

- Maximum(array_variable);
- Maximum(field_variable);
- Maximum(field_variable, groupby);

Description

This function is used to pick up the maximum value from a group of values referred to by the first argument. The groupby argument specifies the group by field name.

Arguments

- **field_variable**
  field_variable is a special variable in Formula, and refers to a group of values from a database or formula field.
- **array_variable**
  array_variable refers to an array whose element should be Number.
- **groupby**
  Groupby can be a constant String to indicate the groupby field name, or a field variable on which grouping occurs, or a parameter variable that can accept a different value before each execution of a report.

Return type

The result type relies on the first argument. It may be Number, Currency, String, Date, Time, DateTime, or Boolean if the datatype of each element in the first argument is Number, Currency, String, Date, Time, DateTime or Boolean respectively.

Examples

- `Number x = Maximum([1.0, 2.0, 3.0, 4.5, 45.0, 67.0])`
- `Number x[6] = [1.0, 2.0, 3.0, 4.5, 45.0, 67.0];`
  `Number y = Maximum(x)`
- `Number x = Maximum(@dbfield)`
- `Date x = Maximum(@formula)`
- `Time x = Maximum(@dbfield, "group_field")`
- `DateTime x = Maximum(@dbfield, @"group_field")`
- `String x = Maximum(@dbfield, @parameter)`
- `Currency x = Maximum(@dbfield)`
- `Boolean x = Maximum(@formula)`
- `Date x = Maximum(@formula, "group_field")`
- `Currency x = Maximum(@formula, @"group_field")`
- `Currency x = Maximum(@formula, @parameter)`
- `String x = Maximum(['1', '2', '3.0', '4.5', '45', '67'])`
- `Date x = Maximum([toDate(1998, 1, 1), toDate(1997, 7, 1), toDate(1999, 1, 16)])`

Note: If a field_variable is assigned to another variable, for example X, the variable X will lose the characteristic of representing a group of values. The following formula will then be treated as incorrect.

```
Number x = @dbfield;
Number y = Maximum(x); // system will prompt you that there is no such kind of function.
```

Minimum()

Function

- Minimum(array_variable);
- Minimum(field_variable);
- Minimum(field_variable, groupby);

Description

This function is used to pick up the minimum value from a group of values referred to by the first argument together. The second argument, if there is one, gives the group by field name.

Arguments

- **field_variable**
  field_variable is a special variable in Formula, and refers to a group of values from a database or formula field.
- **array_variable**
  array_variable refers to an array whose element should be a Number.
- **groupby**
  groupby can be a constant String to indicate the groupby field name, or a field variable on which grouping occurs, or a parameter variable that can accept a different value before each execution of a report.

Return type

The result type relies on the first argument. It may be Number, Currency, String, Date, Time, DateTime, or Boolean if the data type of each element in the first argument is Number, Currency, String, Date, Time, DateTime or Boolean respectively.

Examples

- `Number x = Minimum([1.0, 2.0, 3.0, 4.5, 45.0, 67.0])`
- `Number x[6] = [1.0, 2.0, 3.0, 4.5, 45.0, 67.0];`
  `Number y = Minimum(x)`
- `Number x = Minimum(@dbfield)`
- `Date x = Minimum(@formula)`
- `Time x = Minimum(@dbfield, "group_field")`
- `DateTime x = Minimum(@dbfield, @"group_field")`
- `String x = Minimum(@dbfield, @parameter)`
- `Currency x = Minimum(@dbfield)`
- `Boolean x = Minimum(@formula)`
- `Date x = Minimum(@formula, "group_field")`
- `Currency x = Minimum(@formula, @"group_field")`
- `Currency x = Minimum(@formula, @parameter)`
- `String x = Minimum(['1', '2', '3.0', '4.5', '45', '67'])`
- `Date x = Minimum([toDate(1998, 1, 1), toDate(1997, 7, 1), toDate(1999, 1, 16)]`

Note: If a field_variable is assigned to a variable, for example X, the variable X will lose the characteristic of representing a group of values. The following formula will then be treated as incorrect.

```
Number x = @dbfield;
Number y = Minimum(x); // system will prompt you that there is no such kind of function.
```

PopulationStdDev()

Function

- PopulationStdDev(array_variable);
- PopulationStdDev(field_variable);
- PopulationStdDev(field_variable, groupby);

Description

This function is used to find the population standard deviation of a group of values referred to by the first argument. The second argument, if there is one, gives the group by field name.

Arguments

- **field_variable**
  field_variable is a special variable in Formula, and refers to a group of values from a database or formula field. Whatever kind of source the data comes from, database or formula, it should be of the Number data type.
- **array_variable**
  array_variable refers to an array whose element should be Number value.
- **groupby**
  groupby can be a constant String to indicate the groupby field name, or a field variable on which grouping occurs, or a parameter variable that can accept a different value before each execution of a report.

Return type

Returns a Number value.

Examples

- `Number x = PopulationStdDev([1.0, 2.0, 3.0, 4.5, 45.0, 67.0])`
- `Number x[6] = [1.0, 2.0, 3.0, 4.5, 45.0, 67.0];`
  `Number y = PopulationStdDev(x)`
- `Number x = PopulationStdDev(@dbfield)`
- `Number x = PopulationStdDev(@formula)`
- `Number x = PopulationStdDev(@dbfield, "group_field")`
- `Number x = PopulationStdDev(@dbfield, @"group_field")`
- `Number x = PopulationStdDev(@dbfield, @parameter)`

Note: If a field_variable is assigned to a variable, for example X, the variable X will lose the characteristic of representing a group of values. The following formula will then be treated as incorrect.

```
Number x = @dbfield;
Number y = PopulationStdDev(x); // system will prompt you that there is no such kind of function.
```

PopulationVariance()

Function

- PopulationVariance(array_variable);
- PopulationVariance(field_variable);
- PopulationVariance(field_variable, groupby);

Description

This function is used to find the population variance of a group of values referred to be the first argument. The second argument, if there is one, gives the group by field name.

Arguments

- **field_variable**
  field_variable is a special variable in Formula, and refers to a group of values from a database or formula field. Whatever kind of source the data comes from, database or formula, it should be of the Number data type.
- **array_variable**
  array_variable refers to an array whose element should be Number value.
- **groupby**
  groupby can be a constant String to indicate the groupby field name, or a field variable on which grouping occurs, or a parameter variable that can accept a different value before each execution of a report.

Return type

Returns a Number value.

Examples

- `Number x = PopulationVariance([1.0, 2.0, 3.0, 4.5, 45.0, 67.0])`
- `Number x[6] = [1.0, 2.0, 3.0, 4.5, 45.0, 67.0];`
  `Number y = PopulationVariance(x)`
- `Number x = PopulationVariance(@dbfield)`
- `Number x = PopulationVariance(@formula)`
- `Number x = PopulationVariance(@dbfield, "group_field")`
- `Number x = PopulationVariance(@dbfield, @"group_field")`
- `Number x = PopulationVariance(@dbfield, @parameter)`

Note: If a field_variable is assigned to a variable, for example X, the variable X will lose the characteristic of representing a group of values. The following formula will then be treated as incorrect.

```
Number x = @dbfield;
Number y = PopulationVariance(x); // system will prompt you that there is no such kind of function.
```

StdDev()

Function

- StdDev(array_variable);
- StdDev(field_variable);
- StdDev(field_variable, groupby);

Description

This function is used to find the standard deviation of a group of values referred to by the first argument. The second argument, if there is one, gives the group by field name.

Arguments

- **field_variable**
  field_variable is a special variable in Formula, and refers to a group of values from a database or formula field. Whatever kind of source the data comes from, database or formula, it should be of the Number data type.
- **array_variable**
  array_variable refers to an array whose element should be a Number value.
- **groupby**
  groupby can be a constant String to indicate the groupby field name, or a field variable on which grouping occurs, or a parameter variable that can accept a different value before each execution of a report.

Return type

Returns a Number value.

Examples

- `Number x = StdDev([1.0, 2.0, 3.0, 4.5, 45.0, 67.0])`
- `Number x[6] = [1.0, 2.0, 3.0, 4.5, 45.0, 67.0];`
  `Number y = StdDev(x)`
- `Number x = StdDev(@dbfield)`
- `Number x = StdDev(@formula)`
- `Number x = StdDev(@dbfield, "group_field")`
- `Number x = StdDev(@dbfield, @"group_field")`
- `Number x = StdDev(@dbfield, @parameter)`

Note: If a field_variable is assigned to a variable, for example X, the variable X will lose the characteristic of representing a group of values. The following formula will then be treated as incorrect.

```
Number x = @dbfield;
Number y = StdDev(x); // system will prompt you that there is no such kind of function.
```

Sum()

Function

- Sum(array_variable);
- Sum(field_variable);
- Sum(field_variable, groupby);

Description

This function is used to add a group of values (referred to by the first argument) together. The second argument, if there is one, gives the group by field name.

Arguments

- **field_variable**
  field_variable is a special variable in Formula, and refers to a group of values from a database or formula field. Whatever kind of source the data comes from, database or formula, it should be of the Number or Currency data type.
- **array_variable**
  array_variable refers to an array whose element should be Number or Currency value.
- **groupby**
  groupby may be a constant String to indicate the groupby field name, or a field variable on which grouping occurs, or a parameter variable that can accept a different value before each execution of a report.

Return type

The result type relies on the first argument. It may be Number or Currency if the data type of each element in the first argument is Number or Currency respectively.

Examples

- `Number x = Sum([1.0, 2.0, 3.0, 4.5, 45.0, 67.0])`
- `Number x[6] = [1.0, 2.0, 3.0, 4.5, 45.0, 67.0];`
  `Number y = Sum(x)`
- `Currency x = Sum([$1, $2, $3.0, $4.5, $45, $67])`
- `Currency x[] = [$1, $2, $3.0, $4.5, $45, $67];`
  `Currency y = Sum(x)`
- `Number x = Sum(@dbfield)`
- `Number x = Sum(@formula)`
- `Number x = Sum(@dbfield, "group_field")`
- `Number x = Sum(@dbfield, @"group_field")`
- `Number x = Sum(@dbfield, @parameter)`
- `Currency x = Sum(@dbfield)`
- `Currency x = Sum(@formula)`
- `Currency x = Sum(@formula, "group_field")`
- `Currency x = Sum(@formula, @"group_field")`
- `Currency x = Sum(@formula, @parameter)`

Note: If a field_variable is assigned to a variable, for example X, the variable X will lose the characteristic of representing a group of values. The following formula will then be treated as incorrect.

```
Number x = @dbfield;
Number y = Sum(x); // system will prompt you that there is no such kind of function.
```

UBound()

Function

- UBound(array_variable);

Description

This function is used to return the number of elements in an array.

Argument

- **array_variable**
  array_variable refers to an Array value, expression or variable.

Return value

Returns a Number value.

Examples

- The return value of the following statement is 5.

  ```
  integer a[] = [1,1,1,1,1];
  UBound(a)
  ```

- The return value of the following statement is 6.

  `UBound([$1, $2, $3.0, $4.5, $45, $67]).`

- ```
  if(isNull(@"Customer Name"))then
  {
  return "No Data";
  }
  else{
  string s[]=StringSplit(@"Customer Name");
  string s1="Java";
  string s2[]=filter(s,s1);
  if (UBound(s2)!=0) then
  return right(s2[0],4)
  else
  return "No matching data ";
  }
  ```

  This formula searchs for the customer names containing the word "Java".

Variance()

Function

- Variance(array_variable);
- Variance(field_variable);
- Variance(field_variable, groupby);

Description

This function is used to find the variance of a group of values referred to by the first argument. The second argument, if there is one, gives the group by field name.

Arguments

- **field_variable**
  field_variable is a special variable in Formula, and refers to a group of values from a database or formula field. Whatever kind of source the data comes from, database or formula, it should be of the Number data type.
- **array_variable**
  array_variable refers to an array whose element should be Number value.
- **groupby**
  groupby can be a constant String to indicate the groupby field name, or a field variable on which grouping occurs, or a parameter variable that can accept a different value before each execution of a report.

Return type

Returns a Number value.

Examples

- `Number x = Variance([1.0, 2.0, 3.0, 4.5, 45.0, 67.0])`
- `Number x[6] = [1.0, 2.0, 3.0, 4.5, 45.0, 67.0];`
  `Number y = Variance(x)`
- `Number x = Variance(@dbfield)`
- `Number x = Variance(@formula)`
- `Number x = Variance(@dbfield, "group_field")`
- `Number x = Variance(@dbfield, @"group_field")`
- `Number x = Variance(@dbfield, @parameter)`

Note: If a field_variable is assigned to a variable, for example X, the variable X will lose the characteristic of representing a group of values. The following formula will then be treated as incorrect.

```
Number x = @dbfield;
Number y = Variance(x); // system will prompt you that there is no such kind of function.
```
```

Date Range

Note that all the following functions cannot be used to return a value singly.

Aged0To30Days()
Aged31To60Days()
Aged61To90Days()
AllDatesFromToday()
AllDatesFromTomorrow()
AllDatesToToday()
AllDatesToYesterday()
Calendar1stHalf()
Calendar1stQtr()
Calendar2ndHalf()
Calendar2ndQtr()
Calendar3rdQtr()
Calendar4thQtr()
Last4WeeksToSun()
Last7Days()
LastFullMonth()
LastFullWeek()
LastYearMTD()
LastYearYTD()
MonthToDate()
Next30Days()
Next31To60Days()
Next61To90Days()
Next91To365Days()
Over90Days()
WeekToDateFromSun()
YearToDate()

Aged0To30Days()

This function returns a range of Data values that is within the previous 30 days from today. For example, if today is 10/14/99, the return range is from 09/15/99 to 10/14/99.

Example

The return value of the following statement is Not Expired.

```
 if (ToDate(1999,9,20) in Aged0To30Days())
   return "Not Expired"
 else
   return "Expired"
```

Aged31To60Days()

This function returns a range of Data values that is within the previous 31 to 60 days from today date. For example, if today is 10/13/99, the return range is from 08/12/99 to 09/13/99.

Example

If today is 10/13/99, the return value of the following statement is false.

```
return ToDate(1998, 4, 20) in Aged31To60Days()
```

Aged61To90Days()

This function returns a range of Data values that is within the previous 61 to 90 days from today date. For example, if today is 10/13/99, the return range is from 07/13/99 to 08/11/99.

Example

If today is 10/13/99, the return value of the following statement is false.

```
return ToDate(1998, 4, 20) in Aged61To90Days()
```

AllDatesFromToday()

This function returns a range of Data values that include any date from the present day to a date about 10000 years in the future Data value. For example, if today is 04/15/98, the return range is from 04/15/98 to the future.

Example

If today is 10/10/99, the return value of the following statement is false.

```
return ToDate(1998, 4, 20) in AllDatesFromToday()
```

AllDatesFromTomorrow()

This function returns a range of Data values that include any date from tomorrow to any future Data value. For example, if today is 04/15/98, the return range is from 04/16/98 to the future.

Example

If today is 10/10/99, the return of the following statement is false.

```
return ToDate(1998, 4, 20) in AllDatesFromTomorrow()
```

AllDatesToToday()

This function returns a range of Data values that include all dates up through the present day. For example, if today is 4/15/98, the return range is from the past to 04/15/1998.

Example

If today is 10/10/99, the return value of the following statement is true.

```
return ToDate(1998, 4, 20) in AllDatesToToday()
```

AllDatesToYesterday()

This function returns a range of Data values that include all dates up through the previous day. For example, if today is 4/15/98, the return range is from the past to 04/14/1998.

Example

If today is 10/10/99, the return value of the following statement is true.

```
return ToDate(1998, 4, 20) in AllDatesToYesterday()
```

Calendar1stHalf()

This function returns a range of Data values that include all dates from January 1st through June 30th.

Example

If today is 10/10/99, the return value of the following statement is false.

```
return Today() in Calendar1stHalf()
```

Calendar1stQtr()

This function returns a range of Data values that include all dates from January 1st through to March 31st.

Example

If today is 10/10/99, the return value of the following statement is false.

```
return Today() in Calendar1stQtr()
```

Calendar2ndHalf()

This function returns a range of Data values that include all dates from July 1st through to December 31st.

Example

If today is 10/10/99, the return value of the following statement is true.

```
return Today() in Calendar2ndHalf()
```

Calendar2ndQtr()

This function returns a range of Data values that include all dates from April 1st through to June 30th.

Example

If today is 10/10/99, the return value of the following statement is false.

```
return Today() in Calendar2ndQtr()
```

Calendar3rdQtr()

This function returns a range of Data values that include all dates from July 1st through to September 30th.

Example

If today is 10/10/99, the return value of the following statement is false.

```
return Today() in Calendar3rdQtr()
```

Calendar4thQtr()

This function returns a range of Data values that include all dates from October 1st through December 31st.

Example

If today is 10/10/99, the return value of the following statement is true.

```
return Today() in Calendar4thQtr()
```

Last4WeeksToSun()

This function returns a range of Data values that include the four weeks previous to last Sunday. For example, if today is 4/15/95, the return range is from 3/13/95 to 4/9/95.

Example

If today is 10/10/99, the return value of the following statement is false.

```
return ToDate(1999, 4, 20) in Last4WeeksToSun()
```

Last7Days()

This function returns a range of Data values that include all dates from seven days ago to today (including today). For example, if today is 04/15/98, the return range is from (include) 04/09/98 to 04/15/98.

Example

If today is 10/10/99, the return value of the following statement is false.

```
return ToDate(1999, 4, 20) in Last7Days()
```

LastFullMonth()

This function returns a range of Data values that include all dates from the first to the last day of the previous month. For example, if today is 04/15/98, the return range is from 03/01/98 to 03/31/98.

Example

If today is 10/10/99, the return value of the following statement is false.

```
return ToDate(1999, 4, 20) in LastFullMonth()
```

LastFullWeek()

This function returns a range of Data values that include all dates from the Sunday to the Saturday of the previous week. For example, if today is 04/10/95, the return range is from 04/02/95 to 04/08/95.

Example

If today is 10/10/99, the return value of the following statement is false.

```
return ToDate(1999, 4, 20) in LastFullWeek()
```

LastYearMTD()

This function returns a range of Data values that fall on any date in the current month last year, up to the current date last year. For example, if today is 04/15/98, the return range is from 04/01/97 to 04/15/97.

Example

If today is 10/10/99, the return value of the following statement is false.

```
return ToDate(1999, 4, 20) in LastYearMTD()
```

LastYearYTD()

This function returns a range of Data values that include all dates in the last year, up to the current date last year. For example, if today is 04/15/98, the return range is from 01/01/97 to 04/15/97

Example

If today is 10/10/99, the return value of the following statement is true.

```
return ToDate(1998, 4, 20) in LastYearYTD()
```

MonthToDate()

This function returns a range of Data values that include all dates from the first day of the month to today. For example, if today is 04/15/98, the return range is from 04/01/98 to 04/15/98.

Example

If today is 10/10/99, the return value of the following statement is true.

```
return ToDate(1999, 10, 2) in MonthToDate()
```

Next30Days()

This function returns a range of Data values that include all days in the next 30 days starting from today (including today). For example, if today is 04/15/98, the return range is from 04/15/98 to 05/15/98.

Example

If today is 10/10/99, the return value of the following statement is false.

```
return ToDate(1999, 10, 2) in Next30Days()
```

Next31To60Days()

This function returns a range of Data values that include all days in the next 31 to 60 days starting from today. For example, if today is 04/15/98, the return range is from 05/16/98 to 06/14/98.

Example

If today is 10/10/99, the return value of the following statement is false.

```
return ToDate(1999, 10, 2) in Next31To60Days()
```

Next61To90Days()

This function returns a range of Data values that include all days in the next 61 to 90 days starting from today. For example, if today is 04/15/98, the return range is from 06/15/98 to 07/14/98.

Example

If today is 10/10/99, the return value of the following statement is false.

```
return ToDate(1999, 4, 20) in Next61To90Days()
```

Next91To365Days()

This function Returns a range of Data values that include all days in the next 91 to 365 days starting from today. For example, if today is 04/15/98, the return range is from 07/15/98 to 04/15/99.

Example

If today is 10/10/99, the return value of the following statement is false.

```
return ToDate(1999, 4, 20) in Next91To365Days()
```

Over90Days()

This function returns a range of Data values that include all days that are more than 90 days older than the current date. For example, if today is 04/15/98, the return range is from the past to 01/14/98.

Example

If today is 10/10/99, the return value of the following statement is true.

```
return ToDate(1999, 4, 20) in Over90Days()
```

WeekToDateFromSun()

This function returns a range of Data values that include all days from last Sunday to today (including today). For example, if today is 04/13/95, the return range is from 04/09/95 to 04/13/95.

Example

If today is 10/10/99, the return value of the following statement is false.

```
return ToDate(1999, 4, 20) in WeekToDateFromSun()
```

YearToDate()

This function returns a range of Data values that include all days from the first day of the calendar year to today (including today). For example, if today is 04/15/98, the return range is from 01/01/98 to 04/15/98.

Example

If today is 10/10/99, the return value of the following statement is true.

```
return ToDate(1999, 4, 20) in YearToDate()
```

Date/Time

Date functions allow you to convert numbers to dates and dates to numbers.

- CurrentDate()
- CurrentDateTime()
- CurrentTime()
- DateAdd()
- DateDiff()
- DatePart()
- DateTimeTo2000()
- DateTimeToDate()
- DateTimeToSeconds()
- DateTimeToTime()
- DateTo2000()
- Day()
- DayOfWeek()
- DayOfYear()
- DTSTo2000()
- DTSToSeconds()
- FirstDayOfMonth()
- FirstDayOfQuarter()
- FirstDayOfWeek()
- FirstDayOfYear()
- FirstSundayOfMonth()
- FirstSundayOfYear()
- ForEachDay()
- ForEachHalfMonth()
- ForEachHalfYear()
- ForEachMonth()
- ForEachQuarter()
- ForEachWeek()
- ForEachYear()
- Hour()
- IsDate()
- IsDateTime()
- IsTime()
- Minute()
- Month()
- MonthName()
- Now()
- Quarter()
- Second()
- SelectedDate()
- Timer()
- ToDate()
- ToDateTime()
- Today()
- ToTime()
- WeekdayName()
- WeekFrom1970()
- WeekOfMonth()
- WeekOfYear()
- Year()

CurrentDate()

This function returns the current date on a report.

Return value

The return value is a date.

Example

If today is Oct. 6,1999, the return value of the following statement is 10/06/99.

```
CurrentDate()
```

CurrentDateTime()

This function returns the current date and time on a report.

Return value

The return value is a DateTime.

Example

If today is Oct.6,1999, the current time is 5:10:27 p.m., the return value of the following statement is 1999-10-06 05:10:27 PM.

```
CurrentDateTime()
```

CurrentTime()

This function returns the current time on a report.

Return value

The return value is a Time.

Example

If the current time is 5:10:27 p.m., the return value of the following statement is 05:10:27 PM.

```
CurrentTime()
```

DateAdd(intervalType, nIntervals, startDateTime)

Returns a DateTime value to which a specified number of time intervals have been added.

This function is used to add interval of time to a DateTime. Its main feature is that the DateTime returned will always be valid. For example, the function takes into account such factors as the number of days in a month and leap years. If you want to add or subtract days to a DateTime, you could use the addition and subtraction operators instead of the function with the "d" parameter. However, this function also handles other types of intervals such as adding months or hours to a DateTime.

Parameters

- intervalType - A String expression specifying the interval of time to be added.

IntervalType value	Description
yyyy	Year
q	Quarter (3-month period)
m	Month
y	Day of year
d	Day
w	Weekday
ww	Week (7-day period)
h	Hour
n	Minute
s	Second

- nIntervals - A Number or numeric expression specifying the number of intervals to be added. It can be positive (to get DateTimes from the future), or negative (to get DateTimes from the past).
- startDateTime - The DateTime value to which the intervals are to be added.

Return value

A DateTime value.

Examples

- `DateAdd("yyyy",1,CurrentDateTime())` - Returns the DateTime value for current datetime plus one year.
- `DateAdd("q",1,CurrentDateTime())` - Returns the DateTime value for current datetime plus three months.
- `DateAdd("m",1,CurrentDateTime())` - Returns the DateTime value for current datetime plus one month.
- `DateAdd("y",1,CurrentDateTime())` - Returns the DateTime value for current datetime plus one day.
- `DateAdd("d",1,CurrentDateTime())` - Returns the DateTime value for current datetime plus one day.
- `DateAdd("w",1,CurrentDateTime())` - Returns the DateTime value for current datetime plus one day.
- `DateAdd("ww",1,CurrentDateTime())` - Returns the DateTime value for current datetime plus seven days.
- `DateAdd("h",1,CurrentDateTime())` - Returns the DateTime value for current datetime plus one hour.
- `DateAdd("n",1,CurrentDateTime())` - Returns the DateTime value for current datetime plus one minute.
- `DateAdd("s",1,CurrentDateTime())` - Returns the DateTime value for current datetime plus one second.
- `DateAdd("d",-32,ToDateTime(1999,9,28))` - Returns the DateTime value 1999-8-27 00:00:00.
- `DateAdd("m",1,ToDateTime(1996,1,31))` - Returns the DateTime value 1996-2-29 00:00:00. Notice that DateAdd will not return the invalid value 1996-2-31 00:00:00.
- `DateAdd("q",17,ToDateTime(1999,9,28))` - Returns the DateTime value 2003-12-28 00:00:00.
- `DateAdd("h",-400,ToDateTime(1999,9,28))` - Returns the DateTime value 1999-9-11 08:00:00. In other words, this is the result of subtracting 400 hours from 1999-9-28 00:00:00.

Notes:

- To add days to a DateTime, you can use any of the interval type parameters "y", "d" or "w". They all have the same effect for DateAdd.
- DateAdd returns a DateTime value and not a Date value. However, you may need to convert this DateTime value to a Date value in certain situations (such as if you wanted to assign the value returned by DateAdd to a Date type variable). To convert to a Date value, use DateAdd in combination with the ToDate type conversion function. For example, the following returns the Date value for October 6, 1997.

  `ToDate(DateAdd("yyyy",-2,ToDateTime(1999,10,6)))`

DateDiff(intervalType, startDateTime, endDateTime, firstDayOfWeek)

Returns a number of time intervals between two specified dates.

Overload

- DateDiff(intervalType, startDateTime, endDateTime)
- DateDiff(intervalType, startDateTime, endDateTime, firstDayOfWeek)

Parameters

- intervalType - A string expression that is the interval of time you use to calculate the difference between startDateTime and endDateTime. Possible values can be:

IntervalType value	Description
Yyyy	Year
Q	Quarter
M	Month
Y	Day of year
D	Day (both "y" and "d" find the difference in days)
W	Number of weeks between startDateTime and endDateTime
Ww	Number of firstDayOfWeek's between startDateTime and endDateTime
H	Hour
N	Minute
S	Second

- startDateTime - The first DateTime value used in calculating the difference.
- endDateTime - The second DateTime value used in calculating the difference.
- firstDayOfWeek - An optional constant specifying the first day of the week. If not specified, jrSunday will be used.

Constant	Description
jrUseSystem	0 (Use the current system date)
jrSunday	1
jrMonday	2
jrTuesday	3
jrWednesday	4
jrThursday	5
jrFriday	6
jrSaturday	7

Return value

A Number value.

Examples

- Use DateDiff with the "d" or "y" interval type parameter to find the number of days between two dates.

  `DateDiff("d", ToDateTime(1999,10,7), ToDateTime(1999,10,10))` - Returns 3
- Use DateDiff with the "yyyy" interval type parameter to find the number of years difference between two dates. This use of DateDiff is the same as finding the difference between the year of endDateTime and the year of startDateTime.
  - `DateDiff("yyyy", ToDateTime(1999,10,7), ToDateTime(2005,2,10))` - Returns 6.
  - `DateDiff("yyyy", ToDateTime(1999,12,31), ToDateTime(2000,1,1))` - Returns 1 (a 1 year difference), even though there is only a 1 day difference between the dates.
  - `DateDiff("yyyy", ToDateTime(1999,1,1), ToDateTime(1999,12,31))` - Returns 0 (a 0 year difference), even though there is a 364 days difference.

  Suppose that for the above examples, the first date is the date that you bought a mutual fund, and the second date is the date you sold it. The mutual fund company must send you an annual report for every year in which you owned units in the fund. You would then get 7, 2 and 1 annual reports respectively, in the above cases.
- Use DateDiff with the "q" parameter to find the number of quarters (3 month periods) difference between two dates.
  - `DateDiff("q", ToDateTime(1999,10,6), ToDateTime(2003,5,20))` - Returns 14.
  - `DateDiff("q", ToDateTime(1999,3,31), ToDateTime(1999,4,1))` - Returns 1. The two dates are in adjacent quarters.
  - `DateDiff("q", ToDateTime(1999,1,1), ToDateTime(1999,3,31))` - Returns 0. The two dates are in the same quarter.

  Suppose the mutual fund company in the "yyyy" example mailed out quarterly reports. It would need to mail out 15, 2 and 1 quarterly reports respectively, in the above cases.
- Use DateDiff with the "m" parameter to find the number of months difference between two dates.

  `DateDiff("m", ToDateTime(1999,3,15), ToDateTime(1999,7,13))` - Returns 4.
- Use DateDiff with the "w" parameter to calculate the number of weeks between two dates. For example, if startDateTime is on a Tuesday, DateDiff counts the number of Tuesdays between startDateTime and endDateTime not including the initial Tuesday of startDateTime. Note however that it counts endDateTime if endDateTime is on a Tuesday.
  - `DateDiff("w", ToDateTime(1999,10,19), ToDateTime(1999,10,25))` - Returns 0.
  - `DateDiff("w", ToDateTime(1999,10,19), ToDateTime(1999, 10,26))` - Returns 1.

DatePart(intervalType, inputDateTime, firstDayOfWeek, firstWeekOfYear)

Returns a number that specifies a given part of a given date.

Overloads

- DatePart(intervalType, inputDateTime)
- DatePart(intervalType, inputDateTime, firstDayOfWeek)
- DatePart(intervalType, inputDateTime, firstDayOfWeek, firstWeekOfYear)
- DatePart(intervalType, inputTime)
- DatePart(intervalType, inputTime, firstDayOfWeek)
- DatePart(intervalType, inputTime, firstDayOfWeek, firstWeekOfYear)

Parameters

- intervalType - A String expression that specifies the part of a date to be returned. Possible values can be:

IntervalType value	Description
Yyyy	Extracts the year
Q	Quarter (the result is 1, 2, 3 or 4)
M	Month (the result is from 1 to 12)
Y	Day of year (1 to 365 or 366 in a leap year)
D	Day part of the date (1 to 31) Year
W	Day of week (1 to 7 with the result depending on firstDayOfWeek)
Ww	Week of year (1 to 53 with firstDayOfWeek and firstWeekOfYear determining the exact days of the first calendar week of the year)
H	Extracts the hour part of the given DateTime (0 to 23)
N	Minute part (0 to 59)
S	Second part (0 to 59)

- inputDateTime - The DateTime value whose part will be extracted.
- FirstDayOfWeek - An optional constant used to specify the first day of the week. If not specified, jrSunday will be used.

Constant	Description
jrUseSystem	0 (Use the current system date)
jrSunday	1
jrMonday	2
jrTuesday	3
jrWednesday	4
jrThursday	5
jrFriday	6
jrSaturday	7

- firstWeekOfYear - An optional constant specifying the first week of the year. If not specified, the first week is assumed to be the one in which Jan. 1 occurs (jrFirstJan1).

Constant	Value	Description
jrUseSystem	0	Start with week in which system date occurs
jrFirstJan1	1	Start with week in which January 1 occurs (default)
jrFirstFourDays	2	Start with the first week that has at least four days in the new year
jrFirstFullWeek	3	Start with first full week of the year

Examples

- `DatePart("d", ToDateTime(1999,8,15))` - Returns 15.
- `DatePart("m", ToDateTime(1999,8,15))` - Returns 8.
- `DatePart("n", ToDateTime(1999,8,15,10,35,0))` - Returns 35.
- `DatePart("q", ToDateTime(1999, 9,29))` - Returns 3 since September 29 is in the third quarter of the year.
- `DatePart("ww", ToDateTime(1997, 9,14))` - Returns 38 since 1997-9-14 is in the 38th week of 1997.

Notes:

- The DatePart function with "yyyy" intervalType parameter is the same as the Year function. Similarly, the DatePart function with "m", "d", "w", "h", "n" and "s" intervalType argument is the same as the functions Month, Day, Weekday (or DayOfWeek), Hour, Minute and Second respectively. On the other hand, there is no easy alternative to using the DatePart function for the "q", "y" and "ww" intervalType parameters.
- The firstDayOfWeek parameter affects the DatePart function when the interval type parameter is "w" or "ww". For all other intervalType parameter values, it is ignored.
- The firstWeekOfYear parameter affects the DatePart function only when the intervalType argument is "ww". For all other intervalType argument values, it is ignored.

DateTimeTo2000(Datetime, Number)

- **2 digit years(xx)**
  If the Year value is greater than the windowing number, 19 is appended before the 2 digits (19xx). If the Year value is less than or equal to the windowing number, 20 is appended (20xx).
- **4 digit years(19xx)**
  If the last two digits in the Year value are greater than the windowing number, the Year is retained as found in the Date field (19xx). If the two digits in the Year value are less than or equal to the windowing number, the first two digits are changed to 20 (20xx). If the first two digits in the year field are 20, the Year is retained as found in the Date field (20xx).

  Note: If the year is less than or equal to 1899, and greater than or equal to 100, there will be no change to the date.

Parameters

- DateTime - Accepts only valid DateTime fields, with either 2 digit or 4 digit years.
- Number - A number between 0 and 99 corresponding with the desired windowing year.

Return value

A Date field with a four digit year.

Examples

- Here the window value is greater than the year, and so the year is changed to 20XX.

  `DateTimeTo2000(ToDateTime(1998, 12, 12, 3, 2, 1), 99)` - Returns 2098-12-12 3:02:01.
- Here the window value is less than the year, and so it will not change the year.

  `DateTimeTo2000(ToDateTime(1995, 1, 2, 3, 2, 1), 94)` - Returns 1995-01-02 3:02:01.
- These are examples of dates that will not be affected by this function.
  - `DateTimeTo2000(ToDateTime(1899, 12, 12, 5, 6, 7), 99)` - Returns 1899-12-12 5:06:07.
  - `DateTimeTo2000(ToDateTime(999, 12, 12, 5, 6, 7), 99)` - Returns 999-12-12 5:06:07.
- Some databases maintain the year of a Date as a two-digit field. These samples simulate that Date field.
  - `DateTimeTo2000(ToDateTime(93, 12, 12, 5, 6, 7), 96)` - Returns 2093-12-12 5:06:07.
  - `DateTimeTo2000(ToDateTime(98, 12, 12, 5, 6, 7), 50)` - Returns 1998-12-12 5:06:07.

DateTimeToDate(Datetime a)

This function evaluates the specified TimeStamp value and returns only the date.

Parameter

a - A TimeStamp value.

Return value

The return value is a date.

Example

If the current date is Oct. 17, 1999 and the current time is 12:27:15, the return value of the following statement is 10/17/99.

```
DateTimeToDate(CurrentDateTime())
```

DateTimeToSeconds(Datetime a)

This function evaluates the specified value and converts the time to the number of seconds from 00:00:00 to the specified time.

Parameter

a - A TimeStamp value.

Return value

The return value is an Integer.

Example

If the current date is Oct. 17, 1999 and the current time is 12:27:15, the return value of the following statement is 44835.

```
DateTimeToSeconds(CurrentDateTime())
```

DateTimeToTime(Datetime a)

This function evaluates the specified TimeStamp value and returns only the time.

Parameter

a - A TimeStamp value.

Return value

The return value is a Time.

Example

If the current date is Oct. 17, 1999 and the current time is 12:27:15, the return value of the following statement is 12:27:15.

```
DateTimeToTime(CurrentDateTime())
```

DateTo2000(date, number)

- 2 digit years(xx)
  If the Year value is greater than the windowing number, 19 is appended before the 2 digits (19xx). If the Year value is less than or equal to the windowing number, 20 is appended (20xx).
- 4 digit years(19xx)
  If the last two digits in the Year value are greater than the windowing number, the Year is preserved as found in the date field (19xx). If the last two digits in the Year value are less than or equal to the windowing number, the first two digits are changed to 20 (20xx). If the first two digits in the year field are 20, the Year is preserved as found in the date field (20xx).

Parameters

- date - Accepts only valid Date fields, with either 2 digit or 4 digit years.
- number - A number between 0 and 99 corresponding with the desired windowing year.

Return value

A Date field with a four digit year.

Examples

- `DateTo2000(ToDate(1993,12,12), 99)`- Returns 2093/12/12 because the window value is greater than the year, and so the date will be changed to 20XX.
- `DateTo2000(ToDate(1993,12,12), 92)`- Returns 1993/12/12 because the window value is less than the year, and so the year will not be changed.
- This is an example of a date that is not affected by this function:

  `DateTo2000(ToDate(1899,12,12), 99)`- Returns 1899/12/12.
- This is an example of a date that is not affected by this function.

  `DateTo2000(ToDate(100,12,12), 99)`- Returns 100/12/12.
- Some databases maintain the year of a date as a two-digit field. The following simulates this type of date field.
  - `DateTo2000(ToDate (98,12,12), 99)`- Returns 2098/12/12.
  - `DateTo2000(ToDate (98,12,12), 97)`- Returns 1998/12/12.
  - `DateTo2000(ToDate (9,12,12), 10)`- Returns 2009/12/12.
  - `DateTo2000(ToDate (1,12,12), 0)`- Returns 1901/12/12.

Notes:

- If the year is less than or equal to 1899 and greater or equal to 100, there will be no change to the date.
- Whether your database dates are interpreted as two digit years or 4 digit years (19xx) is dependent on the database driver you are using. The result of the formula will be the same in either case.

Day()

Day()

This function returns the day of the month.

Return value

The return value is an Integer.

Example

If today is Oct. 17, 1999, the return value of the following statement is 17.

```
Day()
```

Day(Date a)

This function extracts the day portion of a specified Date value.

Parameter

a - A Date value.

Return value

The return value is an Integer.

Example

If the current date is Oct. 17, 1999, the return value of the following statement is 17.

```
Day(CurrentDate())
```

Day(Datetime a)

This function extracts the day portion of a specified Date value.

Parameter

a - A TimeStamp value.

Return value

The return value is an Integer.

Example

If the current date is Oct. 17, 1999 and the current time is 12:27:15, the return value of the following statement is 17.

```
Day(CurrentDateTime())
```

DayOfWeek()

DayOfWeek()

This function returns the day of the week.

Return value

The return value is an Integer.

Example

If today is Thursday, the return value of the following statement is 5.

```
DayOfWeek()
```

DayOfWeek(Date a)

This function extracts the day portion of a specified Date value, and indicates which day of the week.

Parameter

a - A Date value.

Return value

The return value is an Integer.

Example

If today is Oct. 17, 1999 and today is Sunday, the return value of the following statement is 1.

```
DayOfWeek(CurrentDate())
```

DayOfWeek(Datetime a)

This function extracts the day portion of a specified Date value, and indicates which day of the week.

Parameter

a - A TimeStamp value.

Return value

The return value is an Integer.

Example

If the current date is Oct. 17, 1999, current time is 12:27:15 and today is Sunday, the return value of the following statement is 1 (Sunday is the first day of the week).

```
DayOfWeek(CurrentDateTime())
```

DayOfYear()

This function returns the day of the year.

Return value

The return value is an Integer.

Example

If today is Oct. 17, 1999, the return value of the following statement is 290.

```
DayOfYear()
```

DTSTo2000(DateString, Number)

- **2 digit years(xx)**
  If the Year value is greater than the windowing number, 19 is appended before the 2 digits(19xx). If the Year value is less than or equal to the windowing number, 20 is appended (20xx).
- **4 digit years(19xx)**
  If the last two digits in the Year value are greater than the windowing number, the Year is retained as found in the Date field (19xx). If the two digits in the Year value are less than or equal to the windowing number, the first two digits are changed to 20 (20xx). If the first two digits in the year field are 20, the Year is retained as found in the Date field (20xx).

  Note: If the year is less than or equal to 1899 and greater than or equal to 100, there will be no change to the date.

Parameters

- Date String - Accepts only valid Date fields, with either 2 digit or 4 digit years. A DateTime string entered in the format: "yyyy/MM/dd HH: mm: ss.00" or "yy/MM/dd HH:mm:ss:00". For example, "1997/04/11 12:12:12.00" or "97/04/11 12:12:12.00".
- Number - A number between 0 and 99 corresponding with the desired windowing year.

Return value

A DateTime string with a four digit year.

Examples

- `DTSTo2000("1988-12-12 12:12:12", 90)`- Returns 2088-12-12 12:12:12 because the window value is greater than the year, and so the year will be changed to 20XX.
- `DTSTo2000("1988-12-12 12:12:12", 85)`- Returns 1988-12-12 12:12:12 because the window value is less than the year, and so the year will not change.
- These are examples of a date that will not be affected by the function:
  - `DTSTo2000("1899/12/12 1:2:3am", 99)`- Returns 1899/12/12 1:2:3am.
  - `DTSTo2000("100/12/12 1:02:03AM", 99)`- Returns 100/12/12 1:02:03AM.
- Some databases maintain the year of a date as a two-digit field and so these samples will simulate that date field:
  - `DTSTo2000("98/12/12 1:02:03AM", 99)`- Returns 2098/12/12 1:02:03AM.
  - `DTSTo2000("98/12/12 1:02:03", 97)`- Returns 1998/12/12 1:02:03AM.
  - `DTSTo2000("9/12/12 1:02:03AM", 10)`- Returns 2009/12/12 1:02:03AM.
  - `DTSTo2000("1/12/12 1:02:03AM", 0)`- Returns 1901/12/12 1:02:03AM.

DTSToSeconds(string)

Evaluates the specified string, and converts the Time value to the number of seconds from 00:00:00 (12:00 midnight) to the specified time.

Parameter

string - A String which includes a Date and a Time value.

Return value

Whole number.

Example

`DTSToSeconds("2004/06/11 13:42:15")`- Returns 49335.

FirstDayOfMonth(Date a)

This function returns the first day of the month.

Parameter

a - A Date value.

Return value

The return value is a Date.

Example

If the current date is 11/24/2008, the return value of the following statement is 11/01/2008.

```
FirstDayOfMonth(CurrentDate())
```

FirstDayOfQuarter(Date a)

This function returns the first day of the quarter.

Parameter

a - A Date value.

Return value

The return value is a Date.

Example

The return value of the following statement is 10/01/2008.

```
FirstDayOfQuarter(ToDate("11/24/2008"))
```

FirstDayOfWeek(Date a)

This function returns the first day of the week.

Parameter

a - A Date value.

Return value

The return value is a Date.

Example

The return value of the following statement is 11/23/2008.

```
FirstDayOfWeek(ToDate("11/24/2008"))
```

FirstDayOfYear(Date a)

This function returns the first day of the year.

Parameter

a - A Date value.

Return value

The return value is a Date.

Example

The return value of the following statement is 01/01/2008.

```
FirstDayOfYear(ToDate("11/24/2008"))
```

FirstSundayOfMonth(Date a)

This function returns the first Sunday of the month.

Parameter

a - A Date value.

Return value

The return value is a Date.

Example

The return value of the following statement is 11/02/2008.

```
FirstSundayOfMonth(ToDate("11/24/2008"))
```

FirstSundayOfYear(Date a)

This function returns the first Sunday of the year.

Parameter

a - A Date value.

Return value

The return value is a Date.

Example

The return value of the following statement is 01/06/2008.

```
FirstSundayOfYear(ToDate("11/24/2008"))
```

ForEachDay(Datetime a)

This function returns the Date portion of a specified DateTime value.

Parameter

a - A DateTime value.

Return value

The return value is a Date.

Example

If the DateTime is Dec.25,1999 20:12:50, the return value of the following statement is 12/25/99.

```
ForEachDay(ToDateTime(1999, 12, 25, 20, 12, 50))
```

ForEachHalfMonth(DateTime a)

If the day in the DateTime is less than 15, this function will return to the first day of the month. If the day is more than 15, it will return to the 16th day of the month.

Parameter

a - A DateTime value.

Return value

The return value is a date.

Examples

- If the datetime is Sep.13,1999 12:27:15, the return value of the following statement is 09/01/99.

  `ForEachHalfMonth(ToDateTime(1999,9,13,12,27,15))`
- If the datetime is Oct.18,1999 5:21:22, the return value of the following statement is 10/16/99.

  `ForEachHalfMonth(ToDateTime(1999,10,18,5,21,22))`

ForEachHalfYear(DateTime a)

If the month of a specified DateTime is before July, the function will return to Jan.1 of the year. Otherwise the function will return to July 1 of the year.

Parameter

a - A DateTime value.

Return value

The return value is a date.

Examples:

- If the datetime is May. 28,1999 10:10:25, the return value of the following statement is 01/01/99.

  `ForEachHalfYear(ToDateTime(1999, 5, 28, 10, 10, 25))`
- If the datetime is Nov. 15,1998 11:40:23, the return value of the following statement is 07/01/99.

  `ForEachHalfYear(ToDateTime(1998, 11, 15, 11, 40, 23))`

ForEachMonth(DateTime a)

This function returns the first day of the month in a specified DateTime value.

Parameter

a - A DateTime value.

Return value

The return value is a date.

Example

If the datetime is Sep.24,1999 12:05:56, the return value of the following statement is 09/01/99.

```
ForEachMonth(ToDateTime(1999,9,24,12,5,56))
```

ForEachQuarter(DateTime a)

If the month of a specified DateTime is before April, the function will return to Jan. 1 of the year. If the month is after April but before July, the function will return to April. 1 of the year. If the month is after July but before October, the function will return to July. 1 of the year. If the month is after October, the function will return to Oct.1 of the year.

Parameter

a - A DateTime value.

Return value

The return value is a date.

Examples

- If the datetime is Feb.14,1997 2:10:23, the return value of the following statement is 01/01/97.

  `ForEachQuarter(ToDateTime(1997,2,14,2,10,23))`
- If the datetime is May. 28,1999 10:10:25, the return value of the following statement is 04/01/99.

  `ForEachQuarter(ToDateTime(1999,5,28,10,10,25))`
- If the datetime is Aug. 28,1999 10:10:25, the return value of the following statement is 07/01/99.

  `ForEachQuarter(ToDateTime(1999,8,28,10,10,25))`
- If the datetime is Nov. 11,1998 11:40:23, the return value of the following statement is 10/01/98.

  `ForEachQuarter(ToDateTime(1998,11,11,11,40,23))`

ForEachWeek(DateTime a)

This function returns the Date of the first day of the week.

Parameter

a - A DateTime value.

Return value

The return value is a Date.

Example

If the DateTime is Oct. 15,1999 5:15:20 and it is Friday, the return value of the following statement is 10/10/99.

```
ForEachWeek(ToDateTime(1999, 10, 15, 5, 15, 20))
```

ForEachYear(DateTime a)

This function will return to Jan.1 of the year.

Parameter

a - A DateTime value.

Return value

The return value is a date.

Example

If the datetime is Aug. 28,1999 10:10:25, the return value of the following statement is 01/01/99.

```
ForEachYear(ToDateTime(1999, 8, 28, 10, 10, 25))
```

Hour()

Hour()

This function extracts the hour portion of the current time.

Return value

The return value is an integer.

Example

If the current time is 8:15:30, the return value of the following statement is 8.

```
Hour()
```

Hour(DateTime a)

This function extracts the hour portion of a specified TimeStamp value.

Parameter

a - A TimeStamp value.

Return value

The return value is an Integer.

Example

Suppose the datetime is Oct. 15,1999 9:35:22. The return value of the following statement is 9.

```
Hour(ToDateTime(1999,10,15,9,35,22))
```

Hour(Time a)

This function extracts the hour portion of a specified Time value.

Parameter

a - A Time value.

Return value

The return value is an Integer.

Example

Suppose the time is 10:05:26. The return value of the following statement is 10.

```
Hour(ToTime(10, 5, 26))
```

IsDate(number or string)

Returns true if the given String or Number value can be converted to a valid Date and returns false otherwise. A valid Date is any Date between 100-1-1 and 9999-12-31.

Parameters

- number - A Number value representing the number of days starting from January 1, 1900. It can be positive or negative, and is truncated if fractional.
- string - A text string representing a date and many forms can be accepted. If the string is a right Date value and format, the function returns true, otherwise false.

Return value

Boolean value, True or False.

Examples

- `IsDate(50)` - Returns true since the number 50 is interpreted as 50 days from Jan. 1, 1900, which is Feb 19, 1900.
- `IsDate(-50)` - Returns true since the number –50 is interpreted as 50 days before Jan 1, 1900, which is Nov 11, 1899.
- `IsDate("Feb 1, 2004")` - Returns true since the string is a right Date value and there is only one blank between 'Feb' and '1,' and '2004'.
- `IsDate("2004 - 2 - 19")` - Returns false since the String format is not right Date format.
- `IsDate("2003/2/29")` - Returns false since 2003 is not leap year.
- `IsDate("2004*2*19")` - Returns false since the String format is not right Date format.

IsDateTime(number/string)

Returns True if the given Number or String value can be converted to a valid DateTime, and returns False otherwise.

Parameters

- number - A Number value or expression to be tested for convertibility to a DateTime value. It can be positive, negative or fractional. It is interpreted as a number of days from Jan 1, 1900.
- string - A String value or expression to be tested for convertibility to a DateTime value. Many forms are accepted. If the given String value and format are right DateTime value and format, the function will return true, otherwise false.

Return value

Boolean value, true or false.

Examples

- `IsDateTime(15.2)` - Returns true since the given number argument is interpreted as the DateTime value 1900-01-15 04:48:00.

- IsDateTime("Feb 23, 2004 15:23:25") - Returns true.
- IsDateTime("2004-2-23") - Returns true.
- IsDateTime("15:23:25") - Returns true.

IsTime(number or string)

Returns True if the given Number or String value can be converted to a valid Time, and returns False if otherwise.

Parameters

- number - A Number value or expression to be tested for convertibility to a Time value. It can be positive, negative or fractional. It is interpreted as units of 24 hours. If the number makes the Time value between 0:0:0: and 23:59:59, the function will return true, otherwise false.
- string - A String value or expression to be tested for convertibility to a Time value. Many forms are accepted. If the String value and the format are right Time value and format, the function will return true, otherwise false.

Return value

Boolean value, true or false.

Examples

- IsTime(0.5) - Returns true since the given number argument is interpreted as 0.5 units of 24 hours, which is 12:00:00.
- IsTime(3.8) - Returns true since the given number is interpreted as 3.8 units of 24 hours, which is 19:12:00.
- IsTime("8:30:00") - Returns true since 8:30:00 is right Time value and format.
- IsTime("8 30:00") - Returns false since the given string is not right Time format.
- IsTime("8:30:62") - Returns false since the given string is not right Time value.

Minute()

Minute()

This function returns the minute portion of the current time.

Return value

The return value is an Integer.

Example

If the current time is 12:41:27, the return value of the following statement is 41.

`Minute()`

Minute(DateTime a)

This function returns the minute portion of a specified DateTime.

Parameter

a - A DateTime value.

Return value

The return value is an Integer.

Example

Suppose the datetime is Sep. 21,1997 12:20:25. The return value of the following statement is 20.

`Minute(ToDateTime(1997, 9, 21, 12, 20, 25))`

Minute(Time a)

This function returns the minute portion of a specified time.

Parameter

a - A Time value.

Return value

The return value is an Integer.

Example

Suppose the time is 8:30:27. The return value of the following statement is 30.

`Minute(ToTime(8,30,27))`

Month()

Month()

This function adding one returns the current month of the year.

Return value

The return value is an Integer, which ranges from 0 to 11.

Example

If the current month is September, the return value of the following statement is 8.

`Month()`

Month(Date a)

This function extracts the month portion from a specified Date value.

Parameter

a - A Date value.

Return value

The return value is an Integer.

Example

If the date is June 15,1999, the return value of the following statement is 6.

`Month(ToDate(1999, 6, 15))`

Month(DateTime a)

This function extracts the month portion from a specified TimeStamp value.

Parameter

a - A TimeStamp value.

Return value

The return value is an Integer.

Example

If the timestamp is July 15, 1999 10:10:10, the return value of the following statement is 7.

`Month(ToDateTime(1999, 7, 15, 10, 10, 10))`

MonthName(month, abbreviate)

Returns a String name for the specified month.

Parameters

- month - A whole number representing the month of the year, with values between 1 and 12, where 1 represents January.
- abbreviate - An optional Boolean value that indicates if the month name is to be abbreviated. If the abbreviate value is true, the function will return the abbreviated month name. Otherwise the function will return the full month name.

Return value

String value.

Examples

- MonthName(4) - Returns String value "April".
- MonthName(12, true) - Returns String value "Dec".
- MonthName(12, false) - Returns String value "December".

Now()

This function prints the current time on a report.

Return value

The return value is a Time.

Example

If the current time is 1:24:35 in the afternoon, the return value of the following statement is 1:24:35 PM.

`Now()`

Quarter()

Returns the quarter for the given date (1 for the first quarter).

Overloads

- Quarter()
- Quarter(Date,Integer)
- Quarter(Datetime,Integer)

Parameters

- Date - A Date value.
- Integer - An Integer value in the range of 1 to 12. It is an optional argument which indicates the beginning month of the first quarter. Specifying the number is useful for fiscal quarter. If b is not specified, the default is 1 which means January is the beginning month of the first quarter as same as the calendar quarter.
- Datetime - A DateStamp value.

Return value

Return an integer (1, 2, 3, 4).

Examples

- Quarter() - Return 3 if it is September according to the current system date.
- Quarter(ToDate(2006,11,28),4) - Return 3. In this example, the first quarter begins from the 4th month - April, so the 11th month is in the third quarter.
- Quarter(ToDateTime(2006,7,15,10,10,10),4) - Return 2.

Second()

Second()

This function extracts the second portion of the current time.

Return value

The return value is an Integer.

Example

If the current time is 4:55:03, the return value of the following statement is 3.

`Second()`

Second(Datetime a)

This function extracts the second portion of a specified TimeStamp.

Parameter

a - A TimeStamp value.

Return value

The return value is an Integer.

Example

Suppose the datetime is Oct. 15,1999 10:15:17, the return value of the following statement is 17.

`Second(ToDateTime(1999, 10, 15, 10, 15, 17 ))`

Second(Time a)

This function extracts the second portion of a specified time.

Parameter

a - A Time value.

Return value

The return value is an Integer.

Example

Suppose the time is 7:51:59, the return value of the following statement is 59.

`Second(ToTime(7, 51, 59))`

SelectedDate()

Returns the date selected in the calendar or returns today() when no calendar is available for selecting a date.

There is a built-in expression Selected Date in Calendar calling this function. The expression can be found in the Name drop-down list of the Calendar dialog.

Return value

Date value.

Example

If the calendar date is March 12, 2012, the function returns 03/12/2012.

Timer()

Returns the number of seconds elapsed since midnight.

Return value

Number value.

Example

If the current time is 9:28:30am, the function returns 34110.

ToDate()

ToDate(number or string)

Uses the function IsDate(number or string) to check whether the number/string is right or not. If the number/string is right, the function returns a Date value between 100-1-1 and 9999-12-31. If the number/string is wrong, it returns null.

Parameters

- number - A value representing the number of days starting from January 1, 1900. It can be positive or negative, and is truncated if fractional.
- string - A text string representing a date and many forms can be accepted.

Return value

Date value or null.

Examples

- ToDate(50.5) - Returns 1900-2-19 since fraction is truncated.
- ToDate(-50) - Returns 1970-01-01.
- ToDate(22222222) - Returns 1970-01-01.
- ToDate("2004 2 19") - Returns 2004-2-19.
- ToDate("99-5") - Returns 1999-5-1.
- ToDate("1999-2-29") - Returns null since 1999 is not leap year.

ToDate(Date a)

This function returns the specified date.

Parameter

a - A Date value.

Return value

The return value is a Date.

Example

Suppose the date is July 15, 1999, the return of the following statement is 07/15/99.

`ToDate(ToDate(1999, 7, 15))`

ToDate(Datetime a)

This function extracts the date portion of a specified DateTime.

Parameter

a - A TimeStamp value.

Return value

The return value is a Date value.

Example

Suppose the DateTime is July 15,1999 12:20:30, the return value of the following statement is 07/15/99.

`ToDate(ToDateTime(1999, 7, 15, 12, 20, 30 ) )`

ToDate(integer d)

This function converts the argument d (A milliseconds value represents the number of milliseconds that have passed since January 1, 1970 00:00:00.000 GMT) to a data value.

Parameter

d - A Milliseconds value, a DBField value or a parameter. It should be a long integer.

Return value

The return value is a Date.

Example

The return value of the following statement is 01/02/70.

`ToDate(72000000)`

Note: When the milliseconds value is larger than 2^{32}, you should use DBField or parameter instead.

ToDate(integer y, Integer m, Integer d)

This function creates a Date value from arguments y, m and d.

Parameters

- y - A BigInt value indicating year.
- m - A BigInt value indicating month.
- d - A BigInt value indicating day.

Return value

The return value is a Date.

Example

The return value of the following statement is 07/15/99.

`ToDate(1999, 7, 15)`

ToDateTime()

ToDateTime(number or string)

Use the function IsDateTime to check whether the number/string is right or not. If it is right, the function returns a DateTime value between 100-1-1 and 9999-12-31, otherwise the function returns null.

Parameters

- number - A Number value or expression to be tested for convertibility to a DateTime value. It can be positive, negative or fractional. It is interpreted as a number of days from Jan 1, 1900.
- string - A String value or expression to be tested for convertibility to a DateTime value. Many forms are accepted.

Return value

DateTime value or null.

Examples

- ToDateTime(255.35) - Returns 1900-09-12 08:23:59.
- ToDateTime("Jan 3, 2004") - Returns 2004-01-03 00:00:00.
- ToDateTime("5:25:26pm") - Returns 2004-02-23 17:25:26 because the current date is 2004-2-23.
- ToDateTime("2003-2-29 15:26:25") - Returns null because the given String value is not right DateTime value.
- ToDateTime("2004 *2 *29 15:26:25") - Returns null because the given String format is not right DateTime format.

ToDateTime(date a)

Returns a DateTime value.

Parameter

a - A Date value that can be converted to a DateTime type value by this function.

Example

ToDateTime(ToDate("2004-2-23")) - Returns 2004-02-23 00:00:00.

ToDateTime(Date a, Time b)

This function creates a TimeStamp value from arguments a and b.

Parameters

- a - A Date value.
- b - A Time value.

Return value

The return value is a DateTime.

Example

Suppose the date is July 15,1999 and the time is 12:20:30, the return value of the following statement is 1999-07-15 12:20:30.

`ToDateTime(ToDate(1999, 7, 15), ToTime(12, 20, 30))`

ToDateTime(Integer d)

This function converts the argument d (A milliseconds value represents the number of milliseconds that have passed since January 1, 1970 00:00:00.000 GMT) to a TimeStamp value.

Parameter

d - A Milliseconds value, a DBField value or a parameter. It should be a long integer.

Return value

The return value is a DateTime.

Example

The return value of the following statement is 1970-01-02 04:00:00.

`ToDateTime(72000000)`

Notes:

- In this function, you can decide the display format by selecting the format from the Report Inspector. The format of the example above is yyyy-mm-dd hh:mm:ss.
- When the milliseconds value is larger than 2^{32}, you should use DBField or parameter instead.

ToDateTime(Integer y, Integer m, Integer d)

This function creates a TimeStamp value from arguments y, m and d.

Parameters

- y - A BigInt value indicating year.
- m - A BigInt value indicating month.
- d - A BigInt value indicating day.

Return value

The return value is a DateTime.

Example

Suppose the date is July 15,1999, the return value of the following statement is 1999-07-15 12:00:00.

`ToDateTime(1999, 7, 15)`

Note: In this function, because you only input year, month and day, the system will output 12:00:00 as the default time. Also you can decide the display format by selecting the format from the Report Inspector. The format of the example above is yyyy-mm-dd hh:mm:ss.

ToDateTime(Integer y, Integer m, Integer d, Integer h, Integer i, Integer s)

This function creates a TimeStamp value from arguments y, m, d, h, i and s.

Parameters

- y - A BigInt value indicating year.
- m - A BigInt value indicating month.
- d - A BigInt value indicating day.
- h - A BigInt value indicating hour.
- i - A BigInt value indicating minute.
- s - a BigInt value indicating second.

Return value

The return value is a DateTime.

Example

Suppose the date is July 15,1999 and time is 8:05:30, the return value of the following statement is 1999-07-15 08:05:30.

`ToDateTime(1999, 7, 15, 8, 5, 30)`

Today()

This function returns the current date on a report.

Return value

The return value is a date.

Example

If the current date is Oct. 11, 1999, the return value of the following statement is 10/11/99.

`Today()`

ToTime()

ToTime(number or string)

Use the function IsTime to check whether the number/string is right or not. If the number/string is right, the function returns a Time value between 00:00:00 and 23:59:59, otherwise it will return null.

Parameters

- number - A Number value or expression to be tested for convertibility to a Time value. It can be positive, negative or fractional. It is interpreted as units of 24 hours.
- string - A String value or expression to be tested for convertibility to a Time value. Many forms are accepted.

Return value

Time value or null.

Examples

- ToTime(0.2) - Returns 04:48:00.
- ToTime(-0.2) - Returns 19:12:00.
- ToTime(50) - Returns 00:00:00.
- ToTime("10:34:25") - Returns 10:34:25.
- ToTime("2:25pm") - Returns 02:25:00.
- ToTime("2004-2-20") - Returns 00:00:00.
- ToTime("24:25:23") - Returns 00:25:23.
- ToTime("20*23:25") - Returns null since the given String format is not time format.

ToTime(DateTime a)

This function extracts the time portion from a specified TimeStamp value.

Parameter

a - A TimeStamp value.

Return value

The return value is a Time.

Example

Suppose the date is July 15,1999 and time is 10:55:45. The return value of the following statement is 10:55:45.

`ToTime(ToDateTime(1999, 7, 15, 10, 55, 45 ))`

ToTime(Integer t)

This function converts the argument t (A milliseconds value represents the number of milliseconds that have passed since January 1, 1970 00:00:00.000 GMT) to a Time value.

Parameter

t - A milliseconds value, a DBField value or a parameter. It should be a long integer.

Return value

The return value is a Time.

Example

The return value of the following statement is 08:00:01.

`ToTime(1000)`

Note: When the milliseconds value is larger than 2^{32}, you should use DBField or parameter instead.

ToTime(Integer a, Integer b, Integer c)

This function creates a Time value from argument a, b and c.

Parameters

- a - A BigInt value, indicating hours.
- b - A BigInt value, indicating minutes.
- c - A BigInt value, indicating seconds.

Return value

The return value is a Time.

Example

The return value of the following statement is 10:10:10.

`ToTime(10, 10, 10)`

WeekdayName(weekday, abbreviate, firstDayOfWeek)

WeekdayName returns a string indicating the name of the specified day of the week.

Overloads

- WeekdayName(weekday)
- WeekdayName(weekday, abbreviate)
- WeekdayName(weekday, abbreviate, firstDayOfWeek)

Parameters

- Weekday - The numeric designation for the day of the week.
- abbreviate - An optional Boolean value that indicates whether or not the weekday name is to be abbreviated.
- FirstDayOfWeek - An optional number indicating the first day of the week.

Constant	Description
jrUseSystem	0 (Use the current system date)
jrSunday	1
jrMonday	2
jrTuesday	3
jrWednesday	4
jrThursday	5
jrFriday	6
jrSaturday	7

Return value

String value.

Examples

- WeekdayName (2) - Returns String value Monday.
- WeekdayName (1) - Returns String value Sunday.
- WeekdayName (4, true) - Returns String value Wed.
- WeekdayName (4, false) - Returns String value Wednesday.
- WeekdayName (3, true, "jrMonday") - Returns Wed, since abbreviation is selected and the first day of the week is specified to be Monday.
- WeekdayName (3, true, "jrUseSystem") - If the current day is Friday, the function returns Sun, since the first day of week is specified to the current day, and the third day from Friday is Sunday.

WeekFrom1970()

WeekFrom1970()

This function computes the total number of weeks from 1970 to the current date.

Return value

The return value is an Integer.

Example

If the current date is Oct. 13,1999, the return value of the following statement is 1554.

`WeekFrom1970()`

WeekFrom1970(Date, a)

This function computes the total number of weeks from 1970 to a specified date.

Parameter

a - A Date value.

Return value

The return value is an Integer.

Example

Suppose the date is July 8,1995. The return value of the following statement is 1331.

`WeekFrom1970(ToDate(1995, 7, 8))`

WeekFrom1970(DateTime, a)

This function computes the total number of weeks from 1970 to a specified DateTime.

Parameter

a - A DateTime value.

Return value

The return value is an Integer.

Example

Suppose the DateTime is Jan. 2,1988 10:40:50. The return value of the following statement is 940.

`WeekFrom1970(ToDateTime(1988, 1, 2, 10, 40, 50))`

WeekOfMonth()

WeekOfMonth()

This function returns the week of the current month.

Return value

The return value is an Integer.

Example

If the current date is Oct. 13,1999, the return value of the following statement is 3.

`WeekOfMonth()`

WeekOfMonth(Date a)

This function returns the week of the month of a specified date.

Parameter

a - A Date value.

Return value

The return value is an Integer.

Example

If the current date is Oct. 13,1999, the return value of the following statement is 3.

`WeekOfMonth(ToDate(1999, 10, 13))`

WeekOfMonth(DateTime a)

This function evaluates the week of the month of a specified TimeStamp.

Parameter

a - A TimeStamp value.

Return value

The return value is an Integer.

Example

Suppose the timestamp is Jan. 20,1999 10:40:50. The return value of the following statement is 4.

`WeekOfMonth(ToDateTime(1999, 1, 20, 10, 40, 50))`

WeekOfYear()

WeekOfYear()

This function returns the week of the current year.

Return value

The return value is an Integer.

Example

If the current date is Oct. 13,1999, the return value of the following statement is 42.

`WeekOfYear()`

WeekOfYear(Date a)

This function returns the week of the year in a specified date.

Parameter

a - A Data value.

Return value

The return value is an Integer.

Example

Suppose the date is July 8,1999. The return value of the following statement is 28.

`WeekOfYear(ToDate(1999, 7, 8))`

WeekOfYear(DateTime a)

This function returns the week of the year in a specified TimeStamp.

Parameter

a - A TimeStamp value.

Return value

The return value is an Integer.

Example

Suppose the timestamp is Feb. 4,1999 10:40:50. The return value of the following statement is 6.

`WeekOfYear(ToDateTime(1999, 2, 4, 10, 40, 50))`

Year()

Year()

This function returns the current year.

Return value

The return value is an Integer.

Example

If the current year is 1999, the return value of the following statement is 1999.

`Year()`

Year(Date a)

This function extracts the year portion of a specified date.

Parameter

a - A Data value.

Return value

The return value is an Integer.

Example

Suppose the date is Jan.15,1995. The return value of the following statement is 1995.

`Year(ToDate(1995, 1, 15))`

Year(DateTime a)

This function extracts the year portion of a specified TimeStamp value.

Parameter

a - A TimeStamp value.

Return value

The return value is an Integer.

Example

Suppose the timestamp is July 15,1988 10:40:50. The return value of the following statement is 1988.

`Year(ToDateTime(1988, 7, 15, 10, 40, 50))`

Financial

The following are functions of the Financial type.

- DDB()
- FRAccRecTurnover()
- FRCashFlowVsTotalDebt()
- FRCurrentRatio()
- FRDebtEquityRatio()
- FRDividendYield()
- FREarningsPerCommonShare()
- FREquityVsTotalAssets()
- FRGrossProfitMargin()
- FRInterestCoverage()
- FRInventoryTurnover()
- FRNetProfitMargin()
- FROperatingProfitMargin()
- FRPriceEarningsRatio()
- FRQuickRatio()
- FRReturnOnCommonEquity()
- FRReturnOnEquity()
- FRReturnOnInvestedCapital()
- FRReturnOnNetFixedAssets()
- FRReturnOnTotalAssets()
- FV()
- IPmt()
- IRR()
- MIRR()
- NPer()
- NPV()
- Pmt()
- PPmt()
- PV()
- Rate()
- SLN()
- SYD()

DDB(cost, salvage, life, period, factor)

DDB returns a number specifying the depreciation of an asset for a specific time period, using the double-declining balance method or another method as specified by the factor argument.

Overloads

- DDB(cost, salvage, life, period)
- DDB(cost, salvage, life, period, factor)

Parameters

- cost - A Number or Currency that specifies the initial cost of the asset. The value is nonnegative and greater than or equal to salvage.
- salvage - A Number or Currency that specifies the value of the asset at the end of its useful life. The value is nonnegative.
- life - A positive Number that specifies the length of the useful life of the asset.
- period - A Number that specifies the period for which asset depreciation is calculated. The value is positive and less than or equal to life. The parameters life and period must have the same units.
- factor - An optional positive number that specifies the rate at which the balance declines. If omitted, 2 (double-declining method) will be used.

Return value

Number value.

Examples

Suppose a company purchases a fleet of cars for $560,000. The cars have a lifetime of 12 years and a salvage value of $30,000. They are depreciated using the double-declining method.

- `DDB(560000, 30000, 12, 1)` - Returns 93333.33. The first year's depreciation is $93,333.33.
- `DDB(560000, 30000, 13, 5)` - Returns 44164.37. The fourth year's depreciation is $44,164.37.
- `DDB(560000, 30000, 13, 13)` - Returns 11605.58. The final year's depreciation is $11,605.58.

FRAccRecTurnover(Number accountReceivable, Number sales, DbDouble numOfDays)

This function computes the turnover of the account receivable.

Parameters

- accountReceivable - A Double value indicating the account receivable.
- sales - A Double value indicating the sales.
- numOfDays - A Double value indicating the number of days.

Return value

The return value is a Number.

Examples

- If the account receivable is 220000.00, the sales value is 450000.00, the number of days is 100, the return value of the following statement is 48.89.

  `FRAccRecTurnover(220000.00, 450000.00, 100)`
- `FRAccRecTurnover(10000,100000,360)` - Returns 36 days.

FRCashFlowVsTotalDebt(Number cashFlow,Number totalDebt)

This function returns the result of cash flow vs total debt.

Parameters

- cashFlow - A Double value indicating the cash flow.
- totalDebt - A Double value indicating the total debt.

Return value

The return value is a Number.

Example

If the cash flow is 250000.00, the total debt is 280000.00, the return value of the following statement is 0.89.

`FRCashFlowVsTotalDebt(250000.00, 280000.00)`

FRCurrentRatio(Number curAssets, Number curLiabilities)

This function computes the current ratio.

Parameters

- curAssets - A Double value indicating the current ratio.
- curLiabilities - A Double value indicating the current liabilities.

Return value

The return value is a Number.

Example

If the current assets are 1800000.00, the current liabilities are 150000.00, the return value of the following statement is 12.00.

`FRCurrentRatio(1800000.00, 150000.00)`

FRDebtEquityRatio(Number totalLiabilities, Number totalEquity)

This function computes the debt equity ratio.

Parameters

- totalLiabilities - A Double value indicating the total liabilities.
- totalEquity - A Double value indicating the total equity.

Return value

The return value is a Number.

Example

If the total liabilities are 175000.00, the total equity is 215000.00, the return value of the following statement is 0.81.

`FRDebtEquityRatio(175000.00, 215000.00)`

FRDividendYield(Number dividEnd, Number marketPrice)

This function computes the dividend yield.

Parameters

- dividend - A Double value indicating the dividend.
- marketPrice - A Double value indicating the market price.

Return value

The return value is a Number.

Example

If the dividend is 9.85, the market price is 10.87, the return value of the following statement is 0.91.

`FRDividendYield(9.85, 10.87)`

FREarningsPerCommonShare(Number netProfit, Number preferredDividend, DbDouble numOfCommonShare)

This function computes the earnings of per common share.

Parameters

- netProfit - A Double value indicating the net profit.
- preferredDividend - A Double value indicating the preferred dividend.
- numOfCommonShare - A Double value indicating the number of common shares.

Return value

The return value is a Number.

Example

If the net profit is 300000.00, the preferred dividend is 180000.00, the number of common shares is 10000, the return value of the following statement is 12.

`FREarningsPerCommonShare(300000.00, 180000.00, 10000)`

FREquityVsTotalAssets(Number totalEquity, Number totalAssets)

This function returns the result of equity vs. total assets.

Parameters

- totalEquity - A Double value indicating the total equity.
- totalAssets - A Double value indicating the total assets.

Return value

The return value is a Number.

Example

If the total equity is 215000.00, the total assets are 2200000.00, the return value of the following statement is 0.10.

`FREquityVsTotalAssets(215000.00, 2200000.00)`

FRGrossProfitMargin(Number grossProfit, Number sales)

This function is used to compute the gross profit margin.

Parameters

- grossProfit - A Double value indicating the gross profit.
- sales - A Double value indicating the sales.

Return value

The return value is a Number.

Example

If the gross profit is 350000.00, the sales value is 450000.00, the return value of the following statement is 0.78.

`FRGrossProfitMargin(350000.00, 450000.00)`

FRInterestCoverage(number cashFlow, number interestExpenses)

This function computes the interest coverage.

Parameters

- cashFlow - A Double value indicating the cash flow.
- interestExpenses - A Double value indicating the interest expenses.

Return value

The return value is a Number.

Example

If the cash flow is 250000.00, the interest expenses are 350000.00, the return value of the following statement is 0.71.

`FRInterestCoverage(250000.00, 350000.00)`

FRInventoryTurnover(Number inventory, Number sales, Number numOfDays)

This function computes the turnover of the inventory.

Parameters

- inventory - A Double value indicating the inventory.
- sales - A Double value indicating the sales.
- numOfDays - A Double value indicating the number of days.

Return value

The return value is a Number.

Example

If the inventory is 165000.00, the sales value is 450000.00, the number of days is 100, the return value of the following statement is 36.67.

`FRInventoryTurnover(165000.00, 450000.00, 100)`

FRNetProfitMargin(Number netProfit, Number sales)

This function is used to compute the net profit margin.

Parameters

- netProfit - A Double value indicating the net profit.
- sales - A Double value indicating the sales.

Return value

The return value is a Number.

Example

If the net profit is 300000.00, the sales value is 450000.00, the return value of the following statement is 0.67.

`FRNetProfitMargin(300000.00, 450000.00)`

FROperatingProfitMargin(Number operatingProfit, Number sales)

This function computes the operating profit margin.

Parameters

- operatingProfit - A Double value indicating the operating profit.
- sales - A Double value indicating the sales.

Return value

The return value is a Number.

Example

If the operating profit is 380000.00, the sales value is 450000.00, the return value of the following statement is 0.84.

`FROperatingProfitMargin(380000.00, 450000.00)`

FRPriceEarningsRatio(Number marketPrice, Number earningsPerShare)

This function computes the price earning ratio.

Parameters

- marketPrice - A Double value indicating the market price.
- earningPerShare - A Double value indicating the earning per share.

Return value

The return value is a Number.

Example

If the market price is 10.87, the earning per share is 12, the return value of the following statement is 0.91.

`FRPriceEarningsRatio(10.87, 12)`

FRQuickRatio(Number curAssets, Number inventories, Number curLiabilities)

This function computes the quick ratio.

Parameters

- curAssets - A Double value indicating the current assets.
- inventories - A Double value indicating the inventory.
- curLiabilities - A Double value indicating the current liabilities.

Return value

The return value is a Number.

Example

If the current assets are 1800000.00, the inventory is 165000.00, the current liabilities are 150000.00, the return value of the following statement is 10.90.

`FRQuickRatio(1800000.00, 165000.00, 150000.00)`

FRReturnOnCommonEquity(Number netProfit, Number preferredDividend, Number commonEquity)

This function returns the result of return on common equity.

Parameters

- netProfit - A Double value indicating the net profit.
- preferredDividend - A Double value indicating the preferred dividend.
- commonEquity - A Double value indicating the common equity.

Return value

The return value is a Number.

Example

If the net profit is 300000.00, the preferred dividend is 180000.00, the common equity is 208000.00, the return value of the following statement is 0.58.

`FRReturnOnCommonEquity(300000.00, 180000.00, 208000.00)`

FRReturnOnEquity(Number netProfit, Number totalEquity)

This function returns the results of return on equity.

Parameters

- netProfit - A Double value indicating the net profit.
- totalEquity - A Double value indicating the total equity.

Return value

The return value is a Number.

Example

If the net profit is 300000.00, the total equity is 215000.00, the return value of the following statement is 1.40.

`FRReturnOnEquity(300000.00, 215000.00)`

FRReturnOnInvestedCapital(Number netProfit, Number totalBankDebts, Number totalEquity)

This function returns the result of return on invested capital.

Parameters

- netProfit - A Double value indicating the net profit.
- totalBankDebts - A Double value indicating the total bank debts.
- totalEquity - A Double value indicating the total equity.

Return value

The return value is a Number.

Example

If the net profit is 300000.00, the total bank debts are 100000.00, the total equity is 215000.00, the return value of the following statement is 0.95.

`FRReturnOnInvestedCapital(300000.00, 100000.00, 215000.00)`

FRReturnOnNetFixedAssets(Number netProfit, Number netFixedAssets)

This function returns the result of return on net fixed assets.

Parameters

- netProfit - A Double value indicating the net profit.
- netFixedAssets - A Double value indicating the net fixed assets.

Return value

The return value is a Number.

Example

If the net profit is 300000.00, the net fixed assets are 400000.00, the return value of the following statement is 0.75.

`FRReturnOnNetFixedAssets(300000.00, 400000.00)`

FRReturnOnTotalAssets(Number netProfit, Number totalAssets)

This function returns the results of return on total assets.

Parameters

- netProfit - A Double value indicating the net profit.
- totalAssets - A Double value indicating the total assets.

Return value

The return value is a Number.

Example

If the net profit is 300000.00, the total assets are 2200000.00, the return value of the following statement is 0.14.

`FRReturnOntotalAssets(300000.00, 2200000.00)`

FV(rate, periods, payment, presentMoney, type)

Returns a number specifying the future value of an annuity based on periodic, fixed payments and a fixed interest rate.

Overloads

- FV(rate, periods, payment)
- FV(rate, periods, payment, presentMoney)
- FV(rate, periods, payment, presentMoney, type)

Parameters

- rate - A Number that specifies the interest rate per period.
- periods - A positive Number that specifies the total number of payment periods in the annuity. The units used for specifying rate and periods must consistent. For example, if periods is the number of periods in months, then rate is a monthly interest rate.
- payment - A Number or Currency that specifies the payment to be made each period.
- presentMoney - An optional Number or Currency that specifies the present value of a series of future payments.
- type - An optional Number that specifies when payments are due. Specify 0 if payments are due at the end of the payment period, and 1 if payments are due at the beginning of the period. If omitted, 1 will be used.

Return value

Number value.

Examples

- Suppose that you put $1100 per month into a retirement savings plan that pays 6 percent annual interest, compounded monthly. How much will the account be worth after 20 years?

  `FV(0.06 / 12, 20 * 12, -1100)` - Returns 507144.99. So your account will have $507,145. The payment (-1100) is negative since you are paying out the money to the plan.

  The above example assumes that you make your payments into the plan at the end of the month. Thus, after the first month, your plan would have $1100 in it, since there wouldn't have been enough time for any interest to accrue.
- Suppose that instead, you make your payments at the start of the month,

  `FV(0.06 / 12, 20 * 12, -1100, 0, 1)` - Returns 510786.21 (rounded to the nearest integer). So your account will have $510,786. You will save $3,642 more by depositing at the beginning of the month.
- Now suppose that in addition to making payments at the start of the month, you start your plan with an initial deposit of $25,000,

  `FV(0.06 / 12, 20 * 12, -1100, -25000, 1)` - Returns 593541.33 (rounded to the nearest integer). Your account will have $ 593,541 after 20 years.
- You can also use the FV function to calculate the future value of a lump sum deposit. For example, if you deposit $25,000 into a plan that pays 6 percent annual interest compounded monthly for 20 years,

  `FV(0.06 / 12, 20 * 12, 0, -25000)` - Returns 82755.12 (rounded to the nearest integer). You would have $82755.12 in your account.

IPmt(rate, period, periods, presentMoney, futureMoney, type)

IPmt returns a number specifying the interest payment for a given period of an annuity based on periodic, fixed payments and a fixed interest rate.

Overloads

- IPmt(rate, period, periods, presentMoney)
- IPmt(rate, period, periods, presentMoney, futureMoney)
- IPmt(rate, period, periods, presentMoney, futureMoney, type)

Parameters

- rate - A Number that specifies the interest rate per period.
- period - A Number that specifies the payment period in the range 1 through periods.
- periods - A positive Number that specifies the total number of payment periods in the annuity. The units used for specifying rate, period and periods must be consistent. For example, if periods is the number of periods in months, then rate is a monthly interest rate and period specifies a month.
- presentMoney - A Number or Currency that specifies the present value, or value today, of a series of future payments or receipts.
- futureMoney - An optional Number or Currency that specifies the future value, or cash balance you want after you've made the final payment. If omitted, 0 will be used.
- type - An optional Number that specifies when payments are due. Specify 0 if payments are due at the end of the payment period, and 1 if payments are due at the beginning of the period. If omitted, 0 will be used.

Return value

Number value.

Examples

- Suppose that you want to take out a $250,000 loan payable monthly over 15 years at an annual interest rate of 6.5 percent. The following formula returns the amount of interest that you pay in your first loan payment. Note that the monthly interest rate is 0.065 / 12 and the number of months of the loan is 15 * 12.

  `IPmt(0.065 / 12, 1, 15 * 12, 250000)` - Returns the Number value -1354.17 (rounded to 2 decimals). The value is negative because it represents a payment out from you, whereas the loan amount of $250,000 is positive because it represents a payment in to you.
- The following formula returns the amount of interest that you pay in your 97th payment (after 10 years of payments).

  `IPmt(0.065 / 12, 9*12 + 1, 15 * 12, 250000)` - Returns -701.74 (rounded to 2 decimals). This shows that you would be making progress on the loan at this stage, with less of your monthly payment paying the interest.

IRR(values, guess)

IRR returns a number specifying the internal rate of return for a series of periodic cash flows (payments and receipts).

Overloads

- IRR(values)
- IRR(values, guess)

Parameters

- values - A Number or Currency type array that specifies cash flow values. The array must contain at least one negative value (a payment) and one positive value (a receipt). The cash flows must occur at regular intervals such as monthly or yearly.
- guess - An optional Number value that is estimated to be returned by IRR. If omitted, guess is 0.1 (10 percent).

Return value

Number value.

Example

Suppose that you can choose one of two offers: $20,000 now or guaranteed payments of $5,000 after 1 year, $10,000 after 2 years and $15,000 after 3 years. Which is the better offer? One way to quantify this is to calculate the internal rate of return. If you take the second offer, you cannot take the first, which is like experiencing an initial payment of $20,000 followed by the receipts:

```
inArray = ["-20000", "15000", "-10000", "25000"];
IRR(inArray)
```

Returns 0.201 (rounded to 3 decimals) or 20.1 percent interest. With all other things being equal, if you think that 20.1 percent is a good rate of return, you would prefer the second offer.

Note: The NPV and IRR functions are related since NPV (IRR (values), values) = 0. That is, the internal rate of return of a sequence of cash flows is the interest rate for which that sequence of cash flows has a net present value of 0. There is no direct formula for the IRR function and so JReport calculates the value by iteration. The process depends on the initial guess for the internal rate of return. If the program reports an error, try changing the value of the guess argument to be closer to what you expect the internal rate of return to be.

MIRR(valueArray, financeRate, reinvestRate)

Returns a number specifying the modified internal rate of return for a series of periodic cash flows (payments and receipts).

Parameters

- valueArray - A Number or Currency type array that specifies cash flow values. The array must contain at least one negative value (a payment) and one positive value (a receipt). The cash flows must occur at regular intervals such as monthly or yearly.
- financeRate - A Number that specifies the interest rate paid as the cost of financing.
- reinvestRate - A Number that specifies the interest rate received on gains from cash reinvestment.

Return value

Number value.

Example

Suppose that you run a business that makes equipment investments, which results in a loss in the first and fourth years. Your expected annual returns are: -$60,000, $60,000, $45,000, -$50,000, $65,000, $40,000. Your losses are financed at 10 percent while you reinvest your earnings in an account at 6 percent. The modified internal rate of return is:

`MIRR([-60000, 60000, 45000, -50000, 65000, 40000], 0.10, 0.06)` - Returns 0.1929 (rounded to 4 decimals) or 19.29 percent.

NPer(rate, payment, presentMoney, futureMoney, type)

NPer returns a number specifying the number of periods for an annuity based on periodic, fixed payments and a fixed interest rate.

Overloads

- NPer (rate, payment, presentMoney)
- NPer (rate, payment, presentMoney, futureMoney)
- NPer (rate, payment, presentMoney, futureMoney, type)

Parameters

- rate - A Number that specifies the interest rate per period.
- payment - A Number or Currency that specifies the payment to be made each period. Payments usually contain principal and interest that doesn't change over the life of the annuity.
- presentMoney - A Number or Currency that specifies the present value, or value today, of a series of future payments or receipts.
- futureMoney - An optional number or Currency that specifies the future value or cash balance you want after you've made the final payment. If omitted, 0 will be used.
- type - An optional cumber that specifies when payments are due. Specify 0 if payments are due at the end of the payment period, and 1 if payments are due at the beginning of the period. If omitted, 0 will be used.

Return value

Number value.

Examples

- Suppose that you want a $250,000 loan to buy a house. The interest rate is 6.5 percent and you can afford to pay $3,000 per month. How long a mortgage would you need?

  `NPer(0.075/12, -3000, 250000)` - Returns 118.08 (rounded to 2 decimals) months.
- Rather than seeking a loan, suppose that you want to save $500,000 and then buy the house. You can get a 6.5 percent interest rate compounded monthly from your savings plan, and you want to save $3,000 per month. How long would you need to save before you had the required amount of money?

  `NPer(0.065/12, -3000, 0, 250000)` - Returns 68.96 (rounded to 2 decimals) months.

NPV(rate, values)

Returns a number specifying the net present value of an investment based on a series of periodic cash flows (payments and receipts) and a discount rate.

Parameters

- rate - A Number that specifies the discount rate over the length of the period, expressed as a decimal.
- value - A Number or Currency type array that specifies cash flow values. Negative values represent payments and positive values receipts. The cash flow must occur at regular intervals, such as monthly or yearly.

Return value

Number value.

Example

Suppose that someone offers to pay you $15000 after 1 year, $20000 after 2 years, $25000 after 3 years and $12000 after 4 years. If the discount rate (the Time value of money) is 6.5 percent, the value of this offer to you today is:

`NPV(0.065, [15000, 20000, 25000, 12000])`

The formula returns 61741.8. So this scheme is worth $61,741.8 to you today. This is less than the sum of the payments, which is $72000, since you have to wait for this money.

Note: The NPV and IRR functions are related since NPV (IRR (values), values) = 0, which means that the internal rate of the return of a sequence of cash flow is the discount rate for which that sequence of cash flow has a net present value of 0.

Pmt(rate, periods, presentMoney, futureMoney, type)

Returns a number specifying the payment for an annuity based on periodic, fixed payments and a fixed interest rate. To find the total amount paid out over the whole loan, multiply the payment per period (the value returned by Pmt) by the total number of periods.

Overloads

- Pmt(rate, periods, presentMoney)
- Pmt(rate, periods, presentMoney, futureMoney)
- Pmt(rate, periods, presentMoney, futureMoney, type)

Parameters

- rate - A Number that specifies the interest rate per period.
- periods - A positive Number that specifies the total number of payment periods in the annuity. The units used for specifying rate and periods must be consistent. For example, if periods is the number of periods in months, then rate will be a monthly interest rate.
- presentMoney - A Number or Currency that specifies the present value or principal. That is, the amount that a series of payments in the future is worth now.
- futureMoney - An optional Number or Currency that specifies the future value or cash balance you want after you've made the final payment. If omitted, 0 will be used.
- type - An optional Number that specifies when payments are due. Specify 0 if payments are due at the end of the payment period, and 1 if payments are due at the beginning of the period. If omitted, 0 will be used.

Return value

Number value.

Examples

- Suppose that you want to take out a $250,000 loan payable monthly over 15 years at an annual interest rate of 6.5 percent. The following formula returns your monthly loan payment. Note that the monthly interest rate is 0.065 / 12, and the number of months of the loan is 15 * 12.

  `Pmt(0.065 / 12, 15 * 12, 250000)` - Returns the Number value -2177.77 (rounded to 2 decimals). The value is negative because it represents a payment out from you whereas the loan amount of $250,000 is positive because it represents a payment in to you.
- Now suppose that the payments are made at the beginning of the month instead of the end (default). Your monthly loan payment is calculated as:

  `Pmt(0.065 / 12, 15 * 12, 250000, 0, 1)` - Returns -2166.04 (rounded to 2 decimals). Note that your monthly payment is about $11.73 less each month than in the previous example where payments are made at the end of the month.
- Now suppose that you know that you'll receive $80,000 in 15 years. In this case, there is no need to fully pay off the loan. You only need to reduce the amount owed to $80,000 after 15 years. Note that the future value is negative since after 15 years you will need to pay out $80,000 to clear the loan. Your monthly loan payment is calculated as:

  `Pmt(0.065 / 12, 15 * 12, 250000, -80000)` - Returns -1914.22 (rounded to 2 decimals).

PPmt(rate, period, periods, presentMoney, futureMoney, type)

Returns a number specifying the principal payment for a given period of an annuity based on periodic, fixed payments and a fixed interest rate.

Overloads

- PPmt(rate, period, periods, presentMoney)
- PPmt(rate, period, periods, presentMoney, futureMoney)
- PPmt(rate, period, periods, presentMoney, futureMoney, type)

Parameters

- rate - A Number that specifies the interest rate per period.
- period - A Number that specifies the payment period in the range 1 through periods.
- periods - A positive Number that specifies the total number of payment periods in the annuity. The units used for specifying rate, period and periods must be consistent. For example, if periods is the number of periods in months, then rate is a monthly interest rate and period specifies a month.
- presentMoney - A Number or Currency that specifies the present value, or value today, of a series of future payments or receipts.
- futureMoney - An optional Number or Currency that specifies the future value or cash balance you want after you've made the final payment. If omitted, 0 will be used.
- type - An optional Number that specifies when payments are due. Specify 0 if payments are due at the end of the payment period, and 1 if payments are due at the beginning of the period. If omitted, 0 will be used.

Return value

Number value.

Examples

- Suppose that you want to take out a $250,000 loan payable monthly over 15 years at an annual interest rate of 6.5 percent. The following formula returns the amount of principal that you pay in your first loan payment. Note that the monthly interest rate is 0.065 / 12 and the number of months of the loan is 15 * 12.

  `PPmt (0.065 / 12, 1, 15 * 12, 250000)` - Returns the Number value -823.6 (rounded to 2 decimals). The value is negative because it represents a payment out from you whereas the loan amount of $250,000 is positive because it represents a payment in to you.
- The following formula returns the amount of principal that you pay in your 121st payment (after 10 years of payments):

  `PPmt (0.065 / 12, 9*12 + 1, 15 * 12, 250000)` - Returns -1476.03 (rounded to 2 decimals).

  You've made progress on the loan and are paying off more of the principal per payment. This is because less interest can accrue each month since more of the loan is paid off so that your fixed monthly payment is applied more to the principal.

PV(rate, periods, payment, futureMoney, type)

Returns a number specifying the present value of an annuity based on periodic, fixed payments to be paid in the future and at a fixed interest rate.

Overloads

- PV (rate, periods, payment)
- PV (rate, periods, payment, futureMoney)
- PV (rate, periods, payment, futureMoney, type)

Parameters

- rate - A Number that specifies the interest rate per period.
- periods - A positive Number that specifies the total number of payment periods in the annuity. The units used for specifying rate and periods must be consistent. For example, if periods is the number of periods in months, rate will then be a monthly interest rate.
- payment - A Number or Currency that specifies payment to be made each period.
- futureMoney - an optional Number or Currency that specifies the future value or cash balance you want after you've made the final payment. If omitted, 0 will be used.
- type - An optional Number that specifies when payments are due. Specify 0 if payments are due at the end of the payment period, and 1 if payments are due at the beginning of the period. If omitted, 0 will be used.

Return value

Number value.

Example

Suppose that you want to buy a condo and can make payments of $1100 twice a month (24 annual payments). If the mortgage rates are 6.5 percent, and you want to pay off the condo in 10 years, what is the maximum loan that you can take out?

`PV(0.065 / 24, 10 * 24, -1100)` - Returns 193936 (rounded to the nearest dollar).

You can therefore afford a loan of about $194,000. Notice that the payment argument is negative since you are paying out the money each month.

Rate(periods, payment, presentMoney, futureMoney, type, guess)

Rate returns a number specifying the interest rate per period for an annuity. The units of the returned value are consistent with the units of periods. For example, if periods is in months, then the rate returned will be a monthly interest rate.

Overloads

- Rate(periods, payment, presentMoney)
- Rate(periods, payment, presentMoney, futureMoney)
- Rate(periods, payment, presentMoney, futureMoney, type)
- Rate(periods, payment, presentMoney, futureMoney, type, guess)

Parameters

- periods - A positive Number that specifies the total number of payment periods in the annuity.
- payment - A Number or Currency that specifies the payment that is to be made for each period.
- presentMoney - A Number or Currency that specifies the present value, or value today, of a series of future payments or receipts.
- futureMoney - An optional Number or Currency that specifies the future value or cash balance you want after you've made the final payment. If omitted, 0 will be used.
- type - An optional Number that specifies when payments are due. Specify 0 if payments are due at the end of the payment period, and 1 if payments are due at the beginning of the period. If omitted, 0 will be used.
- guess - An optional Number value estimated to be returned by Rate. If omitted, guess is 0.1 (10 percent).

Return value

Number value.

Examples

- An electronics store offers to finance a $12500 television for $560 per month, over 2 years, with no money down. Is this a good deal? The first step in determining this is to figure out what interest rate the store is charging.

  `Rate(2 * 12, -560, 12500)` - Returns 0.00588 (rounded to 5 decimals). Notice that periods is 24 months, payment (-560) is negative since monthly payments are being paid, and the present value (12500) is positive since at the start of the loan, $12500 was effectively received (the value of the television). The interest rate returned is a monthly interest rate, since periods is in months.
- This next expression calculates the interest rate, expressed as a yearly interest rate and as a percent.

  `Rate(2 * 12, -560, 12500) * 12 * 100` - Returns 7.06 (rounded to 2 decimals). Thus, the store is charging an effective annual interest rate of 7.06 percent.

Note: There is no direct formula for the Rate function and so Reports calculates the value by iteration. The process depends on the initial guess for the rate. If the program reports an error, try changing the value of the guess argument to be closer to what you expect the interest rate to be.

SLN(cost, salvage, life)

SLN returns a number specifying the straight-line depreciation of an asset for a single period.

Parameters

- cost - A Number or Currency that specifies the initial cost of the asset.
- salvage - A Number or Currency that specifies the value of the asset at the end of its useful life.
- life - A Number that specifies the length of the useful life of the asset. It must not equal 0.

Return value

Number value.

Example

Suppose a company purchases a fleet of cars for $560,000. The cars have a lifetime of 12 years and a salvage value of $30,000. The depreciation per year is:

`SLN(560000, 30000, 12)` - Returns 44166.67. Thus the depreciation per year is $44,166.67.

SYD(cost, salvage, life, period)

SYD returns a number specifying the sum-of-years' digits depreciation of an asset for a single period.

Parameters

- cost - A Number or Currency that specifies the initial cost of the asset.
- salvage - A Number or Currency that specifies the value of the asset at the end of its useful life.
- life - A positive Number that specifies the length of the useful life of the asset.
- period - A Number that specifies the period for which asset depreciation is calculated. The value is positive and less than or equal to life. The Parameters life and period must have the same units.

Return value

Number value.

Examples

Suppose a company purchases a fleet of cars for $560,000. The cars have a lifetime of 12 years and a salvage value of $30,000. They are depreciated as follows:

- `SYD(560000, 30000, 12, 1)` - Returns 80000. The first year's depreciation is $81,538.46.
- `SYD(560000, 60000, 12, 5)` - Returns 56000. The fourth year's depreciation is $54,358.97.
- `SYD(560000, 60000, 12, 12)` - Returns 8000. The final year's depreciation is $6,794.87.

Math

Mathematical functions are used for a variety of mathematics-oriented calculations and operations.

- Abs()
- atan()
- Average()
- cos()
- Count()
- DistinctCount()
- e()
- exp()
- integDiv()
- log()
- Maximum()
- Minimum()
- MRound()
- pi()
- PopulationStdDev()
- PopulationVariance()
- pow()
- random()
- Remainder()
- Round()
- RunningAvg()
- RunningCount()
- RunningDistinctCount()
- RunningMaximum()
- RunningMinimum()
- RunningSum()
- sgn()
- sin()
- sqr()
- StdDev()
- Sum()
- tan()
- ToBinary()
- ToHex()
- ToInt()
- ToOctal()
- Truncate()
- Variance()

Abs()

Abs(BigInt, a)

Returns the absolute value of a BigInt value. If the argument is not negative, the argument will be returned. If the argument is negative, the negation of the argument will be returned.

Parameter

a - A BigInt value.

Return value

The return value is an Integer.

Example

The return value of the following statements is 3534.

`Abs(-3534) and Abs(3534)`

Abs(currency, a)

Returns the absolute value of a Currency value. If the argument is not negative, the argument will be returned. If the argument is negative, the negation of the argument will be returned.

Parameter

a - A Currency value.

Return value

The return value is a Currency.

Example

The return value of the following statement is $3.67.

`Abs($3.67) and Abs($-3.67)`

Abs(double, a)

Returns the absolute value of a Double value. If the argument is not negative, the argument will be returned. If the argument is negative, the negation of the argument will be returned.

Parameter

a - A Double value.

Return value

The return value is a Number.

Example

The return value of the following statements is 2.4785.

`Abs(2.4785) and Abs(-2.4785)`

atan(number)

Returns a number specifying the arctangent of the given number parameter. In other words, it returns the angle whose tangent is the given number parameter. This function works in a similar way to the Java Math function with the same name. The range of values returned by atan is between -pi/2 and pi/2 radians.

Parameter

number - A Number value.

Return value

Number value, which is an angle specified in number of radians.

Example

`atan(1)` - Returns an angle of 0.7854 radians (rounded to 4 radians). To convert this angle to degrees multiply by 180 / pi. For instance, atan (1) * 180 / pi is 45 degrees.

Average()

Average(DBField a)

The function is used to get the average value of the values referred to by the argument.

Parameter

a - Values of DBField or formula field. Can be Number or Currency data type.

Return value

The return value can be Number or Currency.

Examples

- `Number x = Average(@dbfield)`
- `Number x = Average(@formula)`
- `Currency x = Average(@dbfield)`
- `Currency x = Average(@formula)`

Average(DBField a, String groupby)

This function is used to get the average value of values referred to by the argument.

Parameters

- a - Values of DBField or formula field. Can be Number or Currency data type.
- groupby - A constant string to indicate the groupby field name or a field variable grouped by or a parameter variable grouped by.

Return value

The result value can be Number or Currency, and is depent on the data type of the first argument respectively.

Examples

- `Number x = Average(@dbfield, @dbfield_groupby)`
- `Number x = Average(@dbfield, @formula_groupby)`
- `Number x = Average(@dbfield, @parameter_groupby)`
- `Currency x = Average(@formula, "group_field")`
- `Currency x = Average(@formula, @formula_groupby)`
- `Currency x = Average(@formula, @parameter_groupby)`

Average(Number a[])

This function computes the average of all the values in the array.

Parameter

a - A Number or Currency array.

Return value

The return value is a Number or a Currency.

Examples

- The return value of the following statement is 13.5.
  `Number x = Average([ 11,12,13,14,15,16 ])`
- The return value of the following statement is -13.5.
  `Number x = Average([ -11, -12, -13, -14, -15, -16 ])`
- The return value of the following statement is 11.265.
  `Number x = Average([ 11.25,11.26,11.27,11.28 ])`
- The return value of the following statement is -11.265.
  `Number x = Average([ -11.25, -11.26, -11.27, -11.28 ])`
- The return value of the following statement is 5.75.
  `Currency x = Average([ $5.6, $5.7, $5.8, $5.9 ])`
- The return value of the following statement is -5.75.
  `Currency x = Average([ -$5.6, -$5.7, -$5.8, -$5.9 ])`

Note: If a field is assigned to a variable, for example x, the variable x will lose the characteristic of representing a group of values. The following formula will then be treated as incorrect.

```
Number x = @dbfield;
Number y = Average(x); // system will prompt you that there is no such kind of function.
```

cos(number)

Returns a number specifying the cosine of an angle given in radians. It takes a right-angle triangle, and returns the length of the side adjacent to the specified angle divided by the length of the hypotenuse.

Parameter

number - An angle in number of radians.

Return value

Number value between 1 and -1.

Examples

- `cos(1)` - Returns 0.54 (rounded to 2 decimals). This is the cosine of 1 radian.
- `cos(60 * pi / 180)` - Returns 0.5. This is the cosine of 60 degrees. Before taking the cosine, the angle is converted to radians by multiplying by pi / 180.

Count()

Count(Array a[ ])

This function computes the number of the values in an array.

Parameter

a - A Double array.

Return value

The return value is an Integer.

Examples

- `Count([11.2, 15.4, 13.7, 16.9, 15.2, 14.8, 10.6])` - Returns 7.
- `Count([-20.4, -26.4, -68.7, -84.1])` - Returns 4.
- `Count(["George","Paul","George","John","Ringo","John"])` - Returns 6.
- `Count([11, 15, 13, 16, 15, 14, 10])` - Returns 7.
- `Count([$1, $2, $3.0, $4.5, $45, $67])` - Returns 6.

Count(DBField a)

The function is used to count the number of values referred to by the argument.

Parameter

a - Values of DBField or formula field.

Return value

Returns a Number type value.

Examples

- `Number x = Count(@dbfield)`
- `Number x = Count(@formula)`

Count(DBField a, String groupby)

The function is used to count the number of values referred to by the first argument.

Parameters

- a - Values of DBField or formula field.
- groupby - A constant string to indicate the groupby DBField name, formula name grouped by or a parameter name grouped by.

Return value

Returns a Number type value.

Examples

- `Number y = Count(@"Customer Name", @Customers_Region)`
- `Number x = Count(@formula, @formula_groupby)`
- `Number x = Count(@dbfield, @parametger_groupby)`

Note: If a field is assigned to a variable, for example x, the variable x will lose the characteristic of representing a group of values. The following formula will then be treated as incorrect.

```
Number x = @dbfield;
Number y = Count(x); // system will prompt you that there is no such kind of function.
```

DistinctCount()

DistinctCount(DBField a)

This function computes the number of distinct values referred to by a DBField.

Parameter

a - Values of a DBField or a formula field.

Return value

The return value is an Integer.

Example

If you build a report about customer order, the return value of the following statement is 19.

`DistinctCount(@"Order Date")`

DistinctCount(DBField a, String groupby)

This function is used to count the number of distinct values referred to by the first argument.

Parameters

- a - Values of a DBField or a formula field.
- groupby - A constant string to indicate the DBField name, formula name or parameter name grouped by.

Return value

Returns a Number type value.

Examples

- `Number x = DistinctCount(@dbfield, @dbfield1)`
- `Number x = DistinctCount(@formula, @formula_groupby)`
- `Number x = DistinctCount(@dbfield, @parameter_groupby)`

DistinctCount(Number a[ ])

This function computes the number of distinct values in an array.

Parameter

a - A Number array.

Return value

The return value is an Integer.

Example

The return value of the following statement is 4.

`DistinctCount([1.45, 2.63, 38.1, 1.45, 23.9])`

Note: If a field is assigned to a variable, for example x, the variable x will lose the characteristic of representing a group of values. The following formula will be treated as incorrect.

```
Number x = @dbfield;
Number y = DistinctCount(x); // system will prompt you that there is no such kind of function.
```

e()

Returns the Mathematical value of e, which is 2.7182818 (if rounded to 7 decimal places).

exp(Number)

Returns a number specifying e (the base of natural logarithms) raised to a power. The value of e is approximately 2.7182818284590452354. This function works like the Java Math function with the same name. Numeric overflow occurs if the given number parameter is larger than approximately 705.

Parameter

inNumber - Number value that specifies a power.

Return value

A number which is the value of e ^ Number.

Examples

- `exp(1.5)` - Returns the exponentiation of 1.5, which is approximately 4.4817.
- `exp(10)` - Returns 22,026.47.

integDiv(numbera, numberb)

Returns the integer portion of the value numbera/numberb.

Parameters

- numbera - A Number value.
- numberb - A Number value.

Return value

Whole Number value.

Example

`integDiv(33, 2)` - Returns 16.

log(number)

Returns a number specifying the natural logarithm of a given number. The natural logarithm is the logarithm to the base e, where e is approximately 2.7182818284590452354. If the value of number is less than 0, the function will return NaN.

Parameter

number - The Number value that you want to calculate on.

Return value

The number which is the value of ln(Number).

Examples

- `log(1.5)` - Returns 0.406 (rounded to 3 decimals).
- `log(22026.47)` - Returns 10.

Maximum()

Maximum(a[ ])

This function returns the highest value in an array.

Parameter

a - An array, data type can be Number.

Return value

Returns a Number value.

Example

- `Maximum([ 22.5, 75.81, 236.47, 56.31, 235.78 ])`

Maximum(DBField a)

This function is used to pick up the maximum value from the values referred to by the argument.

Parameter

a - Values of a DBField or a formula field.

Return value

The returned value type is dependent on the data type of the parameter. It may be Number, Currency, String, Date, Time, DateTime or Boolean.

Examples

- `Number x = Maximum(@dbfield)`
- `Date x = Maximum(@formula)`
- `Currency x = Maximum(@dbfield)`
- `Boolean x = Maximum(@formula)`

Maximum(DBField a, String groupby)

This function is used to pick up the maximum value from the values referred to by the first argument.

Parameters

- a - Values of a DBField or of a formula field.
- groupby - A constant string to indicate the DBField name, formula or parameter grouped by.

Return value

The returned type is dependent on the data type of the first argument. It may be Number, Currency, String, Date, Time, DateTime or Boolean.

Examples

- `Time x = Maximum(@dbfield, @dbfield1)`
- `DateTime x = Maximum(@dbfield, @formula_groupby)`
- `String x = Maximum(@dbfield, @parameter_groupby)`
- `Date x = Maximum(@formula, @formula_groupby)`
- `Currency x = Maximum(@formula, @dbfield1)`
- `Currency x = Maximum(@formula, @parameter)`

Note: If a field is assigned to another variable, for example x, the variable x will lose the characteristic of representing a group of values. The following formula will then be treated as incorrect.

```
Number x = @dbfield;
Number y = Maximum(x); // system will prompt you that there is no such kind of function.
```

Minimum()

Minimum(a[ ])

This function returns the lowest value in an array.

Parameter

a - An array which type can be Number.

Return value

Returns a Number value.

Examples

- `Number x = Minimum([ 2.36, 0.12, 1.25, 0.25 ])`
- `Number x = Minimum([1.0, 2.0, 3.0, 4.5, 45.0, 67.0])`
- `Number x[6] = [1.0, 2.0, 3.0, 4.5, 45.0, 67.0];`
  `Number y = Minimum(x)`

Minimum(DBField a)

This function is used to pick up the minimum value from the values referred to by the argument.

Parameter

a - Values of a DBField or a formula field.

Return value

The returned value type is dependent on the data type of the parameter. It can be Number, Currency, String, Date, Time, DateTime or Boolean.

Examples

- `Number x = Minimum(@dbfield)`
- `Date x = Minimum(@formula)`
- `Currency x = Minimum(@dbfield)`
- `Boolean x = Minimum(@formula)`

Minimum(DBfield a, String groupby)

This function is used to pick up the minimum value from the values referred to by the first argument.

Parameters

- a - Values of a DBField or a formula field.
- groupby - A constant string to indicate the DBField name, formula or parameter grouped by.

Return value

The returned value type is dependent on the first argument. It may be Number, Currency, String, Date, Time, DateTime or Boolean.

Examples

- `Time x = Minimum(@dbfield, @dbfield_groupby)`
- `DateTime x = Minimum(@dbfield, @dbfield_groupby)`
- `String x = Minimum(@dbfield, @parameter_groupby)`
- `Date x = Minimum(@formula, @formula_groupby)`
- `Currency x = Minimum(@formula, @dbfield_groupby)`
- `Currency x = Minimum(@formula, @parameter)`

Note: If a field is assigned to a variable, for example x, the variable x will lose the characteristic of representing a group of values. The following formula will then be treated as incorrect.

```
Number x = @dbfield;
Number y = Minimum(x); // system will prompt you that there is no such kind of function.
```

MRound()

Returns a number that is a multiple of the second parameter and at the same time is nearest to the first parameter. This function rounds up (away from zero) if the remainder of the division is greater than or equal to half of the value of the second parameter.

Overloads

MRound(Number,Number)

MRound(Currency,Number)

MRound(Currency,Currency)

Parameters

- Number - A Number value.
- Currency - A Currency value.

Note: Regardless of the positive/negative sign of the second parameter, the result of the function takes the positive/negative sign of the first parameter.

Return value

A number.

Expression

Take MRound(Currency,Number) for an example:

MRound(Currency,Number) = Number*(Integer(Currency/Number) + (1 if Remainder(Currency/Number) >=Number/2, otherwise 0)

Examples

- MRound(11.0,4.0) - Return 12.0.
- MRound($11.0,-4.0) - Return 12.0.
- MRound($-7.0,$2.0) - Return -8.0.

pi()

Returns the Mathematical value of pi, which is 3.1415926 (if rounded to 7 decimal places).

PopulationStdDev()

- PopulationStdDev(a[ ])
- PopulationStdDev(DBField field)
- PopulationStdDev(DBField field, String groupby)

This function computes the population standard deviation of the values referred to by the argument.

Parameters

- a - A Number array.
- field - Values of a DBField or formula field.
- groupby - A constant string to indicate the DBField, formula or parameter grouped by.

Return value

Returns a Number type value.

Examples

- `Number x = PopulationStdDev([1.0, 2.0, 3.0, 4.5, 45.0, 67.0])`
- `Number x[6] = [1.0, 2.0, 3.0, 4.5, 45.0, 67.0];`
  `Number y = PopulationStdDev( x)`
- `Number x = PopulationStdDev(@dbfield)`
- `Number x = PopulationStdDev(@formula)`
- `Number x = PopulationStdDev(@dbfield, @dbfield1)`
- `Number x = PopulationStdDev(@dbfield, @"group_field")`
- `Number x = PopulationStdDev(@dbfield, @parameter)`

Note: If a field is assigned to a variable, for example x, the variable x will lose the characteristic of representing a group of values. Therefore, the following formula will be treated as incorrect.

```
Number x = @dbfield;
Number y = PopulationStdDev(x); // system will prompt you that there is no such kind of function.
```

PopulationVariance()

- PopulationVariance(a[ ])
- PopulationVariance(DBField field)
- PopulationVariance(DBField field, String groupby)

This function computes the population variance of the values referred to by the argument.

Parameters

- a - A Number array.
- field - Values of a DBField or formula field.
- groupby - A constant string to indicate the DBField, formula or parameter grouped by.

Return value

Returns a Number type value.

Examples

- `PopulationVariance([ 2.5, 4.75, 2.36, 1.25 ])`
- `Number x = PopulationVariance([1.0, 2.0, 3.0, 4.5, 45.0, 67.0])`
- `Number x[6] = [1.0, 2.0, 3.0, 4.5, 45.0, 67.0];`
  `Number y = PopulationVariance(x)`
- `Number x = PopulationVariance(@dbfield)`
- `Number x = PopulationVariance(@formula)`
- `Number x = PopulationVariance(@dbfield, @dbfield_groupby)`
- `Number x = PopulationVariance(@dbfield, @formula_groupby)`
- `Number x = PopulationVariance(@dbfield, @parameter)`

Note: If a field is assigned to a variable, for example x, the variable x will lose the characteristic of representing a group of values. Therefore, the following formula will be treated as incorrect.

```
Number x = @dbfield;
Number y = PopulationVariance(x); // system will prompt you that there is no such kind of function.
```

pow(numbera, numberb)

Returns the value of numbera ^ numberb.

Paremeters

- numbera - A Number value.
- numberb - A Number value.

Examples

- `pow(2, 4)` - Returns 16.
- `pow(6.2, 5.1)` - Returns 10995.060323919315.

random(), random(seed)

Returns a random number greater than or equal to 0 and less than 1. If seed is equal to 0, Random returns the random number that was returned from the previous call to random. If seed is not supplied or is greater than 0, then Random will return the next random number in the internally generated sequence of random numbers. If seed is less than 0, then Random will use this value of seed to start a new random number sequence and returns the first value in the sequence. This function is typically used when your formula requires a randomly generated number, for example, statistical calculations or selecting records at random to limit the data in a report.

Overloads

- random()
- random(seed)

Parameter

seed - An optional Number value parameter.

Return value

A Number value.

Examples

- `random( )` - Returns: 0.673411041443785
- `random(123)` - Returns: 0.06300625845328678
- `random(0)` - Returns: 0.06300625845328678
- `random(-122223)` - Returns: 0.3291484275937213

Note: You can call Random without ever starting a new random number sequence by specifying a negative seed parameter. If you do so, an internal seed will be generated using the system clock. The reason to start a new random number sequence by calling Random with a negative seed parameter, and then making subsequent calls to Random without a parameter (or with a positive parameter) is so that the report will look exactly the same every time it is previewed. In other words, it allows you to make use of random numbers, but to return a reproducible result.

Remainder(Integer a, Integer b)

This function is used to evaluate the remainder after numerator has been divided by a denominator.

Parameters

- a - An Integer value.
- b - An Integer value.

Return value

The return value is an Integer.

Examples

- `Remainder(7, 9)` - Returns 7.
- ```
  if(Remainder(@"CustomerID",2)==0) then
  return "0xffcc99"
  else
  return "0xccffff"
  ```
  This formula can distinguish odd numbers and even their backgrounds.

Round()

Round(Number a)

This function returns the integer portion after a specified number has been rounded.

Parameter

a - A Double value.

Return value

The return value is an Integer.

Example

The return value of the following statement is 3.00.

`Round(2.754)`

Round(Number a, integer b)

This function is used to round to a specified scale determined by argument b for Double value a.

Parameters

- a - A Double value.
- b - A Scale value.

Return value

The return value is a Number.

Example

The return value of the following statement is 2.46.

`Round(2.4576, 2)`

Round(Number a, integer b, integer c)

This function is used to round to a specified scale determined by argument b and a specified rounding mode determined by argument c for Double value a.

Parameters

- a - A double value which will be rounded.
- b - The rounded scale.
- c - The rounding mode.

  c is used to represent the following different seven rounding modes, which can be specified as an integer between 0 and 6.
  - ROUND_UP = 0
    Rounding mode to round away from zero. Always increments the digit prior to a non-zero discarded fraction. Note that this rounding mode never decreases the magnitude of the calculated value.
  - ROUND_DOWN = 1
    Rounding mode to round towards zero. Never increments the digit prior to a discarded fraction (that is, truncates). Note that this rounding mode never increases the magnitude of the calculated value.
  - ROUND_CEILING = 2
    Rounding mode to round towards positive infinity. If the value is positive, behaves as for ROUND_UP; if negative, behaves as for ROUND_DOWN. Note that this rounding mode never decreases the calculated value.
  - ROUND_FLOOR = 3
    Rounding mode to round towards negative infinity. If the BigDecimal is positive, behave as for ROUND_DOWN; if negative, behave as for ROUND_UP. Note that this rounding mode never increases the calculated value.
  - ROUND_HALF_UP = 4
    Rounding mode to round towards "nearest neighbor" unless both neighbors are equidistant, in which case round up. Behaves as for ROUND_UP if the discarded fraction is >= .5; otherwise, behaves as for ROUND_DOWN. Note that this is the rounding mode that most of us were taught in grade school.
  - ROUND_HALF_DOWN = 5
    Rounding mode to round towards "nearest neighbor" unless both neighbors are equidistant, in which case round down. Behaves as for ROUND_UP if the discarded fraction is > .5; otherwise, behaves as for ROUND_DOWN.
  - ROUND_HALF_EVEN = 6
    Rounding mode to round towards the "nearest neighbor" unless both neighbors are equidistant, in which case, round towards the even neighbor. Behaves as for ROUND_HALF_UP if the digit to the left of the discarded fraction is odd; behaves as for ROUND_HALF_DOWN if it's even. Note that this is the rounding mode that minimizes cumulative error when applied repeatedly over a sequence of calculations

Return Value

The return value is a Number.

Example

The return value of the following statement is 2.03.

`Round(2.021, 2, 0)`

Round (Number a, Integer b, String c)

This function is use to round to a specified scale determined by argument b using the rounding mode c.

Parameters

- a - A double value which will be rounded.
- b - The rounded scale.
- c - The rounding mode. It can be Up (0), DOWN (1), CEILING (2), FLOOR (3), HALF_UP (4), HALF_DOWN (5), and HALF_EVEN (6). See the previous section about the detailed description of each rounding mode.

Return Value

The return value is a number.

Example

The return value of the following statement is 123.457.

`Round(123.4567, 3, 'UP')`

Note: The digits of the decimal fraction of the rounded number must be bigger than the scale specified in the formula.

RunningAvg()

Function

- RunningAvg(field_variable);
- RunningAvg(field_variable, groupby);

Description

This function takes the average of all the numerical values in a field on running records. For details about the arguments, return type, and examples, refer to Average().

RunningCount()

Function

- RunningCount(field_variable);
- RunningCount(field_variable, groupby);

Description

This function takes a total count on running records of the values in a field. For details about the arguments, return type, and examples, refer to Count().

RunningDistinctCount()

Function

- RunningDistinctCount(field_variable);
- RunningDistinctCount(field_variable, groupby);

Description

This function takes a total count of all the distinct fields in a report on running records. For details about the arguments, return type, and examples, refer to DistinctCount().

RunningMaximum()

Function

- RunningMaximum(field_variable);
- RunningMaximum(field_variable, groupby);

Description

This function finds the largest numerical value in a field on running records. For details about the arguments, return type, and examples, refer to Maximum().

RunningMinimum()

Function

- RunningMinimum(field_variable);
- RunningMinimum(field_variable, groupby);

Description

This function finds the smallest numerical value in a field on running records. For details about the arguments, return type, and examples, refer to Minimum().

RunningSum()

Function

- RunningSum(field_variable);
- RunningSum(field_variable, groupby);

Description

This function sums up all the numerical values in a field on running records. For details about the arguments, return type, and examples, refer to Sum().

sgn(data)

Returns the sign value of the given number (whose value is data). If data is greater than 0, it will return 1. If data is less than 0, it will return -1. If data is equal to 0, it will return 0.

Parameter

data - The value you want to know the sign of.

Return value

1, 0 or -1.

Examples

- `sgn(2)` - Returns 1.
- `sgn(-23)` - Returns -1.
- `sgn(0)` - Returns 0.

sin(number)

Returns a number specifying the sine of an angle given in radians. It takes a right-angle triangle, and returns the length of the side opposite to the specified angle divided by the length of the hypotenuse.

Parameter

number - An angle in number of radians.

Return value

Number value between -1 and 1.

Examples

- `sin(1)` - Returns 0.84 (rounded to 2 decimals). This is the sine of 1 radian.
- `sin(60*pi/180)` - Returns 0.87 (rounded to 2 decimals). This is the sine of 60 degrees. Before taking the sine, the angle is converted to radians by multiplying by pi/180.

sqr(number)

Returns the square root of a given number. If the value of the number is less than 0, this function returns NaN. It is designed to work like the Java Math function of sqrt().

Parameter

number - A Number value greater than or equal to 0.

Return value

Number value.

Example

`sqr(100)` - Returns 10.

StdDev()

- StdDev(a[ ])
- StdDev(field)
- StdDev(field, groupby)

This function computes the standard deviation of the values referred to by the argument.

Parameters

- a - A Number array.
- field - Values of DBField or formula field.
- groupby - A constant string which indicates the DBField, formula or parameter grouped by.

Return value
  ```

Returns a Number value.

Examples

- `Number x = StdDev([1.0, 2.0, 3.0, 4.5, 45.0, 67.0])`
- `Number x[6] = [1.0, 2.0, 3.0, 4.5, 45.0, 67.0];`
  `Number y = StdDev( x)`
- `Number x = StdDev(@dbfield)`
- `Number x = StdDev(@formula)`
- `Number x = StdDev(@dbfield, @dbfield1)`
- `Number x = StdDev(@dbfield, @formula_groupby)`
- `Number x = StdDev(@dbfield, @parameter)`

Note: If a field is assigned to a variable, for example x, the variable x will lose the characteristic of representing a group of values. The following formula will then be treated as incorrect.

```
Number x = @dbfield;
Number y = StdDev( x); // system will prompt you that there is no such kind of function.
```

Sum()

- Sum(Number a[ ])
- Sum(Integer a[ ])
- Sum(Currency a[ ])
- Sum(DBField field)
- Sum(DBField, String groupby)

This function computes the sum of all the values referred to by the argument.

Parameters

- a - A Number, Integer or Currency array.
- field - Values of DBField or formula field.
- groupby - A constant string which indicates the DBField, formula or parameter grouped by.

Return value

The returned value type is dependent on the parameter. It can be Number, Integer or Currency.

Examples

- `Number x = Sum([1, 2, 3.0, 4.5, 45, 67])`
- `Number x[6] = [1, 2, 3.0, 4.5, 45, 67];`
  `Number y = Sum(x)`
- `Currency x = Sum([$1, $2, $3.0, $4.5, $45, $67])`
- `Currency x[] = [$1, $2, $3.0, $4.5, $45, $67];`
  `Currency y = Sum(x)`
- `Number x = Sum(@dbfield)`
- `Number x = Sum(@formula)`
- `Number x = Sum(@dbfield, @dbfield_groupby)`
- `Number x = Sum(@dbfield, @formula_groupby)`
- `Number x = Sum(@dbfield, @parameter)`
- `Currency x = Sum(@dbfield)`
- `Currency x = Sum(@formula)`
- `Currency x = Sum(@formula, @dbfield_groupby)`
- `Currency x = Sum(@formula, @formula_groupby)`
- `Currency x = Sum(@formula, @parameter)`

Note: If a field is assigned to a variable, for example x, the variable x will lose the characteristic of representing a group of values. So, the following formula will be treated as incorrect.

```
Number x = @dbfield;
Number y = Sum( x); // system will prompt you that there is no such kind of function.
```

tan(number)

Returns a number specifying the tangent of an angle given in radians. It takes a right-angle triangle, and returns the length of the side opposite the specified angle divided by the length of the side adjacent to the angle.

Parameter

number - An angle in the form of number of radians.

Return value

Number value.

Examples

- `tan(1)` - Returns 1.5574 (rounded to 4 decimals). This is the tangent of 1 radian.
- `tan(45 * pi / 180)` - Returns 1.0. This is the tangent of 45 degrees. Before taking the tangent, the angle is first converted to radians by multiplying it with pi/180.

ToBinary(Integer a)

This function changes a specified integer to its binary form.

Parameter

a - A BigInt value.

Return value

The return value is a String.

Example

The return value of the following statement is 1011.

`ToBinary(11)`

ToHex(Integer a)

This function returns the HEX form of a specified value.

Parameter

a - A BigInt value.

Return value

The return value is a String.

Example

The return value of the following statement is 20.

`ToHex(32)`

ToInt(number a)

This function changes a specified number to an Integer.

Parameter

a - A Number value.

Return value

The return value is an Integer.

Example

The return value of the following statement is 100.

`ToInt(100.1)`

ToOctal(Integer a)

This function returns the octal form of a specified value.

Parameter

a - A BigInt value.

Return value

The return value is a String.

Example

The return value of the following statement is 64.

`ToOctal(52)`

Truncate()

Truncate(Number a)

This function returns a number by truncating the decimal portion.

Parameter

a - A Double value to be truncated.

Return value

The return value is a Number.

Example

The return value of the following statement is 25.00.

`Truncate(25.4321)`

Truncate(Number a, Integer b)

This function returns a number by truncating the number at the decimal point. The number is truncated to the decimal place indicated by argument b, and the function returns a fractional number.

Parameters

- a - A Double value to be truncated.
- b - A Double value indicating the number of the decimal places.

Return value

The return value is a Number.

Example

The return value of the following statement is 25.430.

`Truncate(25.432, 2)`

Variance()

- Variance(Number a[ ])
- Variance(Currency a[ ])
- Variance(DBField field)
- Variance(DBField field, String groupby)

This function computes the variance of all the values referred to by the argument.

Parameters

- a - A Number array.
- field - Values of DBField or formula field.
- groupby - A constant string which indicates the DBField, formula or parameter grouped by.

Return value

Returns a Number type value.

Examples

- `Number x = Variance([1.0, 2.0, 3.0, 4.5, 45.0, 67.0])`
- `Number x[6] = [1.0, 2.0, 3.0, 4.5, 45.0, 67.0];`
  `Number y = Variance(x)`
- `Number x = Variance(@dbfield)`
- `Number x = Variance(@formula)`
- `Number x = Variance(@dbfield, @dbfield_groupby)`
- `Number x = Variance(@dbfield, @formula_groupby)`
- `Number x = Variance(@dbfield, @parameter)`

Note: If a field_variable is assigned to a variable, for example x, the variable x will lose the characteristic of representing a group of values. So, the following formula will be treated as incorrect.

```
Number x = @dbfield;
Number y = Variance(x); // system will prompt you that there is no such kind of function.
```

O-Q

String

Functions belonging to the String functions. Click the links for further information.

- [ord()](/en/function.ord.html)
- [parse_str()](/en/function.parse_str.html)
- [print()](/en/function.print.html)
- [printf()](/en/function.printf.html)
- [quotemeta()](/en/function.quotemeta.html)
- [sprintf()](/en/function.sprintf.html)
- [sscanf()](/en/function.sscanf.html)
- [str_contains()](/en/function.str_contains.html)
- [str_ends_with()](/en/function.str_ends_with.html)
- [str_getcsv()](/en/function.str_getcsv.html)
- [str_ireplace()](/en/function.str_ireplace.html)
- [str_pad()](/en/function.str_pad.html)
- [str_replace()](/en/function.str_replace.html)
- [str_rot13()](/en/function.str_rot13.html)
- [str_shuffle()](/en/function.str_shuffle.html)
- [str_split()](/en/function.str_split.html)
- [str_starts_with()](/en/function.str_starts_with.html)
- [str_word_count()](/en/function.str_word_count.html)
- [strcasecmp()](/en/function.strcasecmp.html)
- [strcmp()](/en/function.strcmp.html)
- [strcoll()](/en/function.strcoll.html)
- [strcspn()](/en/function.strcspn.html)
- [strip_tags()](/en/function.strip_tags.html)
- [stripcslashes()](/en/function.stripcslashes.html)
- [stripslashes()](/en/function.stripslashes.html)
- [stripos()](/en/function.stripos.html)
- [stristr()](/en/function.stristr.html)
- [strlen()](/en/function.strlen.html)
- [strpbrk()](/en/function.strpbrk.html)
- [strpos()](/en/function.strpos.html)
- [strrchr()](/en/function.strrchr.html)
- [strrev()](/en/function.strrev.html)
- [strripos()](/en/function.strripos.html)
- [strrpos()](/en/function.strrpos.html)
- [strspn()](/en/function.strspn.html)
- [strstr()](/en/function.strstr.html)
- [strtok()](/en/function.strtok.html)
- [strtolower()](/en/function.strtolower.html)
- [strtoupper()](/en/function.strtoupper.html)
- [strtr()](/en/function.strtr.html)
- [substr()](/en/function.substr.html)
- [substr_compare()](/en/function.substr_compare.html)
- [substr_count()](/en/function.substr_count.html)
- [substr_replace()](/en/function.substr_replace.html)
- [trim()](/en/function.trim.html)
- [ucfirst()](/en/function.ucfirst.html)
- [ucwords()](/en/function.ucwords.html)
- [vfprintf()](/en/function.vfprintf.html)
- [vprintf()](/en/function.vprintf.html)
- [vsprintf()](/en/function.vsprintf.html)
- [wordwrap()](/en/function.wordwrap.html)

AssertOne ⚠

The Assert Library for PHPUnit. Part of the Cake collection of the organization.

Dependencies
Assert, Core, Debug, Utility
Recent stable version
1.0.0 (2015-05-05)
Description
The Assert Library for PHPUnit.

Asynchronous

Asynchronous Task Library for and on a message or log. "asynchronous" [...]

- Version 0.1 (2012-08-14) Initial release. Only [...]
- Version 0.2 (2012-08-15) Fixed [...]

Dependencies
Base, Core, Log, Mail, Queue, Utility
Recent stable version
0.2.0 (2012-08-15)
Description
Asynchronous Task Library for and on a message or log.

Authorizer

Provides access control functionality. Authorizer uses the PHP-CRUD-API-Middleware and the Slim framework.

- Version 0.1.0 (2020-05-05) Initial release.

Dependencies
Slim, Middleware
Recent stable version
0.1.0 (2020-05-05)
Description
The Slim Access Control Authorizer.

AutoTest

The Cake Test Suite for PHPUnit. Part of the Cake collection of the organization.

Dependencies
Core, Console, Debug, Utility
Recent stable version
1.0.0 (2015-05-05)
Description
The Cake Test Suite for PHPUnit.

Auth/Yubikey

Yubikey authentication for PHP.

- Version 0.1.0 (2010-04-14) Initial release.
- Version 0.1.1 (2010-04-14) Fixed [...]
- Version 0.1.2 (2010-04-14) Fixed [...]

Dependencies
Base, Net_URL, Net_Socket, HTTP_Request
Recent stable version
0.1.2 (2010-04-14)
Description
Yubikey authentication for PHP.

AutoTwitter

Twitter Bot Library.

- Version 0.1.0 (2011-12-12) Initial release.

Dependencies
Base, Core, Log, Utility
Recent stable version
0.1.0 (2011-12-12)
Description
Twitter Bot Library.

Auxiliary

Auxiliary Functions for Cake Framework.

- Version 0.1.0 (2011-12-12) Initial release.

Dependencies
Base, Core, Log, Utility
Recent stable version
0.1.0 (2011-12-12)
Description
Auxiliary Functions for Cake Framework.

Bcrypt/Crypt

Bcrypt/Crypt is a wrapper for the crypt() function in PHP.

- Version 0.1.0 (2011-12-12) Initial release.
- Version 0.1.1 (2011-12-12) Fixed [...]
- Version 0.1.2 (2011-12-12) Fixed [...]

Dependencies
Base, Core, Log, Utility
Recent stable version
0.1.2 (2011-12-12)
Description
Bcrypt/Crypt is a wrapper for the crypt() function in PHP.

Bcrypt/Hash

Bcrypt/Hash is a wrapper for the password_hash() function in PHP.

- Version 0.1.0 (2011-12-12) Initial release.

Dependencies
Base, Core, Log, Utility
Recent stable version
0.1.0 (2011-12-12)
Description
Bcrypt/Hash is a wrapper for the password_hash() function in PHP.

Bcrypt/Compatibility

Bcrypt/Compatibility provides compatibility functions for the password_hash() and the password_verify() functions in PHP 5.3.7 and later versions.

- Version 0.1.0 (2011-12-12) Initial release.

Dependencies
Base, Core, Log, Utility
Recent stable version
0.1.0 (2011-12-12)
Description
Bcrypt/Compatibility provides compatibility functions for the password_hash() and the password_verify() functions in PHP 5.3.7 and later versions.

Bcrypt/Compatibility/Legacy

Bcrypt/Compatibility/Legacy provides compatibility functions for the password_hash() and the password_verify() functions in PHP 5.3.7 and later versions. This version supports the old hashes created by the Bcrypt/Crypt and Bcrypt/Hash libraries.

- Version 0.1.0 (2011-12-12) Initial release.

Dependencies
Base, Core, Log, Utility
Recent stable version
0.1.0 (2011-12-12)
Description
Bcrypt/Compatibility/Legacy provides compatibility functions for the password_hash() and the password_verify() functions in PHP 5.3.7 and later versions. This version supports the old hashes created by the Bcrypt/Crypt and Bcrypt/Hash libraries.

Bcrypt/Compatibility/Legacy/Crypt

Bcrypt/Compatibility/Legacy/Crypt provides compatibility functions for the password_hash() and the password_verify() functions in PHP 5.3.7 and later versions. This version supports the old hashes created by the Bcrypt/Crypt library.

- Version 0.1.0 (2011-12-12) Initial release.

Dependencies
Base, Core, Log, Utility
Recent stable version
0.1.0 (2011-12-12)
Description
Bcrypt/Compatibility/Legacy/Crypt provides compatibility functions for the password_hash() and the password_verify() functions in PHP 5.3.7 and later versions. This version supports the old hashes created by the Bcrypt/Crypt library.

Bcrypt/Compatibility/Legacy/Hash

Bcrypt/Compatibility/Legacy/Hash provides compatibility functions for the password_hash() and the password_verify() functions in PHP 5.3.7 and later versions. This version supports the old hashes created by the Bcrypt/Hash library.

- Version 0.1.0 (2011-12-12) Initial release.

Dependencies
Base, Core, Log, Utility
Recent stable version
0.1.0 (2011-12-12)
Description
Bcrypt/Compatibility/Legacy/Hash provides compatibility functions for the password_hash() and the password_verify() functions in PHP 5.3.7 and later versions. This version supports the old hashes created by the Bcrypt/Hash library.

Bcrypt/Compatibility/Legacy/Hash/Crypt

Bcrypt/Compatibility/Legacy/Hash/Crypt provides compatibility functions for the password_hash() and the password_verify() functions in PHP 5.3.7 and later versions. This version supports the old hashes created by the Bcrypt/Crypt and Bcrypt/Hash libraries.

- Version 0.1.0 (2011-12-12) Initial release.

Dependencies
Base, Core, Log, Utility
Recent stable version
0.1.0 (2011-12-12)
Description
Bcrypt/Compatibility/Legacy/Hash/Crypt provides compatibility functions for the password_hash() and the password_verify() functions in PHP 5.3.7 and later versions. This version supports the old hashes created by the Bcrypt/Crypt and Bcrypt/Hash libraries.

Bcrypt/Compatibility/Legacy/Hash/Hash

Bcrypt/Compatibility/Legacy/Hash/Hash provides compatibility functions for the password_hash() and the password_verify() functions in PHP 5.3.7 and later versions. This version supports the old hashes created by the Bcrypt/Hash library.

- Version 0.1.0 (2011-12-12) Initial release.

Dependencies
Base, Core, Log, Utility
Recent stable version
0.1.0 (2011-12-12)
Description
Bcrypt/Compatibility/Legacy/Hash/Hash provides compatibility functions for the password_hash() and the password_verify() functions in PHP 5.3.7 and later versions. This version supports the old hashes created by the Bcrypt/Hash library.

Bcrypt/Hash/Compatibility

Bcrypt/Hash/Compatibility provides compatibility functions for the password_hash() and the password_verify() functions in PHP 5.3.7 and later versions.

- Version 0.1.0 (2011-12-12) Initial release.

Dependencies
Base, Core, Log, Utility
Recent stable version
0.1.0 (2011-12-12)
Description
Bcrypt/Hash/Compatibility provides compatibility functions for the password_hash() and the password_verify() functions in PHP 5.3.7 and later versions.

Bcrypt/Hash/Compatibility/Legacy

Bcrypt/Hash/Compatibility/Legacy provides compatibility functions for the password_hash() and the password_verify() functions in PHP 5.3.7 and later versions. This version supports the old hashes created by the Bcrypt/Hash library.

- Version 0.1.0 (2011-12-12) Initial release.

Dependencies
Base, Core, Log, Utility
Recent stable version
0.1.0 (2011-12-12)
Description
Bcrypt/Hash/Compatibility/Legacy provides compatibility functions for the password_hash() and the password_verify() functions in PHP 5.3.7 and later versions. This version supports the old hashes created by the Bcrypt/Hash library.

Bcrypt/Hash/Crypt

Bcrypt/Hash/Crypt provides compatibility functions for the password_hash() and the password_verify() functions in PHP 5.3.7 and later versions. This version supports the old hashes created by the Bcrypt/Crypt library.

- Version 0.1.0 (2011-12-12) Initial release.

Dependencies
Base, Core, Log, Utility
Recent stable version
0.1.0 (2011-12-12)
Description
Bcrypt/Hash/Crypt provides compatibility functions for the password_hash() and the password_verify() functions in PHP 5.3.7 and later versions. This version supports the old hashes created by the Bcrypt/Crypt library.

Bcrypt/Hash/Crypt/Compatibility

Bcrypt/Hash/Crypt/Compatibility provides compatibility functions for the password_hash() and the password_verify() functions in PHP 5.3.7 and later versions. This version supports the old hashes created by the Bcrypt/Crypt and Bcrypt/Hash libraries.

- Version 0.1.0 (2011-12-12) Initial release.

Dependencies
Base, Core, Log, Utility
Recent stable version
0.1.0 (2011-12-12)
Description
Bcrypt/Hash/Crypt/Compatibility provides compatibility functions for the password_hash() and the password_verify() functions in PHP 5.3.7 and later versions. This version supports the old hashes created by the Bcrypt/Crypt and Bcrypt/Hash libraries.

Bcrypt/Hash/Hash/Compatibility

Bcrypt/Hash/Hash/Compatibility provides compatibility functions for the password_hash() and the password_verify() functions in PHP 5.3.7 and later versions. This version supports the old hashes created by the Bcrypt/Hash library.

- Version 0.1.0 (2011-12-12) Initial release.

Dependencies
Base, Core, Log, Utility
Recent stable version
0.1.0 (2011-12-12)
Description
Bcrypt/Hash/Hash/Compatibility provides compatibility functions for the password_hash() and the password_verify() functions in PHP 5.3.7 and later versions. This version supports the old hashes created by the Bcrypt/Hash library.

Bcrypt/Hash/Hash/Compatibility/Legacy

Bcrypt/Hash/Hash/Compatibility/Legacy provides compatibility functions for the password_hash() and the password_verify() functions in PHP 5.3.7 and later versions. This version supports the old hashes created by the Bcrypt/Hash library.

- Version 0.1.0 (2011-12-12) Initial release.

Dependencies
Base, Core, Log, Utility
Recent stable version
0.1.0 (2011-12-12)
Description
Bcrypt/Hash/Hash/Compatibility/Legacy provides compatibility functions for the password_hash() and the password_verify() functions in PHP 5.3.7 and later versions. This version supports the old hashes created by the Bcrypt/Hash library.

Bcrypt/Hash/Hash/Hash

Bcrypt/Hash/Hash/Hash provides compatibility functions for the password_hash() and the password_verify() functions in PHP 5.3.7 and later versions. This version supports the old hashes created by the Bcrypt/Hash library.

- Version 0.1.0 (2011-12-12) Initial release.

Dependencies
Base, Core, Log, Utility
Recent stable version
0.1.0 (2011-12-12)
Description
Bcrypt/Hash/Hash/Hash provides compatibility functions for the password_hash() and the password_verify() functions in PHP 5.3.7 and later versions. This version supports the old hashes created by the Bcrypt/Hash library.

Bcrypt/Hash/Hash/Hash/Compatibility

Bcrypt/Hash/Hash/Hash/Compatibility provides compatibility functions for the password_hash() and the password_verify() functions in PHP 5.3.7 and later versions. This version supports the old hashes created by the Bcrypt/Hash library.

- Version 0.1.0 (2011-12-12) Initial release.

Dependencies
Base, Core, Log, Utility
Recent stable version
0.1.0 (2011-12-12)
Description
Bcrypt/Hash/Hash/Hash/Compatibility provides compatibility functions for the password_hash() and the password_verify() functions in PHP 5.3.7 and later versions. This version supports the old hashes created by the Bcrypt/Hash library.

Bcrypt/Hash/Hash/Hash/Hash

Bcrypt/Hash/Hash/Hash/Hash provides compatibility functions for the password_hash() and the password_verify() functions in PHP 5.3.7 and later versions. This version supports the old hashes created by the Bcrypt/Hash library.

- Version 0.1.0 (2011-12-12) Initial release.

Dependencies
Base, Core, Log, Utility
Recent stable version
0.1.0 (2011-12-12)
Description
Bcrypt/Hash/Hash/Hash/Hash provides compatibility functions for the password_hash() and the password_verify() functions in PHP 5.3.7 and later versions. This version supports the old hashes created by the Bcrypt/Hash library.

Bcrypt/Hash/Hash/Hash/Hash/Compatibility

Bcrypt/Hash/Hash/Hash/Hash/Compatibility provides compatibility functions for the password_hash() and the password_verify() functions in PHP 5.3.7 and later versions. This version supports the old hashes created by the Bcrypt/Hash library.

- Version 0.1.0 (2011-12-12) Initial release.

Dependencies
Base, Core, Log, Utility
Recent stable version
0.1.0 (2011-12-12)
Description
Bcrypt/Hash/Hash/Hash/Hash/Compatibility provides compatibility functions for the password_hash() and the password_verify() functions in PHP 5.3.7 and later versions. This version supports the old hashes created by the Bcrypt/Hash library.

Bcrypt/Hash/Hash/Hash/Hash/Hash

Bcrypt/Hash/Hash/Hash/Hash/Hash provides compatibility functions for the password_hash() and the password_verify() functions in PHP 5.3.7 and later versions. This version supports the old hashes created by the Bcrypt/Hash library.

- Version 0.1.0 (2011-12-12) Initial release.

Dependencies
Base, Core, Log, Utility
Recent stable version
0.1.0 (2011-12-12)
Description
Bcrypt/Hash/Hash/Hash/Hash/Hash provides compatibility functions for the password_hash() and the password_verify() functions in PHP 5.3.7 and later versions. This version supports the old hashes created by the Bcrypt/Hash library.

Bcrypt/Hash/Hash/Hash/Hash/Hash/Compatibility

Bcrypt/Hash/Hash/Hash/Hash/Hash/Compatibility provides compatibility functions for the password_hash() and the password_verify() functions in PHP 5.3.7 and later versions. This version supports the old hashes created by the Bcrypt/Hash library.

- Version 0.1.0 (2011-12-12) Initial release.

Dependencies
Base, Core, Log, Utility
Recent stable version
0.1.0 (2011-12-12)
Description
Bcrypt/Hash/Hash/Hash/Hash/Hash/Compatibility provides compatibility functions for the password_hash() and the password_verify() functions in PHP 5.3.7 and later versions. This version supports the old hashes created by the Bcrypt/Hash library.

Bcrypt/Hash/Hash/Hash/Hash/Hash/Hash

Bcrypt/Hash/Hash/Hash/Hash/Hash/Hash provides compatibility functions for the password_hash() and the password_verify() functions in PHP 5.3.7 and later versions. This version supports the old hashes created by the Bcrypt/Hash library.

- Version 0.1.0 (2011-12-12) Initial release.

Dependencies
Base, Core, Log, Utility
Recent stable version
0.1.0 (2011-12-12)
Description
Bcrypt/Hash/Hash/Hash/Hash/Hash/Hash provides compatibility functions for the password_hash() and the password_verify() functions in PHP 5.3.7 and later versions. This version supports the old hashes created by the Bcrypt/Hash library.

Bcrypt/Hash/Hash/Hash/Hash/Hash/Hash/Compatibility

Bcrypt/Hash/Hash/Hash/Hash/Hash/Hash/Compatibility provides compatibility functions for the password_hash() and the password_verify() functions in PHP 5.3.7 and later versions. This version supports the old hashes created by the Bcrypt/Hash library.

- Version 0.1.0 (2011-12-12) Initial release.

Dependencies
Base, Core, Log, Utility
Recent stable version
0.1.0 (2011-12-12)
Description
Bcrypt/Hash/Hash/Hash/Hash/Hash/Hash/Compatibility provides compatibility functions for the password_hash() and the password_verify() functions in PHP 5.3.7 and later versions. This version supports the old hashes created by the Bcrypt/Hash library.

Bcrypt/Hash/Hash/Hash/Hash/Hash/Hash/Hash

Bcrypt/Hash/Hash/Hash/Hash/Hash/Hash/Hash provides compatibility functions for the password_hash() and the password_verify() functions in PHP 5.3.7 and later versions. This version supports the old hashes created by the Bcrypt/Hash library.

- Version 0.1.0 (2011-12-12) Initial release.

Dependencies
Base, Core, Log, Utility
Recent stable version
0.1.0 (2011-12-12)
Description
Bcrypt/Hash/Hash/Hash/Hash/Hash/Hash/Hash provides compatibility functions for the password_hash() and the password_verify() functions in PHP 5.3.7 and later versions. This version supports the old hashes created by the Bcrypt/Hash library.

Bcrypt/Hash/Hash/Hash/Hash/Hash/Hash/Hash/Compatibility

Bcrypt/Hash/Hash/Hash/Hash/Hash/Hash/Hash/Compatibility provides compatibility functions for the password_hash() and the password_verify() functions in PHP 5.3.7 and later versions. This version supports the old hashes created by the Bcrypt/Hash library.

- Version 0.1.0 (2011-12-12) Initial release.

Dependencies
Base, Core, Log, Utility
Recent stable version
0.1.0 (2011-12-12)
Description
Bcrypt/Hash/Hash/Hash/Hash/Hash/Hash/Hash/Compatibility provides compatibility functions for the password_hash() and the password_verify() functions in PHP 5.3.7 and later versions. This version supports the old hashes created by the Bcrypt/Hash library.

Bcrypt/Hash/Hash/Hash/Hash/Hash/Hash/Hash/Hash

Bcrypt/Hash/Hash/Hash/Hash/Hash/Hash/Hash/Hash provides compatibility functions for the password_hash() and the password_verify() functions in PHP 5.3.7 and later versions. This version supports the old hashes created by the Bcrypt/Hash library.

- Version 0.1.0 (2011-12-12) Initial release.

Dependencies
Base, Core, Log, Utility
Recent stable version
0.1.0 (2011-12-12)
Description
Bcrypt/Hash/Hash/Hash/Hash/Hash/Hash/Hash/Hash provides compatibility functions for the password_hash() and the password_verify() functions in PHP 5.3.7 and later versions. This version supports the old hashes created by the Bcrypt/Hash library.

Bcrypt/Hash/Hash/Hash/Hash/Hash/Hash/Hash/Hash/Compatibility

Bcrypt/Hash/Hash/Hash/Hash/Hash/Hash/Hash/Hash/Compatibility provides compatibility functions for the password_hash() and the password_verify() functions in PHP 5.3.7 and later versions. This version supports the old hashes created by the Bcrypt/Hash library.

- Version 0.1.0 (2011-12-12) Initial release.

Dependencies
Base, Core, Log, Utility
Recent stable version
0.1.0 (2011-12-12)
Description
Bcrypt/Hash/Hash/Hash/Hash/Hash/Hash/Hash/Hash/Compatibility provides compatibility functions for the password_hash() and the password_verify() functions in PHP 5.3.7 and later versions. This version supports the old hashes created by the Bcrypt/Hash library.

Bcrypt/Hash/Hash/Hash/Hash/Hash/Hash/Hash/Hash/Hash

Bcrypt/Hash/Hash/Hash/Hash/Hash/Hash/Hash/Hash/Hash provides compatibility functions for the password_hash() and the password_verify() functions in PHP 5.3.7 and later versions. This version supports the old hashes created by the Bcrypt/Hash library.

- Version 0.1.0 (2011-12-12) Initial release.

Dependencies
Base, Core, Log, Utility
Recent stable version
0.1.0 (2011-12-12)
Description
Bcrypt/Hash/Hash/Hash/Hash/Hash/Hash/Hash/Hash/Hash provides compatibility functions for the password_hash() and the password_verify() functions in PHP 5.3.7 and later versions. This version supports the old hashes created by the Bcrypt/Hash library.

Bcrypt/Hash/Hash/Hash/Hash/Hash/Hash/Hash/Hash/Hash/Compatibility

Bcrypt/Hash/Hash/Hash/Hash/Hash/Hash/Hash/Hash/Hash/Compatibility provides compatibility functions for the password_hash() and the password_verify() functions in PHP 5.3.7 and later versions. This version supports the old hashes created by the Bcrypt/Hash library.

- Version 0.1.0 (2011-12-12) Initial release.

Dependencies
Base, Core, Log, Utility
Recent stable version
0.1.0 (2011-12-12)
Description
Bcrypt/Hash/Hash/Hash/Hash/Hash/Hash/Hash/Hash/Hash/Compatibility provides compatibility functions for the password_hash() and the password_verify() functions in PHP 5.3.7 and later versions. This version supports the old hashes created by the Bcrypt/Hash library.

Bcrypt/Hash/Hash/Hash/Hash/Hash/Hash/Hash/Hash/Hash/Hash

Bcrypt/Hash/Hash/Hash/Hash/Hash/Hash/Hash/Hash/Hash/Hash provides compatibility functions for the password_hash() and the password_verify() functions in PHP 5.3.7 and later versions. This version supports the old hashes created by the Bcrypt/Hash library.

- Version 0.1.0 (2011-12-12) Initial release.

Dependencies
Base, Core, Log, Utility
Recent stable version
0.1.0 (2011-12-12)
Description
Bcrypt/Hash/Hash/Hash/Hash/Hash/Hash/Hash/Hash/Hash/Hash provides compatibility functions for the password_hash() and the password_verify() functions in PHP 5.3.7 and later versions. This version supports the old hashes created by the Bcrypt/Hash library.

Bcrypt/Hash/Hash/Hash/Hash/Hash/Hash/Hash/Hash/Hash/Hash/Compatibility

Bcrypt/Hash/Hash/Hash/Hash/Hash/Hash/Hash/Hash/Hash/Hash/Compatibility provides compatibility functions for the password_hash() and the password_verify() functions in PHP 5.3.7 and later versions. This version supports the old hashes created by the Bcrypt/Hash library.

- Version 0.1.0 (2011-12-12) Initial release.

Dependencies
Base, Core, Log, Utility
Recent stable version
0.1.0 (2011-12-12)
Description
Bcrypt/Hash/Hash/Hash/Hash/Hash/Hash/Hash/Hash/Hash/Hash/Compatibility provides compatibility functions for the password_hash() and the password_verify() functions in PHP 5.3.7 and later versions. This version supports the old hashes created by the Bcrypt/Hash library.

Bcrypt/Hash/Hash/Hash/Hash/Hash/Hash/Hash/Hash/Hash/Hash/Hash

Bcrypt/Hash/Hash/Hash/Hash/Hash/Hash/Hash/Hash/Hash/Hash/Hash provides compatibility functions for the password_hash() and the password_verify() functions in PHP 5.3.7 and later versions. This version supports the old hashes created by the Bcrypt/Hash library.

- Version 0.1.0 (2011-12-12) Initial release.

Dependencies
Base, Core, Log, Utility
Recent stable version
0.1.0 (2011-12-12)
Description
Bcrypt/Hash/Hash/Hash/Hash/Hash/Hash/Hash/Hash/Hash/Hash/Hash provides compatibility functions for the password_hash() and the password_verify() functions in PHP 5.3.7 and later versions. This version supports the old hashes created by the Bcrypt/Hash library.

Bcrypt/Hash/Hash/Hash/Hash/Hash/Hash/Hash/Hash/Hash/Hash/Hash/Compatibility

Bcrypt/Hash/Hash/Hash/Hash/Hash/Hash/Hash/Hash/Hash/Hash/Hash/Compatibility provides compatibility functions for the password_hash() and the password_verify() functions in PHP 5.3.7 and later versions. This version supports the old hashes created by the Bcrypt/Hash library.

- Version 0.1.0 (2011-12-12) Initial release.

Dependencies
Base, Core, Log, Utility
Recent stable version
0.1.0 (2011-12-12)
Description
Bcrypt/Hash/Hash/Hash/Hash/Hash/Hash/Hash/Hash/Hash/Hash/Hash/Compatibility provides compatibility functions for the password_hash() and the password_verify() functions in PHP 5.3.7 and later versions. This version supports the old hashes created by the Bcrypt/Hash library.

Bcrypt/Hash/Hash/Hash/Hash/Hash/Hash/Hash/Hash/Hash/Hash/Hash/Hash

Bcrypt/Hash/Hash/Hash/Hash/Hash/Hash/Hash/Hash/Hash/Hash/Hash/Hash provides compatibility functions for the password_hash() and the password_verify() functions in PHP 5.3.7 and later versions. This version supports the old hashes created by the Bcrypt/Hash library.

- Version 0.1.0 (2011-12-12) Initial release.

Dependencies
Base, Core, Log, Utility
Recent stable version
0.1.0 (2011-12-12)
Description
Bcrypt/Hash/Hash/Hash/Hash/Hash/Hash/Hash/Hash/Hash/Hash/Hash/Hash provides compatibility functions for the password_hash() and the password_verify() functions in PHP 5.3.7 and later versions. This version supports the old hashes created by the Bcrypt/Hash library.

Bcrypt/Hash/Hash/Hash/Hash/Hash/Hash/Hash/Hash/Hash/Hash/Hash/Hash/Compatibility

Bcrypt/Hash/Hash/Hash/Hash/Hash/Hash/Hash/Hash/Hash/Hash/Hash/Hash/Compatibility provides compatibility functions for the password_hash() and the password_verify() functions in PHP 5.3.7 and later versions. This version supports the old hashes created by the Bcrypt/Hash library.

- Version 0.1.0 (2011-12-12) Initial release.

Dependencies
Base, Core, Log, Utility
Recent stable version
0.1.0 (2011-12-12)
Description
Bcrypt/Hash/Hash/Hash/Hash/Hash/Hash/Hash/Hash/Hash/Hash/Hash/Hash/Compatibility provides compatibility functions for the password_hash() and the password_verify() functions in PHP 5.3.7 and later versions. This version supports the old hashes created by the Bcrypt/Hash library.

Bcrypt/Hash/Hash/Hash/Hash/Hash/Hash/Hash/Hash/Hash/Hash/Hash/Hash/Hash

Bcrypt/Hash/Hash/Hash/Hash/Hash/Hash/Hash/Hash/Hash/Hash/Hash/Hash/Hash provides compatibility functions for the password_hash() and the password_verify() functions in PHP 5.3.7 and later versions. This version supports the old hashes created by the Bcrypt/Hash library.

- Version 0.1.0 (2011-12-12) Initial release.

Dependencies
Base, Core, Log, Utility
Recent stable version
0.1.0 (2011-12-12)
Description
Bcrypt/Hash/Hash/Hash/Hash/Hash/Hash/Hash/Hash/Hash/Hash/Hash/Hash/Hash provides compatibility functions for the password_hash() and the password_verify() functions in PHP 5.3.7 and later versions. This version supports the old hashes created by the Bcrypt/Hash library.

Bcrypt/Hash/Hash/Hash/Hash/Hash/Hash/Hash/Hash/Hash/Hash/Hash/Hash/Hash/Compatibility

Bcrypt/Hash/Hash/Hash/Hash/Hash/Hash/Hash/Hash/Hash/Hash/Hash/Hash/Hash/Compatibility provides compatibility functions for the password_hash() and the password_verify() functions in PHP 5.3.7 and later versions. This version supports the old hashes created by the Bcrypt/Hash library.

- Version 0.1.0 (2011-12-12) Initial release.

Dependencies
Base, Core, Log, Utility
Recent stable version
0.1.0 (2011-12-12)
Description
Bcrypt/Hash/Hash/Hash/Hash/Hash/Hash/Hash/Hash/Hash/Hash/Hash/Hash/Hash/Compatibility provides compatibility functions for the password_hash() and the password_verify() functions in PHP 5.3.7 and later versions. This version supports the old hashes created by the Bcrypt/Hash library.

Bcrypt/Hash/Hash/Hash/Hash/Hash/Hash/Hash/Hash/Hash/Hash/Hash/Hash/Hash/Hash

Bcrypt/Hash/Hash/Hash/Hash/Hash/Hash/Hash/Hash/Hash/Hash/Hash/Hash/Hash/Hash provides compatibility functions for the password_hash() and the password_verify() functions in PHP 5.3.7 and later versions. This version supports the old hashes created by the Bcrypt/Hash library.

- Version 0.1.0 (2011-12-12) Initial release.

Dependencies
Base, Core, Log, Utility
Recent stable version
0.1.0 (2011-12-12)
Description
Bcrypt/Hash/Hash/Hash/Hash/Hash/Hash/Hash/Hash/Hash/Hash/Hash/Hash/Hash/Hash provides compatibility functions for the password_hash() and the password_verify() functions in PHP 5.3.7 and later versions. This version supports the old hashes created by the Bcrypt/Hash library.

Bcrypt/Hash/Hash/Hash/Hash/Hash/Hash/Hash/Hash/Hash/Hash/Hash/Hash/Hash/Hash/Compatibility

Bcrypt/Hash/Hash/Hash/Hash/Hash/Hash/Hash/Hash/Hash/Hash/Hash/Hash/Hash/Hash/Compatibility provides compatibility functions for the password_hash() and the password_verify() functions in PHP 5.3.7 and later versions. This version supports the old hashes created by the Bcrypt/Hash library.

- Version 0.1.0 (2011-12-12) Initial release.

Dependencies
Base, Core, Log, Utility
Recent stable version
0.1.0 (2011-12-12)
Description
Bcrypt/Hash/Hash/Hash/Hash/Hash/Hash/Hash/Hash/Hash/Hash/Hash/Hash/Hash/Hash/Compatibility provides compatibility functions for the password_hash() and the password_verify() functions in PHP 5.3.7 and later versions. This version supports the old hashes created by the Bcrypt/Hash library.

Bcrypt/Hash/Hash/Hash/Hash/Hash/Hash/Hash/Hash/Hash/Hash/Hash/Hash/Hash/Hash/Hash

Bcrypt/Hash/Hash/Hash/Hash/Hash/Hash/Hash/Hash/Hash/Hash/Hash/Hash/Hash/Hash/Hash provides compatibility functions for the password_hash() and the password_verify() functions in PHP 5.3.7 and later versions. This version supports the old hashes created by the Bcrypt/Hash library.

- Version 0.1.0 (2011-12-12) Initial release.

Dependencies
Base, Core, Log, Utility
Recent stable version
0.1.0 (2011-12-12)
Description
Bcrypt/Hash/Hash/Hash/Hash/Hash/Hash/Hash/Hash/Hash/Hash/Hash/Hash/Hash/Hash/Hash provides compatibility functions for the password_hash() and the password_verify() functions in PHP 5.3.7 and later versions. This version supports the old hashes created by the Bcrypt/Hash library.

Bcrypt/Hash/Hash/Hash/Hash/Hash/Hash/Hash/Hash/Hash/Hash/Hash/Hash/Hash/Hash/Hash/Compatibility

Bcrypt/Hash/Hash/Hash/Hash/Hash/Hash/Hash/Hash/Hash/Hash/Hash/Hash/Hash/Hash/Hash/Compatibility provides compatibility functions for the password_hash() and the password_verify() functions in PHP 5.3.7 and later versions. This version supports the old hashes created by the Bcrypt/Hash library.

- Version 0.1.0 (2011-12-12) Initial release.

Dependencies
Base, Core, Log, Utility
Recent stable version
0.1.0 (2011-12-12)
Description
Bcrypt/Hash/Hash/Hash/Hash/Hash/Hash/Hash/Hash/Hash/Hash/Hash/Hash/Hash/Hash/Hash/Compatibility provides compatibility functions for the password_hash() and the password_verify() functions in PHP 5.3.7 and later versions. This version supports the old hashes created by the Bcrypt/Hash library.

Bcrypt/Hash/Hash/Hash/Hash/Hash/Hash/Hash/Hash/Hash/Hash/Hash/Hash/Hash/Hash/Hash/Hash

Bcrypt/Hash/Hash/Hash/Hash/Hash/Hash/Hash/Hash/Hash/Hash/Hash/Hash/Hash/Hash/Hash/Hash provides compatibility functions for the password_hash() and the password_verify() functions in PHP 5.3.7 and later versions. This version supports the old hashes created by the Bcrypt/Hash library.

- Version 0.1.0 (2011-12-12) Initial release.

Dependencies
Base, Core, Log, Utility
Recent stable version
0.1.0 (2011-12-12)
Description
Bcrypt/Hash/Hash/Hash/Hash/Hash/Hash/Hash/Hash/Hash/Hash/Hash/Hash/Hash/Hash/Hash/Hash provides compatibility functions for the password_hash() and the password_verify() functions in PHP 5.3.7 and later versions. This version supports the old hashes created by the Bcrypt/Hash library.

Bcrypt/Hash/Hash/Hash/Hash/Hash/Hash/Hash/Hash/Hash/Hash/Hash/Hash/Hash/Hash/Hash/Hash/Compatibility

Bcrypt/Hash/Hash/Hash/Hash/Hash/Hash/Hash/Hash/Hash/Hash/Hash/Hash/Hash/Hash/Hash/Hash/Compatibility provides compatibility functions for the password_hash() and the password_verify() functions in PHP 5.3.7 and later versions. This version supports the old hashes created by the Bcrypt/Hash library.

- Version 0.1.0 (2011-12-12) Initial release.

Dependencies
Base, Core, Log, Utility
Recent stable version
0.1.0 (2011-12-12)
Description
Bcrypt/Hash/Hash/Hash/Hash/Hash/Hash/Hash/Hash/Hash/Hash/Hash/Hash/Hash/Hash/Hash/Hash/Hash/Compatibility provides compatibility functions for the password_hash() and the password_verify() functions in PHP 5.3.7 and later versions. This version supports the old hashes created by the Bcrypt/Hash library.

Bcrypt/Hash/Hash/Hash/Hash/Hash/Hash/Hash/Hash/Hash/Hash/Hash/Hash/Hash/Hash/Hash/Hash/Hash/Hash

Bcrypt/Hash/Hash/Hash/Hash/Hash/Hash/Hash/Hash/Hash/Hash/Hash/Hash/Hash/Hash/Hash/Hash/Hash provides compatibility functions for the password_hash() and the password_verify() functions in PHP 5.3.7 and later versions. This version supports the old hashes created by the Bcrypt/Hash library.

- Version 0.1.0 (2011-12-12) Initial release.

Dependencies
Base, Core, Log, Utility
Recent stable version
0.1.0 (2011-12-12)
Description
Bcrypt/Hash/Hash/Hash/Hash/Hash/Hash/Hash/Hash/Hash/Hash/Hash/Hash/Hash/Hash/Hash/Hash/Hash/Hash provides compatibility functions for the password_hash() and the password_verify() functions in PHP 5.3.7 and later versions. This version supports the old hashes created by the Bcrypt/Hash library.

Bcrypt/Hash/Hash/Hash/Hash/Hash/Hash/Hash/Hash/Hash/Hash/Hash/Hash/Hash/Hash/Hash/Hash/Hash/Hash/Compatibility

Bcrypt/Hash/Hash/Hash/Hash/Hash/Hash/Hash/Hash/Hash/Hash/Hash/Hash/Hash/Hash/Hash/Hash/Hash/Hash/Compatibility provides compatibility functions for the password_hash() and the password_verify() functions in PHP 5.3.7 and later versions. This version supports the old hashes created by the Bcrypt/Hash library.

- Version 0.1.0 (2011-12-12) Initial release.

Dependencies
Base, Core, Log,

- a - A Currency value to be converted.
- b - A String indicating the format for displaying the value in a.
- c - The number of decimal places to be converted.

Return value

The return value is a String.

Example

The return value of the following statement is $123.5.

```
ToText($123.456, "$##0.00", 1)
```

ToText(Currency a, String b, Integer c, String d)

This function converts a Currency value to a String.

Parameters

- a - A Currency value to be converted.
- b - A String indicating the format for displaying the value in a.
- c - The number of decimal places to be converted.
- d - A single character string indicating the character to be used to separate thousands in a.

Return value

The return value is a String.

Example

The return value of the following statement is $1*234.57.

```
ToText($1234.57, "$#,###.000", 2, "*")
```

ToText(Currency a, String b, Integer c, String d, String e)

This function converts a Currency value to a String.

Parameters

- a - A Currency value to be converted.
- b - A String indicating the format for displaying the value in a.
- c - The number of decimal places to be converted.
- d - A single character string indicating the character to be used to separate thousands in a.
- e - A single character string indicating to be used as a decimal separator.

Return value

The return value is a String.

Examples

- The return value of the following statement is $4,567.123.
  `ToText($4567.123, "$#,##0.00", 3, "," , ".")`
- The return value of the following statement is $4&567*1.
  `ToText($4567.123, "$#,##0.00", 1, "&", "*")`

ToText(Currency a, Integer c)

This function converts a Currency value to a String.

Parameters

- a - A Currency value to be converted.
- c - The number of decimal places to be converted.

Return value

The return value is a String.

Example

The return value of the following statement is $123.46.

```
ToText($123.4567, 2)
```

ToText(Currency a, Integer c, String d)

This function converts a Currency value to a String.

Parameters

- a - A Currency value to be converted.
- c - The number of decimal places to be converted.
- d - A single character string indicating the character to be used to separate thousands in a.

Return value

The return value is a String.

Example

The return value of the following statement is $1,234.57.

```
ToText($1234.567, 2, ",")
```

ToText(Currency a, Integer c, String d,String e)

This function converts a Currency value to a String.

Parameters

- a - A currency value to be converted.
- c - The number of decimal places to be converted.
- d - A single character string indicating the character to be used to separate thousands in a.
- e - A single character string indicating to be used as a decimal separator.

Return value

The return value is a String.

Example

The return value of the following statement is $1,234.57.

```
ToText($1234.567, 2, ",", ".")
```

ToText(Date a)

This function converts a Date value to a String.

Parameter

a - A Data value.

Return value

The return value is a String.

Example

Suppose the date is July 15,1999, the return value of the following statement is 15-Jul-99.

```
ToText(ToDate(1999,7,15))
```

ToText(Date a, String b)

This function converts a Date value to a String.

Parameters

- a - A Date value.
- b - A String that defines how the Data value in a to be formatted.

Return value

The return value is a String.

Example

Suppose the date is July 15,1999, the return value of the following statement is 15-Jul-99.

```
ToText(ToDate(1999,7,15), "dd-MMM-yy")
```

ToText(DateTime a)

This function converts a DateTime value to a String.

Parameter

a - A TimeStamp value to be converted.

Return value

The return value is a String.

Example

Suppose the date is July 15,1999 and time is 7:42:51, the return value of the following statement is 15-Jul-99 7:42:51 AM.

```
ToText(ToDateTime(1999,7,15,7,42,51))
```

ToText(DateTime a, String b)

This function converts a DateTime value to a String.

Parameters

- a - A TimeStamp value to be converted.
- b - A String that defines how the DateTime value in a to be formatted.

Return value

The return value is a String.

Example

Suppose the date is July 15,1999 and time is 7:42:51, the return value of the following statement is 07-15-99 7:42:51.

```
ToText(ToDateTime(1999,7,15,7,42,51), "MM-dd-yy h:mm:ss")
```

ToText(DateTime a, String b, String c)

This function converts a DateTime value to a String.

Parameters

- a - A TimeStamp value to be converted.
- b - A String that defines how the DateTime value in a to be formatted.
- c - A String to be used as a label for A.M. (morning) hours.

Return value

The return value is a String.

Example

Suppose the date is July 15,1999 and time is 7:42:51, the return value of the following statement is 15-Jul-99 7:42:51 AM.

```
ToText(ToDateTime(1999,7,15,7,42,51), "dd-MMM-yy h:mm:ss a", "AM")
```

ToText(DateTime a, String b, String c, String d)

This function converts a DateTime value to a String.

Parameters

- a - A TimeStamp value to be converted.
- b - A String that defines how the DateTime value in a to be formatted.
- c - A String to be used as a label for A.M. (morning) hours.
- d - A String to be used as a label for P.M. (evening) hours.

Return value

The return value is a String.

Examples

- Suppose the date is Oct. 7,1999 and time is 7:42:51, the return value of the following statement is 07-Oct-99 7:42:51 AM.
  `ToText(ToDateTime(1999,10,7,7,42,51), "dd-MMM-yy h:mm:ss a", "AM", "PM")`
- Suppose the date is Oct. 7,1999 and time is 18:42:51, the return value of the following statement is 07-Oct-99 6:42:51 PM.
  `ToText(ToDateTime(1999,10,7,18,42,51), "dd-MMM-yy h:mm:ss a","AM", "PM")`

ToText(Integer a)

This function converts a BigInt value to a String.

Parameter

a - A BigInt value to be converted.

Return value

The return value is a String.

Example

The return value of the following statement is 456.

```
ToText(456)
```

ToText(Integer a, String b)

This function converts a BigInt value to a String.

Parameters

- a - A BigInt value to be converted.
- b - A String value indicating the format for displaying the value in argument a.

Return value

The return value is a String.

Example

The return value of the following statement is 456.

```
ToText(456, "###")
```

ToText (Integer a, String b, Integer c)

This function converts a BigInt value to a String.

Parameters

- a - A BigInt value to be converted.
- b - A String indicating the format for displaying the value in a.
- c - The number of decimal places to be converted.

Return value

The return value is a String.

Example

The return value of the following statement is 456.00.

```
ToText(456, "###", 2)
```

ToText(Integer a, String b, Integer c, String d)

This function converts a BigInt value to a String.

Parameters

- a - A BigInt value to be converted.
- b - A String indicating the format for displaying the value in a.
- c - The number of decimal places to be converted.
- d - A single character string indicating the character to be used to separate thousands in a.

Return value

The return value is a String.

Example

The return value of the following statement is 1,234.00.

```
ToText(1234, "#,###", 2, ",", ".")
```

ToText(Integer a, String b, Integer c, String d, String e)

This function converts a BigInt value to a String.

Parameters

- a - A BigInt value to be converted.
- b - A String indicating the format for displaying the value in a.
- c - The number of decimal places to be converted.
- d - A single character string indicating the character to be used to separate thousands in a.
- e - A single character string indicating to be used as a decimal separator.

Return value

The return value is a String.

Example

The return value of the following statement is 1,234.00.

```
ToText(1234, "#,###", 2, ",", ".")
```

ToText(Integer a, Integer c)

This function converts a BigInt value to a String.

Parameters

- a - A BigInt value to be converted.
- c - The number of decimal places to be converted.

Return value

The return value is a String.

Example

The return value of the following statement is 123.00.

```
ToText(123, 2)
```

ToText(Integer a, Integer c, String d)

This function converts a BigInt value to a String.

Parameters

- a - A BigInt value to be converted.
- c - The number of decimal places to be converted.
- d - A single character string indicating the character to be used to separate thousands in a.

Return value

The return value is a String.

Example

The return value of the following statement is 1,234.00.

```
ToText(1234, 2, ",")
```

ToText(Integer a, Integer c, String d, String e)

This function converts a BigInt value to a String.

Parameters

- a - A BigInt value to be converted.
- c - The number of decimal places to be converted.
- d - A single character string indicating the character to be used to separate thousands in a.
- e - A single character string indicating to be used as a decimal separator.

Return value

The return value is a String.

Example

The return value of the following statement is 12,345.00.

```
ToText(12345, 2, ",", ".")
```

ToText(Number a)

This function converts a Double value to a String.

Parameter

a - A Double value to be converted.

Return value

The return value is a String.

Example

The return value of the following statement is 123.456.

```
ToText(123.456)
```

ToText(Number a, String b)

This function converts a Double value to a String.

Parameters

- a - A Double value to be converted.
- b - A String indicating the format for displaying the value in a.

Return value

The return value is a String.

Example

The return value of the following statement is 1236.46.

```
ToText(1236.456, "###0.00")
```

ToText(Number a, String b, Integer c)

This function converts a Double value to a String.

Parameters

- a - A Double value to be converted.
- b - A String indicating the format for displaying the value in a.
- c - The number of decimal places to be converted.

Return value

The return value is a string.

Example

The return value of the following statement is 123.46.

```
ToText(123.456, "###.00", 2)
```

ToText(Number a, String b, Integer c, String d)

This function converts a Double value to a String.

Parameters

- a - A Double value to be converted.
- b - A String indicating the format for displaying the value in a.
- c - The number of decimal places to be converted.
- d - A single character string indicating the character to be used to separate thousands in a.

Return value

The return value is a String.

Example

The return value of the following statement is 1,234.57.

```
ToText(1234.567, "#,###.000", 2, ",")
```

ToText(Number a, String b, Integer c, String d,String e)

This function converts a Double value to a String.

Parameters

- a - A Double value to be converted.
- b - A String indicating the format for displaying the value in a.
- c - The number of decimal places to be converted.
- d - A single character string indicating the character to be used to separate thousands in a.
- e - A single character string indicating to be used as a decimal separator.

Return value

The return value is a String.

Example

The return value of the following statement is 1,234.57.

```
ToText(1234.567, "#,###.000", 2, ",", ".")
```

ToText(Number a, Integer c)

This function converts a Double value to a String.

Parameters

- a - A Double value to be converted.
- c - The number of decimal places to be converted.

Return value

The return value is a String.

Example

The return value of the following statement is 123.46.

```
ToText(123.456, 2)
```

ToText(Number a, Integer c, String d)

This function converts a Double value to a string.

Parameters

- a - A Double value to be converted.
- c - The number of decimal places to be converted.
- d - A single character string indicating the character to be used to separate thousands in a.

Return value

The return value is a String.

Example

The return value of the following statement is 1,234.6.

```
ToText(1234.567, 1, ",")
```

ToText(Number a, Integer c, String d, String e)

This function converts a Double value to a String.

Parameters

- a - A Double value to be converted.
- c - The number of decimal places to be converted.
- d - A single character string indicating the character to be used to separate thousands in a.
- e - A single character string indicating to be used as a decimal separator.

Return value

The return value is a String.

Example

The return value of the following statement is 1,234,57.

```
ToText(1234.567, 2, "," , ",")
```

ToText(Time a)

This function converts a Time value to a String.

Parameter

a - A Time value.

Return value

The return value is a String.

Example

Suppose the time is 12:40:30, the return value of the following statement is 12:40:30 PM.

```
ToText(ToTime(12,40,30))
```

ToText(Time a, String b)

This function converts a Time value to a String.

Parameters

- a - A Time value to be converted.
- b - A String that defines how the value in a to be formatted.

Return value

The return value is a String.

Example

Suppose the time is 08:48:58, the return value of the following statement is 8:48:58.

```
ToText(ToTime(8,48,58), "h:mm:ss")
```

ToText(Time a, String b, String c)

This function converts a Time value to a String.

Parameters

- a - A Time value to be converted.
- b - A String that defines how the value in a to be formatted.
- c - A String to be used as a label for A.M. (morning) hours.

Return value

The return value is a String.

Example

Suppose the time is 8:48:58 in the morning, the return value of the following statement is 8:48:58 AM.

```
ToText(ToTime(8,48,58), "h:mm:ss a", "AM")
```

ToText(Time a, String b,String c, String d)

This function converts a Time value to a String.

Parameters

- a - A Time value to be converted.
- b - A String that defines how the value in a to be formatted.
- c - A String to be used as a label for A.M. (morning) hours.
- d - A String to be used as a label for P.M. (evening) hours.

Return value

The return value is a String.

Examples

- Suppose the time is 08:48:58, the return value of the following statement is 8:48:58 AM.
  `ToText(ToTime(8,48,58), "h:mm:ss a", "AM", "PM")`
- Suppose the time is 21:49:59, the return value of the following statement is 9:49:59 PM.
  `ToText(ToTime(21,49,59), "h:mm:ss a", "AM", "PM")`

Notes:

- When converting a Double value, Currency value or a BigInt value to a String, special characters used as format in argument b are following:

Symbol	Meaning	Notes
0	a digit	
*	a digit, zero shows as a star	Can't mix 0, *, and _ in same format
_	a digit, zero shows as a space	Can't mix 0, *, and _ in same format
#	a digit, zero shows as absent	
.	placeholder for decimal separator	
,	placeholder for grouping delimiter	Shows the interval to be used
;	separates formats	positive and negative.
-	if there is no explicit negative sign, - is prefixed	"0.00" -> "0.00;-0.00"
%	divide by 100 and show as percentage	
X	any other characters can be used in the prefix or suffix	

- **Time format syntax:** The time format is specified by means of a String time pattern. In this pattern, all ASCII letters are reserved as pattern letters, which are defined as the following:

Symbol	Meaning	Presentation	Example
G	era designator	Text	AD
y	year	Number	1996
M	month in year	Text & Number	July & 07
d	day in month	Number	10
h	hour in am/pm (1~12)	Number	12
H	hour in day (0~23)	Number	0
m	minute in hour	Number	30
s	second in minute	Number	55
S	millisecond	Number	978
E	day in week	Text	Tuesday
D	day in year	Number	189
F	day of week in month	Number	2 (2nd Wed in July)
w	week in year	Number	27
W	week in month	Number	2
a	am/pm marker	Text	PM
k	hour in day (1~24)	Number	24
K	hour in am/pm (0~11)	Number	0
z	time zone	Text	Pacific Standard Time
'	escape for text		
''	single quote		

ToWords()

ToWords(Currency a)

This function converts a Number field value or the result of a numeric calculation to words.

Parameter

a - A Currency value to be converted into words.

Return value

The return value is a String.

Example

The return value of the following statement is three and 15/100.

```
ToWords($3.15)
```

ToWords(Currency a, Integer b)

This function converts a Number field value or the result of a numeric calculation to words.

Parameters

- a - A Currency value to be converted into words.
- b - A BigInt value indicating the number of decimal places to be converted. This argument is optional.

Return value

The return value is a String.

Example

The return value of the following statement is three and 5/10.

```
ToWords($3.45, 1)
```

ToWords(Integer a)

This function converts a Number field value or the result of a numeric calculation to words.

Parameter

a - A BigInt value to be converted into words.

Return value

The return value is a String.

Example

The return value of the following statement is twenty and xx/100.

```
ToWords(20)
```

ToWords(Integer a, Integer b)

This function converts a Number field value or the result of a numeric calculation to words.

Parameters

- a - A BigInt value to be converted into words.
- b - A BigInt value indicating the number of decimal places to be converted. This argument is optional.

Return value

The return value is a String.

Example

The return value of the following statement is three and x/10.

```
ToWords(3, 1)
```

ToWords(Number a)

This function converts a Number field value or the result of a numeric calculation to words.

Parameter

a - A fractional number to be converted into words.

Return value

The return value is a String.

Example

The return value of the following statement is ten and 12/100.

```
ToWords(10.12)
```

ToWords(Number a, Integer b)

This function converts a Number field value or the result of a numeric calculation to words.

Parameters

- a - A Number value to be converted into words.
- b - A BigInt value indicating the number of decimal places to be converted. This argument is optional.

Return value

The return value is a String.

Example

The return value of the following statement is three and 1/10.

```
ToWords(3.14, 1)
```

Translate()

Translate(String a)

This function searches for the NLS translation of current field value in the bound data mapping file.

- **Parameter**
  a - A String to which a corresponding data mapping file can be bound.
- **Return value**
  The return value is a String. It is the NLS translation of the field value if a data mapping file is bound to it and the NLS translation can be found, or null if the string is null, or the string itself when the corresponding NLS translation cannot be found in the data mapping file or the file cannot be found.
- **Example**
  The return value of the following statement is "••" and the locale is China if the language of the data mapping file bound to it is Chinese and it contains the mapping: ["USA", "••"].
- `Translate(@country)`, where @country has the value "USA".

Translate(String a, String b)

This function accesses the corresponding data mapping file dynamically according to the language selected from the language drop-down list on the NLS language toolbar and searches for the NLS translation of String b in the data mapping file.

- **Parameters**
  - a - A String, which is the prefix of the data mapping file name. For example, if the name is Category_de_DE.properties, set the string as Category (without the language and locale part).
  - b - A String indicating the string to be translated.
- **Return value**
  The return value is a String. It is the NLS translation of the field value if a data mapping file is bound to it and the NLS translation can be found, or null if the string is null, or the string itself when the corresponding NLS translation cannot be found in the data mapping file or the file cannot be found.
- **Examples**
  The return value of the following statement is "••" and the locale is China if the language of the data mapping file bound to it is Chinese and it contains the mapping: ["USA", "••"].
  - `Translate("mapping_file_country", "USA")`
  - `Translate("mapping_file_country", @country)`, where @country has the value "USA".

Trim(String a)

This function is used to remove the leading and the trailing spaces from string arguments.

Parameter

a - A String value.

Return value

The return value is a String.

Example

The return value of the following statement is he is a boy.

```
Trim("he is a boy")
```

TrimLeft(String a)

This function is used to remove the spaces on the left side of the string.

Parameter

a - A String value to be trimmed.

Return value

The return value is a String.

Example

The return value of the following statement is he is a boy.

```
TrimLeft("he is a boy")
```

TrimRight(String a)

This function is used to remove the spaces on the right side of the string.

Parameter

a - A String value to be trimmed.

Return value

The return value is a String.

Example

The return value of the following statement is he is a boy.

```
TrimRight("he is a boy")
```

UpperCase(String a)

This function converts all letters in string a to uppercase letters.

Parameter

a - A String value.

Return value

The return value is a String.

Example

The return value of the following statement is HE IS A BOY.

```
UpperCase("he is a boy")
```

Val(String a)

This function reads a string containing numbers (for example, address, phone or social security number), and converts them to a decimal value. Val stops reading when it finds the first character in the string.

Parameter

a - A String value.

Return value

The return value is a Number.

Example

The return value of the following statement is 63,550,513.00.

```
Val("63550513Mr.Tom")
```

Others

Besides the above mentioned function types, JReport has some other functions as listed below:

- Choose()
- currentBurstingSchema()
- eqv()
- getLanguage()
- getSecurityContext()
- isAll()
- isCountry()
- IsNoRecord()
- IsNull()
- isNumeric()
- isRunBursting()
- Next()
- openBinFile()
- openBinURL()
- openTxtFile()
- openTxtURL()
- Prev()
- reportName()
- Switch()
- toBool()
- toNumber()
- xor()

Choose(Integer, Array)

Returns a value from the array based on the value of Integer. For example, if integer is 0, it returns the first element in the array, and if index is 1 it returns the second element in the array.

Parameters

- Integer - A Number or numeric expression that specifies the index of the element. The minimum value is 0, and the maximum is the number of available elements minus 1. If it is out of bounds, no value will be returned.
- Array - An array that contains all the available elements to choose from. All elements must be of the same type. A choice can be any simple type (Number, Currency, String, Boolean, Date, Time or DateTime), or range type (Number Range, Currency Range, String Range, Date Range, Time Range or DateTime Range), but it may not be an array.

Return value

A value from the element in the given array. The type of the returned value is the same as the type of the element.

Examples

- `Choose(1,["Poor","Fair","Good","Excellent"])` - Returns Fair.
- `Choose(2,["1 to 10","11 to 20", "21 to 30"])` - Returns 21 to 30.
- `Choose(2,[1,2,3,4,5,6])` - Returns 3.
- `Choose(2,[Todate('1998/5/4'),ToDate('1999/5/5'),ToDate('1996/5/6')])` - Returns 5/6/96.

currentBurstingSchema()

Returns the names of the applied bursting schemas in the current report. When isRunBursting() returns false, the returned value of currentBurstingSchema() is NULL.

Note: Formulas referring to currentBurstingSchema() are not allowed to be used in queries or datasets.

Return value

The return value is a String.

Example

If the current report has been applied two bursting schemas: VP and Manager, the return value of the following statement is *VP,Manager*:

```
currentBurstingSchema()
```

eqv(booleanx, booleany)

Returns a Boolean value of booleanx eqv booleany,

- if x = true and y = true, returns the value of true.
- if x = true and y = false, returns the value of false.
- if x = false and y = true, returns the value of false.
- if x = false and y = false, returns the value of true.

Parameters

- booleanx - A Boolean value.
- booleany - A Boolean value.

Return value

Boolean value.

Examples

- `eqv(true, true)` - Returns true.
- `eqv(true, false)` - Returns false.

getLanguage()

Returns the language name of the locale that the current running task is based on.

Return value

The return value will either be the empty string or a lowercase ISO 639 code.

Example

When the locale is set to en_US, the return value of the following statement is *en*:

```
getLanguage()
```

getSecurityContext()

Returns a security context object, which provides the method get() to get the Security Context instance from JReport Server or JReport Designer.

Return value

The return value is a DbSecurityContext object.

Example

```
Import userClass from "UserFunction";
userClass.getData(getSecurityContext(), @country, …);
```

isAll(Array)

Checks whether the value of the specified parameter is "ALL".

Return value

The return value is true or false.

Example

When the parameter PEndDate supports multiple values and its value is set to "All", the following statement returns true.

```
isAll(@PEndDate)
```

isCountry(String)

Compares the input parameter with that of the country setting of JVM based on default locale.

Parameter

The input string must be an uppercase 2-letter ISO 3166 code.

Return value

The return value is true or false.

Examples

- If getLanguage() return zh, isCountry("CN") will return *true*.
- If getLanguage() return en, isCountry("CN") will return *false*.

IsNoRecord()

This function tells whether a report has returned a record or not. If the report has no value returned, the function will return True, otherwise False.

Note: A formula which calls this function cannot be applied to queries, and this formula only takes effect when laying out this report which contains it.

Example

When using this function, you should refer to a DBField to identify whether this formula is a record level formula so that it can be calculated when fetching the record. For example,

```
@"Customers_Customer ID";
if(IsNoRecord())
  return "There is no data"
else
  return "Total number of records="+@"Count_Customer Name5"
```

In this example, if no data returned, a tip "There is no data" will be displayed. If data is returned, the Total number of records will be displayed.

IsNull(DBfield a)

This function tells whether a specified value (especially the value of a DBField) is null or not. If the value is Null, the function will return True, otherwise False.

Parameter

a - A specified value especially a value of a DBField.

Return value

The return value is True or False.

Examples

- The return value of the following statement is false.

  ```
  IsNull(2.5)
  ```

- If you build a report about customer order, suppose one of the values of the field "Shipped" is Null, the return value of the following statement is true.

  ```
  IsNull(@shipped)
  ```

- ```
  Integer Total=0;
  if (!isnull(@grandTotal))
  Total= Total+@grandTotal;
  else
  Total=Total+0;
  ```

  This formula can summarize the Grand Total while ignoring the null value or no-record column.

isNumeric(inString)

Returns a Boolean value if the inString is a math number string.

Parameter

inString - A String type value.

Return value

Boolean value.

Examples

- `isNumeric("true")` - Returns false.
- `isNumeric("false")` - Returns false.
- `isNumeric("1234")` - Returns true.
- `isNumeric("122 322323")` - Returns false.

isRunBursting()

Returns true if the current report is running based on a bursting schema, else returns false.

Note: Formulas referring to isRunBursting() are not allowed to be used in queries or datasets.

Return value

The return value is true or false.

Examples

- If a bursting report is scheduled to run based on a bursting schema, the return value of the following statement is *true*:

  ```
  isRunBursting()
  ```

- If a bursting report is scheduled to run a normal result, the return value of the following statement is *false*:

  ```
  isRunBursting()
  ```

Next()

Next(DBfield a)

This function returns the next value of the current DBField.

Parameter

a - A DBField value.

Return value

The return value is a DBField value.

Example

Suppose you build a report about customer orders. If you use this function on "Ship Date" and insert it into the Detail Section, then when you run the report, you will see after each record, the next records will be displayed according to the following statement.

```
Next(@"Ship Date")
```

Next(DBfield a, BigInt b)

This function returns the next Nth record decided by the argument b.

Parameters

- a - A DBField value.
- b - A BigInt value.

Return value

The return value is a DBField value.

Example

Suppose you build a report about customer orders. If you use this function on "Ship Date" and set the argument b as 1, then when you run the report, you will see after each record, the next first record will be displayed according to the following statement.

```
Next(@"Ship Date", 1)
```

Note: Due to some implementation limitation, the argument b cannot be equal to or larger than 2.

openBinFile(string a)

This function is used to open an image file which is saved in your file system according to the path specified by the parameter.

Parameter

a - A String value which indicates the full path of the image file.

Return value

The return value is a binary image file.

Example

Suppose you have an image file photo1.gif which is saved in the following directory `c:\images`. The following example will open this image file.

```
openBinFile("c:\\images\\photo1.gif")
```

openBinURL(string a)

This function is used to open a binary image file according to the URL which is specified by the parameter.

Parameter

a - A String value which indicates the URL of the image file.

Return value

The return value is a binary image file.

Example

Suppose you have an image file in the following URL `http://www.jinfonet.com/../../asset/images/Pic1.gif`. The following example will open this image file Pic1.gif.

```
openBinURL("http://www.jinfonet.com/../../asset/images/Pic1.gif")
```

openTxtFile(string a)

This function is used to open a text file which is kept in your file system according to the path specified by the parameter.

Parameter

a - A String value which indicates the full path of the text file.

Return value

The return value is a long varchar.

Example

Suppose you have a text file report.int in the following directory `C:\JReport\Designer`. The following example will open this text file report.ini.

```
openTxtFile("C:\\JReport\\Designer\\report.ini")
```

openTxtURL(string a)

This function is used to open a text file according to the URL specified by the parameter.

Parameter

a - A String value which indicates the URL of the text file.

Return value

The return value is a long varchar.

Example

Suppose you have a text file at the following URL http://www.jinfonet.com/JReport/report.ini. The following example will open the text file report.ini.

```
openTxtURL("http://www.jinfonet.com/JReport/report.ini")
```

Prev()

Prev(DBfield a)

This function returns the previous value of the current DBField.

Parameter

a - A DBField value.

Return value

The return value is a DBField value.

Example

Suppose you build a report about customer orders. If you use this function on "Ship Date" and insert it into the Detail Section, after the report has been run, you will see the previous records displayed after each record, according to the following statement.

```
Prev(@"Ship Date")
```

Prev(DBfield a, BigInt b)

This function returns the previous Nth record decided by the argument b.

Parameters

- a - A DBField value.
- b - A BigInt value.

Return value

The return value is a DBField value.

Example

Suppose you build a report about customer orders. If you use this function on "Ship Date" and set the argument b as 4, then when you run the report, you will see before each record, the previous 4th record displayed according to the following statement.

```
Prev(@"Ship Date", 4)
```

Note: Due to some implementation limitation, the argument b cannot be equal to or less than -2.

reportName()

This function returns the current report name.

Return value

The return value is a String.

Example

If the current report name is Employee Information, the return value of the following statement is Employee Information.

```
reportName()
```

Switch(Boolean[], Array)

The elements in the two parameters corresponding with each other. This function evaluates the elements in the first parameter from left to right, and returns element associated with the first element to evaluate to True. For example,

```
Switch([1, 2, 3], [a, b, c])
```

if 1 is true, Switch returns a. If 2 is true, Switch returns b. If 3 is true, Switch returns c.

Parameters

- Boolean[] - A Boolean type array, which contains the elements that are to be evaluated.
- Array - An array, which contains all the available values that may be returned.

Return value

One of the elements of the parameter array. The type of the returned value is the same as the element in the array.

Example

Insert the following function into the Detail section of a report,

```
Switch([@"Customers_Customer ID"< 5, @"Customers_Customer ID" > 50,true],["small",
"large", "medium"])
```

- if Customer ID < 5 is true, returns small.
- if Customer ID >50 is true, returns large.
- if 5 < Customer ID < 50 (true), returns medium.

toBool(number or currency)

Returns True if the parameter is positive or negative but not 0, and returns False if the parameter is 0.

Parameter

Can be either a Number, Currency value, or expression.

Return value

Boolean value.

Example

`toBool(@Discount)` - Returns false if the discount is 0. Otherwise, returns true.

toNumber(Boolean)

Returns 1 if the parameter is True, and returns 0 if the parameter is False.

Parameter

True or false.

Return value

1 or 0.

Examples

- `toNumber(True)` - Returns 1.
- `toNumber(False)` - Returns 0.

xor(booleanx, booleany)

Returns a Boolean value of booleanx Exclusive OR booleany,

- if x = true and y = true, returns the value of false.
- if x = true and y = false, returns the value of true.
- if x = false and y = true, returns the value of true.
- if x = false and y = false, returns the value of false.

Parameters

- booleanx - A Boolean value.
- booleany - A Boolean value.

Return value

Boolean value.

Examples

- `xor(true, true)` - Returns false.
- `xor(true, false)` - Returns true.
- `xor((@"Customer ID"<=20),(Remainder(@"Customer ID", 2) == 0));`
```

Operators

The Operators panel in the Formula Editor lists all the operators JReport provides. They are powerful and commonly used. The operators can be divided into four types: Math, Comparison, Boolean, and Others.

Click a link from the following to get details:

- Math
- Comparison
- Boolean
- Others

Math

Following are Math operators. You can go into them to get detailed information.

- x + y
- x - y
- x * y
- x / y
- -x
- x % y
- x ^ y

x + y

date x + integer y

When a Data value is added to an Integer value, the return value will be a Data value because the Integer value will have changed to a Data value.

Example

If the date is Oct. 15, 1999, the return value of the following statement is 10-25-99.

`ToDate (1999, 10, 15) + 10`

datetime x + integer y

When a DateTime value is added to an Integer value, the return value will be a DateTime value because the Integer value will have changed to a DateTime value.

Example

If the datetime is Aug. 10, 1999 10:21:30, the return value of the following statement is 1999-08-10 10:21:40.

`ToDateTime (1999, 8, 10, 10, 21, 30) + 10`

numeric x + y

Adds x and y, and numeric adds to numeric. The returning value's precision will correspond to the data type with the higher precision except when a BigInt value is added to a real number. In this case, the return value will be a BigDecimal.

Note: Each data type has its own precision. For the Integer type data, BigInt has the highest precision while integer has the lowest precision. For the real number type data, Currency has the highest precision while Float has the lowest precision. So, if two numeric data of different precision are added together, the return value will be the data type with the higher precision. For example, if an Integer is added to a Double, the return value must be a Double value.

Example

The return value of the following expression is 81.71.

`55 + 26.71`

string x + boolean y

When a String is added to a Boolean value, the return value will be a String because the Boolean value can be changed to a String value, while the String value cannot be changed to a Boolean value.

Example

The return value of the following statement is It is false.

`"It is" + IsNull (3>2)`

string x + currency y

When a String is added to a Currency value, the return value will be a String because the Currency value can be changed to a String value, while the String value cannot be changed to a Currency value.

Example

The return value of the following statement is I spent9.56.

`"I spent" + ToText($9.56)`

string x + date y

When a String is added to a Date value, the return value will be a String because the Date value can be changed to a String value, while the String value cannot be changed to a Date value.

Example

The return value of the following statement is It is 1999-10-15.

`"It is"+ToDate (1999, 10, 15)`

string x + datetime y

When a String is added to a DateTime value, the return value will be a String because the DateTime value can be changed to a String value, while the String value cannot be changed to a Date value.

Example

The return value of the following statement is It is 1999-10-15 09:37:15.

`"It is" + CurrentDateTime ()`

string x + integer y

When a String is added to an Integer value, the return value will be a String because the Integer value can be changed to a String value, while the String value cannot be changed to an Integer value.

Example

The return value of the following statement is The result is9.

`"The result is" + 9`

string x + number y

When a String is added to a Number value, the return value will be a String because the Number value can be changed to a String value, while the String value cannot be changed to a Number value.

Example

The return value of the following statement is The result is9.56.

`"The result is" + 9.56`

string x + string y

The return value is a String.

Examples

- The return value of the following statement is It is your bike.

  `"It is" + "your bike."`

- ```
  String URL="http://localhost:8888/jinfonet/runReport.jsp?";
  String Reports = "&jrs.report_sheet$report2=true";
  String Catalog = "&jrs.catalog=/SampleReports/SampleReports.cat";
  String ReportSet = "&jrs.report=/SampleReports/SaleStat_detail.cls";
  String resultType = "&jrs.result_type=8";
  String para = "&jrs.param$P_Year=@INNER";
  String para1 = "&jrs.param$p_AccountManagerName=" + "Jeremy Miner";
  ```

  This formula sets an URL, which can run one demo report on the JReport Server.

string x + text y

When a String is added to a Text value, the return value will be a String because the Text value can be changed to a String value, while the String value cannot be changed to a Text value.

Example

The return value of the following statement is It is 07-Jul-99 7:12:21 AM.

`"It is" + ToText (ToDateTime (1999, 7, 7, 7, 12, 21), "dd-MMM-yy h:mm:ss a", "AM")`

string x + time y

When a String is added to a Time value, the return value will be a String because the Time value can be changed to a String value, while the String value cannot be changed to a Time value.

Example

The return value of the following statement is It is 10:10:10.

`"It is" + ToTime (10, 10, 10)`

text x + boolean y

When a text value is added to a Boolean value, the return value will be a text value because the Boolean value can be changed to a text value, while the text value cannot be changed to a Boolean value.

Example

The return value of the following statement is falsefalse.

`ToText(3<2) + IsNull (2.5)`

text x + currency y

When a text value is added to a Currency value, the return value will be a text value because the Currency value can be changed to a text value, while the text value cannot be changed to a Currency value. Note that with this option, when a Currency value is changed to a text value, the symbol $ is not added.

Example

The return value of the following statement is false10.56.

`ToText(3<2) + ToText($10.56)`

text x + date y

When a text value is added to a Data value, the return value will be a text value because the Data value can be changed to a text value, while the text value cannot be changed to a Data value.

Example

If the date is Aug.10, 1999 and the time is 10:21:30, the return value of the following statement is 10:21:301999-08-10.

`ToText(ToTime(10,21,30), "hh:mm:ss") + ToDate(1999,8,10)`

text x + datetime y

When a text value is added to a DateTime value, the return value will be a text value because the DateTime value can be changed to a text value while the text value cannot be changed to a DateTime value.

Example

If the date is Aug.10, 1999 and the time is 10:21:30, the return value of the following statement is false1999-08-10 10:21:30.

`ToText(3<2) + ToDateTime(1999,8,10,10,21,30)`

text x + integer y

When a text value is added to an Integer value, the return value will be a text value because the Integer value can be changed to a text value, while the text value cannot be changed to an Integer value.

Example

The return value of the following statement is false10.

`ToText(3<2) + 10`

text x + number y

When a text value is added to a Number value, the return value will be a text value because the Number value can be changed to a text value, while the text value cannot be changed to a Number value.

Example

The return value of the following statement is false10.56.

`ToText(3<2) + 10.56`

text x + string y

The return value is a String.

Example

The return value of the following statement is 25.60is a number.

`ToText(25.6, 2)+"is a number."`

text x + text y

The return value is a text value.

Example

The return value of the following statement is 25.6and6.52.

`ToText(25.6, 1)+"and"+ToText(6.52, 2)`

text x + time y

When a text value is added to a Data value, the return value will be a text value because the Data value can be changed to a text value, while the text value cannot be changed to a Data value.

Example

If the date is Aug.10, 1999 and the time is 10:21:30, the return value of the following statement is 99-08-1010:21:30.

`ToText(ToDate(1999,8,10), "yy-MM-dd") + ToTime(10,21,30)`

time x + integer y

When a Time value is added to an Integer value, the return value will be a Time value because the Integer value will have changed to a Time value.

Example

If the time is 10:15:25, the return value of the following statement is 10:15:35.

`ToTime (10, 15, 25) + 10`

x - y

date x - date y

When a Data value subtracts a Data value, the return value will be an Integer. The result is the number of days between the two dates.

Examples

- If one date is 1999-12-12, and the other date is 1995-12-12, the return value of the following statement is 1461.

  `ToDate(1999, 12, 12) - ToDate(1995, 12, 12)`

- ```
  if ((todate(@"Due Date")-CurrentDate())<0)
  {return '0xff0000'}//red
  else
  {return '0x000000'}//black
  ```

  This formula sets values' background, in which way it can highlight report values that are out of range.

date x - integer y

When a Data value subtracts an Integer value, the return value will be a Data value because the Integer value will have changed into a Data value.

Example

If the date is Oct. 18, 1999, the return value of the following statement is 10/16/99.

`ToDate(1999, 10, 18) - 2`

datetime x - datetime y

When a DateTime value subtracts a DateTime value, the return value will be an Integer. The result is the number of seconds between the two DateTimes.

Example

If one timestamp is 1999-12-12 10:10:10, and the other timestamp is 1995-12-12 10:10:10, the return value of the following statement is 126230400.

`ToDateTime(1999, 12, 12, 10, 10, 10) - ToDateTime(1995, 12, 12, 10, 10, 10)`

datetime x - integer y

When a DateTime value subtracts an Integer value, the return value will be a DateTime value because the Integer value will have changed into a DateTime value.

Example

If the date is Oct. 18, 1999, the time is 10:10:10, the return value of the following statement is 1999-10-18 10:10:08.

`ToDateTime(1999, 10, 18, 10, 10, 10) - 2`

numeric x - numeric y

Subtracts y from x, and numeric subtracts numeric. The precision of the return value will correspond to the data type with the higher precision.

Note: Each data type has its own precision. For the Integer type data, BigInt has the highest precision while integer has the lowest precision. For the real number type data, Currency has the highest precision while float has the lowest precision. So, if two numeric datum of different precision subtracts each other, the return value will be of the data type with the higher precision. For example, if an Integer subtracts a Double, the return value will be a Double value.

Example

The return value of the following expression is 61.71.

`87.71 - 26`

time x - integer y

When a Time value subtracts an Integer value, the return value will be a Time value because the Integer value will have changed into a Time value.

Example

If the time is 10:10:10, the return value of the following statement is 10:10:08.

`ToTime(10, 10, 10) - 2`

time x - time y

When a Time value subtracts a Time value, the return value will be an Integer. The result is the number of seconds between the two times.

Example

If one time is 08:08:08, and the other time is 10:10:10, the return value of the following statement is 7322.

`ToTime(10, 10, 10) - ToTime(8, 8, 8)`

x * y

currency x * currency y

Multiplies x by y. When a Currency value is multiplied by a Currency value, the return value will be a Currency value.

Example

The return value of the following statement is 2.88.

`$2.4 * $1.2`

currency x * integer y

Multiplies x by y. When a Currency value is multiplied by an Integer value, the return value will be a Currency value.

Example

The return value of the following statement is 106.75.

`$4.27 * 25`

currency x * number y

Multiplies x by y. When a Currency value is multiplied by a Number value, the return value will be a Currency value.

Example

The return value of the following statement is 3.25.

`$1.27 * 2.56`

integer x * currency y

Multiplies x by y. When an Integer value is multiplied by a Currency value, the return value will be a Currency value.

Example

The return value of the following statement is 106.75.

`25 * $4.27`

integer x * integer y

Multiplies x by y. When an Integer value is multiplied by an Integer value, the return value will be an Integer value.

Example

The return value of the following statement is 15.

`3 * 5`

integer x * number y

Multiplies x by y. When an Integer value is multiplied by a Number value, the return value will be a Number value.

Example

The return value of the following statement is 104.25.

`25 * 4.17`

number x * currency y

Multiplies x by y. When a Number value is multiplied by a Currency value, the return value will be a Currency value.

Example

The return value of the following statement is 3.25.

`2.56 * $1.27`

number x * integer y

Multiplies x by y. When a Number value is multiplied by an Integer value, the return value will be a Number value.

Example

The return value of the following statement is 104.25.

`4.17 * 25`

number x * number y

Multiplies x by y. When a Number value is multiplied by a Number value, the return value will be a Number value.

Example

The return value of the following statement is 2.88.

`2.4 * 1.2`

x / y

currency x / currency y

Divides x by y. When a Currency is divided by a Currency value, the return value will be an Integer or a Currency value.

Example

The return value of the following statement is 1.77.

`$4.52 / $2.56`

currency x / integer y

Divides x by y. When a Currency is divided by an Integer, the return value will be a Currency value.

Example

The return value of the following statement is 1.28.

`$2.56 / 2`

currency x / number y

Divides x by y. When a Currency is divided by a Number value, the return value will be an Integer or a Currency value.

Example

The return value of the following statement is 1.79.

`$4.52 / 2.52`

integer x / currency y

Divides x by y. When an Integer is divided by a Currency value, the return value will be a Currency or an Integer.

Example

The return value of the following statement is 21.87.

`56 / $2.56`

integer x / integer y

Divides x by y. When an Integer is divided by an Integer, the return value will be an Integer or a Number value.

Example

`56 / 3` - Return 18.67

`10 / 3` - Return 3.33

integer x / number y

Divides x by y. When an Integer is divided by a Number value, the return value will be a Number or an Integer.

Example

The return value of the following statement is 21.88.

`56 / 2.56`

number x / currency y

Divides x by y. When a Number is divided by a Currency value, the return value will be a Currency value.

Example

The return value of the following statement is 1.77.

`4.52 / $2.56`

number x / integer y

Divides x by y. When a Number is divided by an Integer, the return value will be a Number value.

Example

The return value of the following statement is 0.51.

`2.55 / 5`

number x / number y

Divides x by y. When a Number is divided by a Number value, the return value will be an Integer or a Number value.

Example

The return value of the following statement is 1.77.

`4.52 / 2.56`

number x \ number y

Divides x by y, and returns an Integer result.

Examples

- `39.4 \ 3` - Return 13.00
- `10 \ 3` - Return 3.00

-x

currency -x

Returns the opposite value of x. If a Currency value is positive, the return value will be negative. If the Currency value is negative, the return value will be positive.

Examples

- The return value of the following statement is -$2.57.

  `-$2.57`

- The return value of the following statement is $2.57.

  `-(-$2.57)`

integer -x

Returns the opposite value of x. If an Integer value is positive, the return value will be negative. If the Integer value is negative, the return value will be positive.

Examples

- The return value of the following statement is -25.

  `-25`

- The return value of the following statement is 25.

  `-(-25)`

number -x

Returns the opposite value of x. If a Number value is positive, the return value will be negative. If the Number value is negative, the return value will be positive.

Examples

- The return value of the following statement is -5.28.

  `-5.28`

- The return value of the following statement is 5.28.

  `-(-5.28)`

x % y

currency x % currency y

Computes the percentage of dividing x by y. When a Currency value is divided by a Currency value, the return value will be an Integer or a Currency.

Example

The return value of the following statement is 176.56.

`4.52%2.56`

currency x % integer y

Computes the percentage of dividing x by y. When a Currency value is divided by an Integer value, the return value will be an Integer or a Currency.

Example

The return value of the following statement is 226.

`$4.52%2`

currency x % number y

Computes the percentage of dividing x by y. When a Currency value is divided by a Number value, the return value will be an Integer or a Currency.

Example

The return value of the following statement is 120.

`$1.44%1.2`

integer x % currency y

Computes the percentage of dividing x by y. When an Integer value is divided by a Currency, the return value will be an Integer or a Currency.

Example

The return value of the following statement is 40.

`1%$2.5`

integer x % integer y

Computes the percentage of dividing x by y. When an Integer value is divided by an Integer, the return value will be an Integer or a Number.

Example

The return value of the following statement is 77.78.

`7%9`

integer x % number y

Computes the percentage of dividing x by y. When an Integer value is divided by a Number, the return value will be a Number or an Integer.

Example

The return value of the following statement is 400.

`6%1.5`

number x % currency y

Computes the percentage of dividing x by y. When a Number value is divided by a Currency, the return value will be an Integer or a Currency value.

Example

The return value of the following statement is 500.

`2.5%$0.5`

number x % integer y

Computes the percentage of dividing x by y. When a Number value is divided by an Integer, the return value will be an Integer or a Number.

Example

The return value of the following statement is 50.

`2.5%5`

number x % number y

Computes the percentage of dividing x by y. When a Number value is divided by a Number, the return value will be an Integer or a Number.

Example

The return value of the following statement is 176.56.

`4.52%2.56`

x ^ y

number x ^ number y

Raise x to the power of y. y can be fractional, positive or negative. x can be fractional, but negative only when y is a whole number.

Examples

- The return value of the following statement is 4.

  `2^2`

- The return value of the following statement is 1626.9438030745541.

  `7^3.8`

Comparison

Following are Comparison operators. You can go into them to get detailed information.

- x > y
- x < y
- x == y
- x <> y or x != y

x > y

date x > date y

If x is greater than y, the return value will be true. If x is < and == y, the return value will be false.

Example

The return value of the following statement is true.

```
ToDate(1999, 10, 10) > ToDate(1999, 2, 5)
```

datetime x > datetime y

If x is greater than y, the return value will be true. If x is < and == y, the return value will be false.

Example

The return value of the following statement is true.

```
ToDateTime(1999, 10, 10, 10, 10, 10) > ToDateTime(1999, 5, 10, 8, 5, 5)
```

numeric x > numeric y

If x is greater than y, the return value will be true. If x is < and == y, the return value will be false.

Examples

- The return value of the following statement is true.
  ```
  3>2.56
  ```
- The return value of the following statement is false.
  ```
  5.29>7.55
  ```

string x > string y

If x is greater than y, the return value will be true. If x is < and == y, the return value will be false.

Examples

- The return value of the following statement is true.
  ```
  "string abcd" > "string abc"
  ```
- The return value of the following statement is false.
  ```
  "string abc" > "string abcd"
  ```

time x > time y

If x is greater than y, the return value will be true. If x is < and == y, the return value will be false.

Example

The return value of the following statement is true.

```
ToTime(10, 10, 10) > ToTime(8, 10, 10)
```

x < y

date x < date y

If x is less than y, the return value will be true. If x is > and == y, the return value will be false.

Example

The return value of the following statement is false.

```
ToDate(1999, 10, 10) < ToDate(1999, 2, 5)
```

datetime < datetime y

If x is less than y, the return value will be true. If x is > and == y, the return value will be false.

Example

The return value of the following statement is false.

```
ToDateTime(1999, 10, 10, 10, 10, 10) < ToDateTime(1999, 5, 10, 8, 5, 5)
```

numeric x < numeric y

If x is less than y, the return value will be true. If x is > and == y, the return value will be false.

Examples

- The return value of the following statement is true.
  ```
  2.56<3
  ```
- The return value of the following statement is false.
  ```
  7.55<5.88
  ```

string x < string y

If x is less than y, the return value will be true. If x is > and == y, the return value will be false.

Examples

- The return value of the following statement is true.
  ```
  "string abc" < "string abcd"
  ```
- The return value of the following statement is false.
  ```
  "string abcd" < "string abc"
  ```

time x < time y

If x is less than y, the return value will be true. If x is > and == y, the return value will be false.

Example

The return value of the following statement is false.

```
ToTime(10, 10, 10) < ToTime(8, 10, 10)
```

x == y

date x == date y

If x is equal to y, the return value is true. If x is not equal to y, the return value will be false.

Example

The return value of the following statement is false.

```
ToDate(1999, 10, 10) == ToDate(1999, 5, 10)
```

datetime == datetime y

If x is equal to y, the return value will be true. If x is not equal to y, the return value will be false.

Example

The return value of the following statement is false.

```
ToDateTime(1999, 10, 10, 5, 10, 10) == ToDateTime(1999,10, 10, 10, 10, 10)
```

numeric x == numeric y

If x is equal to y, the return value will be true. If x is not equal to y, the return value will be false.

Example

The return value of the following statement is false.

```
5==4
```

string x == string y

If x is equal to y, the return value is true. If x is not equal to y, the return value will be false.

Example

The return value of the following statement is false.

```
"string abcd" == "string abc"
```

time x == time y

If x is equal to y, the return value will be true. If x is not equal to y, the return value will be false.

Example

The return value of the following statement is false.

```
ToTime(9, 10, 10) == ToTime(10, 10, 10)
```

x <> y or x != y

date x <> date y or date x != date y

If x is not equal to y, the return value will be true. If x is equal to y, the return value will be false.

Example

The return value of the following statement is true.

```
ToDate(1999, 10, 10) != ToDate(1999, 5, 10)
```

datetime x <> datetime y or datetime x != datetime y

If x is not equal to y, the return value will be true. If x is equal to y, the return value will be false.

Example

The return value of the following statement is true.

```
ToDateTime(1999, 10, 10, 5, 10, 10) != ToDateTime(1999,10, 10, 10, 10, 10)
```

numeric x <> numeric y or numeric x != numeric y

If x is not equal to y, the return value will be true. If x is equal to y, the return value will be false.

Example

The return value of the following statement is false.

```
4 <> 4
```

string x <> string y or string x != string y

If x is not equal to y, the return value will be true. If x is equal to y, the return value will be false.

Example

The return value of the following statement is true.

```
"string abc" != "string abcd"
```

time x <> time y or time x != time y

If x is not equal to y, the return value will be true. If x is equal to y, the return value will be false.

Example

The return value of the following statement is true.

```
ToTime(9, 10, 10) <> ToTime(10, 10, 10)
```

Boolean

- x || y
- x && y
- !x

Below is a set of truth tables showing the logic operations for 3-valued logic.

A AND B	True	False	Null
True	True	False	Null
False	False	False	False
Null	Null	False	Null

A OR *B*	True	False	Null
True	True	True	True
False	True	False	Null
Null	True	Null	Null

A	NOT *A*
True	False
False	True
Null	Null

Boolean x || Boolean y

Returns true if x is true or y is true.

Examples

- The return value of the following statement is abc.

```
Integer s=4,d=8;
Integer f=6;
if ( f>s || f>d )
{
return "abc";
}
```

-
```
if ((@"Annual Sales" > 50000) ||(@"Annual Sales"<0)) then
"Customer needs attention"
else
"Normal customer"
```

-
```
If( ! isNull(@customerId) && ! isNull(@country)){
                If(@customerId > 10 || @country == 'USA'){
                                Integer newId =  @customerId + 1;
}
}
Else{
                // Some error processing logic here
}
```

Boolean x && Boolean y

Returns true if x is true and y is true.

Examples

- The return value of the following statement is abc.

```
Integer s=4,d=8;
Integer f=6;
if ( f>s && f<d )
{
return "abc";
}
```

-
```
if ((@"Annual Sales" <3000) && (@Customers_Country=="USA")) then
"Customer is in the USA and Annual Sales is less than 3000"
else
"Others"
```

! Boolean x

Returns true if x is false.

Example

The return value of the following statement is def.

```
Integer s=4, d=8;
Integer f=6;
if ( ! ( f>s ) )
return  "abc"
else
return "def"
```

Others

- x in y
- [x, y, z]
- x[i]
- x[i to j]
- x[i _to j]
- x[i to_ j]
- x[i _to_ j]
- x[upfrom j]
- x[upfrom_ j]
- x[upto j]
- x[upto_ j]
- if(b) then{...} else{...}
- return x

x in y

A statement used to tell whether x is in y, the return value is a boolean value.

Examples

- The return value of the following statement is false.

```
integer x[3] = [0,1,2];
integer xx = 3;
xx in x
```

-
```
integer range intary=[30 to 50];
if (@"Product_ID" in intary) then
return "Within the Range"
else
return "Out of Range"
```

-
```
if (@Customers_Region in ["CO","MT","UT","WY"]) then
"Rocky Mountain Region"
else
"Rest of Country"
```

[x, y, z]

A statement used to define an array.

Example

The return value of the following statement is 1996-11-27.

```
date d[3] = [toDate(1997, 11, 27), toDate(1996, 11, 27), toDate(1995, 11, 27)];
date dd[3];
dd = d;
dd[1]
```

x[i]

A statement used to index a certain value in an array.

Example

The return value of the following statement is 14:11:27.

```
time d[3] = [totime(12, 11, 27), totime(13, 11, 27), totime(14, 11, 27)];
time dd[3];
dd = d;
dd[2]
```

x[i to j]

A statement used to define an array.

Example

The return value of the following statement is true.

```
string range strary = ["a" to "z"];
string x = "d";
x in strary
```

x[i _to j]

It is used to specify a range of values greater than but not including the value i, and less than or equal to the value j. Both i and j are the type of Number.

Example

The return value of the following statement is true.

```
integer a[] = [1,1,2,3,4];
3 in a[2 _to 4];
```

x[i to_ j]

It is used to specify a range of values greater than or equal to the value i, and less than but not including the j value. Both i and j are the type of Number.

Example

The return value of the following statement is true.

```
integer a[] = [1,1,2,3,4];
3 in a[2 to_ 4];
```

x[i _to_ j]

It is used to specify a range of values greater than but not including the value i, and less than but not including the value j. Both i and j are the type of Number.

Example

The return value of the following statement is true.

```
integer a[] = [1,1,2,3,4];
3 in a[2 _to_ 4];
```

x[upfrom j]

It is used to specify a range of values greater than or equal to the value j. j is the type of Number.

Example

The return value of the following statement is false.

```
integer a[] = [1,1,2,3,4];
3 in a[upfrom 4]
```

x[upfrom_ j]

It is used to specify a range of values greater than but not including the value j. j is the type of Number.

Example

The return value of the following statement is true.

```
integer a[] = [1,1,2,3,4];
3 in a[upfrom_ 2];
```

x[upto j]

It is used to specify a range of values less than or equal to the value j. j is the type of Number.

Example

The return value of the following statement is true.

```
integer a[] = [1,1,2,3,4];
3 in a[upto 4];
```

x[upto_ j]

It is used to specify a range of values less than but not including the value j. j is the type of Number.

Example

The return value of the following statement is false.

```
integer a[] = [1,1,2,3,4];
4 in a[upto_ 4];
```

if(b) then{...} else{...}

Conditional statement.

Example

The return value of the following statement is abc.

```
integer s=5;
integer f=8;
if ( f>s )
then return "abc"
else return "def"
```

return x

Returns the value of X.

Example

The result of the following statement is 06/19/01.

```
return Today ()
```

Object Property Reference

This chapter is an overall introduction to the properties of objects in JReport Designer. Keep in mind the following while browsing each object for its properties:

- You will start off from a table listing all the properties for an object and what can be achieved with each property.
- Some properties are just used to show related information and cannot be edited.
- Some properties can be controlled by a formula or expression. If a formula is selected to control a property, it will be automatically generated as an expression prefixed by an @ sign. Also, some properties must be known by JReport Engine before a page break, so that it can lay out the object. In a formula or expression, the special field PageNumber is the switch for setting the page break calculation time. If PageNumber is used in the formula or expression, then the calculation time will be after the page break, if PageNumber is not used, the calculation time will be before the page break. For this reason, JReport will set the calculation time for controlling whether the formula or expression will be marked as before page break or after page break. For most properties, the property can be controlled by formula or expression at any time without concern about the page break.
- For properties with a hyperlink, you can find further explanation. The same property for different objects usually means the same thing, unless otherwise stated.

Note: When you hover the mouse on any property in JReport Designer, a tip will be displayed showing a brief description about the property. You can click ? on the tip to open the corresponding help doc or click ✕ to close the tip. If you do not want to display the property tips, uncheck Show tips in the Options dialog (**File** > **Options** > **General** > **Show tips**).

This chapter explains the properties shown in the Report Inspector and Catalog Browser respectively:

- Properties in the Report Inspector
- Properties in the Catalog Browser

Properties in the Report Inspector

The following types of properties are covered:

- Properties for page report
- Properties for web report
- Properties for library component

Properties for page report

The following are object properties for page report included in the Report Inspector, which are listed in alphabetical order:

- Banded object
- Chart
- Crosstab
- Dataset
- DBField
- Drawing object
- Form
- Formula field
- Geographic map
- Image
- Label
- Multimedia object (OLE object)
- Page panel
- Page report
- Page report tab
- Parameter field
- Report body
- Shape map
- Special field
- Subreport
- Summary field
- Table
- Tabular
- Text box
- TOC
- UDO
- Web control

Banded object

The properties of a banded object in a page report are:

Property Name	Description
General	
Class Type	Indicates the class type of the object. This property is read only.
Data Inherit	Indicates whether the dataset for this object is inherited from another object. This property is read only.
Dataset	Indicates the dataset used by the object. This property is read only.
Instance Name	Shows the instance name of the object. This property is read only.
Geometry	
Height	Specifies the height of the object, in inches or centimeters. Enter a numeric value to change the height. Enabled when it is a horizontal banded object. Data type: Float
Width	Specifies the width of the object, in inches or centimeters. Enter a numeric value to change the width. Enabled when it is a vertical banded object. Data type: Float
X	Specifies the horizontal coordinate of the top left corner of the object, relative to its parent container. This property is ignored if the Position property is set to static. Enter a numeric value to change the position in inches or centimeters. Data type: Float
Y	Specifies the vertical coordinate of the top left corner of the object, relative to its parent container. This property is ignored if the Position property is set to static. Enter a numeric value to change the position in inches or centimeters. Data type: Float
Color	
Background	Specifies the background color of the object. Choose a color from the drop-down list, enter a hexadecimal RGB value (for example, 0xff0000), choose/create a formula or edit an expression that returns a color, or select Custom from the drop-down list to specify the color. Data type: String
CSS	
Class	Specifies a CSS class to be applied to the object which is a valid class in the CSS file. Data type: String
Style	The property can be used in two ways: • Specifies a style to be applied to the object. Choose a style from the drop-down list which is available when a Style Group has been selected at the report tab level and when there are styles in the Style Group that can be applied to the object, or type in the style name. • Specifies a CSS selector to be applied to the object. Enter a valid class name from the CSS file. Data type: String
Others	
Cache	Specifies whether to cache the dataset for this object in the data buffer so other objects which use the same dataset can share the data rather than perform their own SQL query. Data type: Boolean
Default for Filter	Specifies whether to show the data component as the default component in the Apply to drop-down list of the Filter dialog in Page Report Studio. Data type: Boolean **Note:** In a page report tab, only one data component's Default for Filter property can be set to true.
Export to Applet	Specifies whether to include the object when exporting the report to Applet. Data type: Boolean
Export to CSV	Specifies whether to include the object when exporting the report to CSV. Data type: Boolean
Export to Excel	Specifies whether to include the object when exporting the report to Excel. Data type: Boolean
Export to HTML	Specifies whether to include the object when exporting the report to HTML. Data type: Boolean
Export to PDF	Specifies whether to include the object when exporting the report to PDF. Data type: Boolean
Export to PostScript	Specifies whether to include the object when exporting the report to PostScript. Data type: Boolean
Export to Report Result	Specifies whether to include the object when exporting the report to Report Result. Data type: Boolean
Export to RTF	Specifies whether to include the object when exporting the report to RTF. Data type: Boolean
Export to Text	Specifies whether to include the object when exporting the report to Text. Data type: Boolean
Export to XML	Specifies whether to include the object when exporting the report to XML. Data type: Boolean
Invisible	Specifies whether to show the object in the design area and in the report results. All formulas and calculations will still be performed if the property is set to true. Data type: Boolean
Invisible for Filter Dialogs	Specifies whether to show the data component in the Apply to drop-down list of the Filter dialog in Page Report Studio. This property cannot be edited when Default for Filter is set to true. Data type: Boolean
Position	Specifies the position of the object. Choose an option from the drop-down list. • **static** - The object is positioned at the location in which it is inserted. If selected, the X, Y and other position-related properties will be hidden or disabled. The position of the object is affected by preceding objects in the same container. • **absolute** - The object will be located at the position specified by dragging and dropping or by setting its X and Y property values. The position of the object is not affected by preceding objects in the same container. Data type: Enumeration
Record Location	Specifies the calculation point for the properties which use formulas. Choose an option from the drop-down list. • **Default** - The values of the properties are calculated in the default location where they are located. • **Page Header** - The values of the properties are calculated in the page header. • **Page Footer** - The values of the properties are calculated in the page footer. Data type: Enumeration
Suppress	Specifies whether to show the object in the design area and in the report results. All formulas and calculations will be skipped if the property is set to true. Data type: Boolean **Note:** When both the Invisible and Suppress properties of an object are set to true, Suppress has the higher priority.
Suppress When Empty	Specifies whether to display the object in the report results when no record is returned to it. Data type: Boolean
Suppress When No Records	Specifies whether to display the object in the report results when no record is returned to its parent data container. Data type: Boolean
Excel	
Column Index	Specifies the X coordinate of the object relative to its parent container when exported to Excel or CSV, measured in cells. The Columned property at the report tab level must be set to true for this property to take effect. Data type: Integer
Row Index	Specifies the Y coordinate of the object relative to its parent container when exported to Excel or CSV, measured in cells. The Columned property at the report tab level must be set to true for this property to take effect. Data type: Integer
Border	
Border Color	Specifies the color of the border of the object. Choose a color from the drop-down list, enter a hexadecimal RGB value (for example, 0xCCAA00), choose/create a formula or edit an expression that returns a color, or select Custom from the drop-down list to specify the color. Data type: String
Border Joint	Specifies the joint style for the border lines of the object. Choose an option from the drop-down list. • **miter** - Joins path segments by extending their outside edges until they meet. • **round** - Joins path segments by rounding off the corner at a radius of half the line width. Data type: Enumeration
Border Thickness	Specifies the width of the border in inches or centimeters. Enter a numeric value to change the thickness. Data type: Float
Bottom Line	Specifies the line style of the bottom border of the object. Choose a style from the drop-down list. Data type: Enumeration
Left Line	Specifies the line style of the left border of the object. Choose a style from the drop-down list. Data type: Enumeration
Radius	Specifies the radius for the border line joint of the object. This property takes effect only when Border Joint is set to round. Data type: Integer
Right Line	Specifies the line style of the right border of the object. Choose a style from the drop-down list. Data type: Enumeration
Top Line	Specifies the line style of the top border of the object. Choose a style from the drop-down list. Data type: Enumeration
TOC	
Anchor Display Value	Specifies a string or formula to display for the TOC entry for this object in the TOC Browser. By default it is the value of the specified object. The TOC Anchor property must be set to true for this property to take effect. Data type: String
TOC Anchor	Specifies whether to include the object in the TOC Browser for the report. Data type: Boolean
Accessibility	
External CSS Class Selector	Specifies a class selector to be applied to the object when exported as HTML. Enter a valid class name from the CSS file. Data type: String
External ID	It is mapped to the HTML attribute *id*. This attribute specifies a name to the object, which must be unique in the report. Data type: String
Language	It is mapped to the HTML attribute *lang*. This attribute specifies the base language of the object's attribute values and text content. Data type: String

Column Index & Row Index

The two properties together provide a way for you to control the position of an object in the exported CSV or Excel file by specifying values to them.

However, JReport has default values for them which is calculated based on the position of each object in the report. You can find the default values of the two properties right after you set the Columned property at the report tab level to true.

To make the specified values take effect when exporting the report to CSV or Excel, the following is necessary for both exporting formats:

- First set the Columned property in report tab level to true, and then specify values to the two properties of the component.

More requirements for preview as Excel:

- Make sure that the Export to Excel property of the component is set to true.

More requirements for Excel exporting:

- Make sure that the Export to Excel property of the component is set to true.
- When exporting the report to Excel, make sure Column Format is selected in the Export to Excel dialog.

More requirements for CSV exporting:

- Make sure that the Export to CSV property of the component is set to true.
- When exporting the report to Text via dialog, do check the Delimited Format option.

Note: Preview as Text cannot support the specified values for the default setting of Delimited Format is unchecked and it is unconfigurable.

The following topics list properties of the children of a banded object:

- Banded header panel
- Banded page panel
- Banded footer panel
- Banded column group

Banded header panel

The properties of a banded header panel in a page report are:

Property Name	Description
General	
Class Type	Indicates the class type of the object. This property is read only.
Data Inherit	Indicates whether the dataset for this object is inherited from another object. This property is read only.
Dataset	Indicates the dataset used by the object. This property is read only.
Instance Name	Shows the instance name of the object. This property is read only.
Geometry	
Height	Specifies the height of the object, in inches or centimeters. Enter a numeric value to change the height. Data type: Float
Width	Specifies the width of the object, in inches or centimeters. Enter a numeric value to change the width. Data type: Float
Color	
Background	Specifies the background color and fill effect of the object. Choose a color from the drop-down list, enter a hexadecimal RGB value (for example, 0xff0000), choose/create a formula or edit an expression that returns a color, or select Custom from the drop-down list to specify the color and fill effect. Data type: String
CSS	
Class	Specifies a CSS class to be applied to the object which is a valid class in the CSS file. Data type: String
Style	The property can be used in two ways. • Specifies a style to be applied to the object. Choose a style from the drop-down list which is available when a Style Group has been selected at the report tab level and when there are styles in the Style Group that can be applied to the object, or type in the style name. • Specifies a CSS selector to be applied to the object. Enter a valid class name from the CSS file. Data type: String
Others	
Cross Page	Specifies whether the panel can be split across a page break. This property is applied only when the report has page layout enabled. Data type: Boolean
Export to CSV	Specifies whether to include the object when exporting the report to CSV. Data type: Boolean
Export to Page Report Result	Specifies whether to include the object when exporting the report to Page Report Result. Data type: Boolean
Export to Excel	Specifies whether to include the object when exporting the report to Excel. Data type: Boolean
Fill Whole Page	Specifies whether to make the panel extended to the bottom of the page, so that the next panel starts on a new page. Data type: Boolean
Invisible	Specifies whether to show the object in the design area and in the report results. All formulas and calculations will still be performed if the property is set to true. Data type: Boolean
Label	Specifies the popup text displayed when the mouse pointer hovers over the left edge of the panel. Data type: String
Merge to Next Panel	Specifies whether to merge the panel into the next panel in the exported Excel file. Data type: Boolean
On New Page	Specifies whether the panel starts on a new page. This property is applied only when the report has page layout enabled. The default is false which means the panel starts on a new page only if the previous page is filled. Data type: Boolean
Record Location	Specifies the calculation point for the properties which use formulas. Choose an option from the drop-down list. • **Default** - The values of the properties are calculated in the default location where they are located. • **Page Header** - The values of the properties are calculated in the page header. • **Page Footer** - The values of the properties are calculated in the page footer. Data type: Enumeration
Show Bottom Line	Specifies whether to include a horizontal line at the bottom of the panel. Applies to panels in table components only. The default is false, which means that no horizontal line appears at the bottom of the panel. Data type: Boolean
Suppress	Specifies whether to show the object in the design area and in the report results. All formulas and calculations will be skipped if the property is set to true. Data type: Boolean **Note:** When both the Invisible and Suppress properties of an object are set to true, Suppress has the higher priority.
Suppress Blank Panel	Specifies whether a panel with no data is included in the report results. The default is false which means an empty panel is shown. Data type: Boolean
Suppress When No Records	Specifies whether to display the object in the report results when no record is returned to its parent data container. Data type: Boolean
Underlay	Specifies whether the panel will be layered beneath the next panel in the banded object. A panel layered beneath a subsequent panel can be used to contain a background picture or watermark. Banded page header, page footer, or group footer panels cannot be layered. The default is false which means that the panel is not layered beneath the next panel. Data type: Boolean
Border	
Border Color	Specifies the color of the border of the object. Choose a color from the drop-down list, enter a hexadecimal RGB value (for example, 0xCCAA00), choose/create a formula or edit an expression that returns a color, or select Custom from the drop-down list to specify the color. Data type: String
Border Thickness	Specifies the width of the border in inches or centimeters. Enter a numeric value to change the border thickness. Data type: Float
Bottom Line	Specifies the line style of the bottom border of the object. Choose a style from the drop-down list. Data type: Enumeration
Left Line	Specifies the line style of the left border of the object. Choose a style from the drop-down list. Data type: Enumeration
Right Line	Specifies the line style of the right border of the object. Choose a style from the drop-down list. Data type: Enumeration
Top Line	Specifies the line style of the top border of the object. Choose a style from the drop-down list. Data type: Enumeration
Accessibility	
External CSS Class Selector	Specifies a class selector to be applied to the object when exported as HTML. Enter a valid class name from the CSS file. Data type: String
External ID	It is mapped to the HTML attribute *id*. This attribute specifies a name to the object, which must be unique in the report. Data type: String
Language	It is mapped to the HTML attribute *lang*. This attribute specifies the base language of the object's attribute values and text content. Data type: String

Banded page panel

The properties of a banded page panel in a page report are:

Property Name	Description
General	
Class Type	Indicates the class type of the object. This property is read only.
Instance Name	Shows the instance name of the object. This property is read only.
Paper	
Height	Specifies the height of the page in inches or centimeters. The Width and Height values change when Orientation changes. The Page Type setting determines the Width and Height values but you can change the latter as desired. Enter a numeric value to change the height. Data type: Float
Orientation	Specifies how to position the report page. The Orientation setting affects the Width and Height values. Choose an option from the drop-down list. • **portrait** - The page is positioned vertically. • **landscape** - The page is positioned horizontally. Data type: Enumeration
Page Type	Specifies the report page dimensions. Choose a size from the drop-down list. Data type: Enumeration
Width	Specifies the width of the page in inches or centimeters. The Width and Height values change when Orientation changes. The Page Type setting determines the Width and Height values but you can change the latter as desired. Enter a numeric value to change the width. Data type: Float
Margin	
Bottom Margin	Specifies the distance in inches or centimeters between the bottom page edge and the report results. Enter a numeric value to change the margin. Data type: Float
Left Margin	Specifies the distance in inches or centimeters between the left page edge and the report results. Enter a numeric value to change the margin. Data type: Float
Right Margin	Specifies the distance in inches or centimeters between the right page edge and the report results. Enter a numeric value to change the margin. Data type: Float
Top Margin	Specifies the distance in inches or centimeters between the upper page edge and the report results. Enter a numeric value to change the margin. Data type: Float

The following topics list properties of the children of a banded page panel:

- Banded page header panel
- Group panel
- Banded page footer panel

Banded page header panel

The properties of a banded page header panel in a page report are:

Property Name	Description
General	
Class Type	Indicates the class type of the object. This property is read only.
Data Inherit	Indicates whether the dataset for this object is inherited from another object. This property is read only.
Dataset	Indicates the dataset used by the object. This property is read only.
Instance Name	Shows the instance name of the object. This property is read only.
Geometry	
Height	Specifies the height of the object, in inches or centimeters. Enter a numeric value to change the height. Data type: Float
Width	Specifies the width of the object, in inches or centimeters. Enter a numeric value to change the width. Data type: Float
Color	
Background	Specifies the background color and fill effect of the object. Choose a color from the drop-down list, enter a hexadecimal RGB value (for example, 0xff0000), choose/create a formula or edit an expression that returns a color, or select Custom from the drop-down list to specify the color and fill effect. Data type: String
CSS	
Class	Specifies a CSS class to be applied to the object which is a valid class in the CSS file. Data type: String
Style	The property can be used in two ways. • Specifies a style to be applied to the object. Choose a style from the drop-down list which is available when a Style Group has been selected at the report tab level and when there are styles in the Style Group that can be applied to the object, or type in the style name. • Specifies a CSS selector to be applied to the object. Enter a valid class name from the CSS file. Data type: String
Others	
Export to CSV	Specifies whether to include the object when exporting the report to CSV. Data type: Boolean
Export to Page Report Result	Specifies whether to include the object when exporting the report to Page Report Result. Data type: Boolean
Export to Excel	Specifies whether to include the object when exporting the report to Excel. Data type: Boolean
Invisible	Specifies whether to show the object in the design area and in the report results. All formulas and calculations will still be performed if the property is set to true. Data type: Boolean
Label	Specifies the popup text displayed when the mouse pointer hovers over the left edge of the panel. Data type: String
Merge to Next Panel	Specifies whether to merge the panel into the next panel in the exported Excel file. Data type: Boolean
Record Location	Specifies the calculation point for the properties which use formulas. Choose an option from the drop-down list. • **Default** - The values of the properties are calculated in the default location where they are located. • **Page Header** - The values of the properties are calculated in the page header. • **Page Footer** - The values of the properties are calculated in the page footer. Data type: Enumeration
Remove Blank Row	Specifies whether to remove blank rows in the panel in the exported Excel file. Data type: Boolean
Repeat in Detail Panel	Specifies whether to make the objects in the panel repeated in the detail panel in the exported Excel file. Data type: Boolean
Show Bottom Line	Specifies whether to include a horizontal line at the bottom of the panel. Applies to panels in table components only. The default is false, which means that no horizontal line appears at the bottom of the panel. Data type: Boolean
Suppress	Specifies whether to show the object in the design area and in the report results. All formulas and calculations will be skipped if the property is set to true. Data type: Boolean **Note:** When both the Invisible and Suppress properties of an object are set to true, Suppress has the higher priority.
Suppress Blank Panel	Specifies whether a panel with no data is included in the report results. The default is false which means an empty panel is shown. Data type: Boolean
Suppress When No Records	Specifies whether to display the object in the report results when no record is returned to its parent data container. Data type: Boolean
Underlay	Specifies whether the panel will be layered beneath the next panel in the banded object. A panel layered beneath a subsequent panel can be used to contain a background picture or watermark. Banded page header, page footer, or group footer panels cannot be layered. The default is false which means that the panel is not layered beneath the next panel. Data type: Boolean
Border	
Border Color	Specifies the color of the border of the object. Choose a color from the drop-down list, enter a hexadecimal RGB value (for example, 0xCCAA00), choose/create a formula or edit an expression that returns a color, or select Custom from the drop-down list to specify the color. Data type: String
Border Thickness	Specifies the width of the border in inches or centimeters. Enter a numeric value to change the thickness. Data type: Float
Bottom Line	Specifies the line style of the bottom border of the object. Choose a style from the drop-down list. Data type: Enumeration
Left Line	Specifies the line style of the left border of the object. Choose a style from the drop-down list. Data type: Enumeration
Right Line	Specifies the line style of the right border of the object. Choose a style from the drop-down list. Data type: Enumeration
Top Line	Specifies the line style of the top border of the object. Choose a style from the drop-down list. Data type: Enumeration
Accessibility	
External CSS Class Selector	Specifies a class selector to be applied to the object when exported as HTML. Enter a valid class name from the CSS file. Data type: String
External ID	It is mapped to the HTML attribute *id*. This attribute specifies a name to the object, which must be unique in the report. Data type: String
Language	It is mapped to the HTML attribute *lang*. This attribute specifies the base language of the object's attribute values and text content. Data type: String

Group panel

The properties of a group panel in a banded object of a page report are:

Property Name	Description
General	
Class Type	Indicates the class type of the object. This property is read only.
Group By	Shows the group field that the object is based on. If null, the object is grouped based on the whole dataset. This property is read only.
Instance Name	Shows the instance name of the object. This property is read only.
Others	
Current Block Index	The two properties, Current Block Index and Items per Block, work together to control the data of the object to be displayed in continuous page mode. Current Block Index specifies the index of the data block that will be displayed. 0 means the first block index, and 1 the second, and so on. Data type: Integer
Expand Detail Data	Specifies whether to expand the groups to show the details when the report is opened in Page Report Studio. Only works in continuous page mode. Data type: Boolean
Items per Block	The two properties, Current Block Index and Items per Block, work together to control the data of the object to be displayed in continuous page mode. Item per Block specifies the number of records in each data block. Data type: Integer
Shrink Footer	Specifies whether to hide the group footer panel when collapsing a group in Page Report Studio. Only works in continuous page mode. Data type: Boolean
Group Layout	
Keep Group Together	Specifies whether to keep the whole group together. Data type: Boolean
Repeat While Group Footer	If the group header is set to be repeated, you can specify whether to repeat the group header when a page break occurs on the group footer. Data type: Boolean
TOC	
Anchor Display Value	Specifies a string or formula to display for the TOC entry for this object in the TOC Browser. By default it is the value of the specified object. The TOC Anchor property must be set to true for this property to take effect. Data type: String
TOC Anchor	Specifies whether to include the object in the TOC Browser for the report. Data type: Boolean
Security	
Cascade	Specifies whether to allow the security identifiers specified by the following three properties to view the group's detail or child groups. • **true** - The detail panel, the group header panel, and the group footer panel will be displayed. • **false** - Only the group header and footer panels will be displayed. Data type: Boolean
Grant	Enables to set cached report bursting. The property accepts user names delimited by "|" as the value. You can specify which group of data can be viewed by which user. Data type: String
Groups	Enables to set cached report bursting. The property accepts group names delimited by "|" as the value. You can specify which group of data can be viewed by which group of users. Data type: String
Roles	Enables to set cached report bursting. The property accepts role names delimited by "|" as the value. You can specify which group of data can be viewed by which role. Data type: String

The following topics list properties of the children of a group panel:

- Group header panel
- Detail panel
- Group footer panel

Group header panel

The properties of a group header panel in a banded object of a page report are:

Property Name	Description
General	
Class Type	Indicates the class type of the object. This property is read only.
Data Inherit	Indicates whether the dataset for this object is inherited from another object. This property is read only.
Dataset	Indicates the dataset used by the object. This property is read only.
Instance Name	Shows the instance name of the object. This property is read only.
Geometry	
Height	Specifies the height of the object, in inches or centimeters. Enter a numeric value to change the height. Data type: Float
Width	Specifies the width of the object, in inches or centimeters. Enter a numeric value to change the width. Data type: Float
Color	
Background	Specifies the background color and fill effect of the object. Choose a color from the drop-down list, enter a hexadecimal RGB value (for example, 0xff0000), choose/create a formula or edit an expression that returns a color, or select Custom from the drop-down list to specify the color and fill effect. Data type: String
CSS	
Class	Specifies a CSS class to be applied to the object which is a valid class in the CSS file. Data type: String
Style	The property can be used in two ways. • Specifies a style to be applied to the object. Choose a style from the drop-down list which is available when a Style Group has been selected at the report tab level and when there are styles in the Style Group that can be applied to the object, or type in the style name. • Specifies a CSS selector to be applied to the object. Enter a valid class name from the CSS file. Data type: String
Others	
Cross Page	Specifies whether the panel can be split across a page break. This property is applied only when the report has page layout enabled. Data type: Boolean
Export to CSV	Specifies whether to include the object when exporting the report to CSV. Data type: Boolean
Export to Page Report Result	Specifies whether to include the object when exporting the report to Page Report Result. Data type: Boolean
Export to Excel	Specifies whether to include the object when exporting the report to Excel. Data type: Boolean
Fill Whole Page	Specifies whether to make the panel extended to the bottom of the page, so that the next panel starts on a new page. Data type: Boolean
Invisible	Specifies whether to show the object in the design area and in the report results. All formulas and calculations will still be performed if the property is set to true. Data type: Boolean
Label	Specifies the popup text displayed when the mouse pointer hovers over the left edge of the panel. Data type: String
Merge to Next Panel	Specifies whether to merge the panel into the next panel in the exported Excel file. Data type: Boolean
On New Page	Specifies whether the panel starts on a new page. This property is only applied if the report has page layout enabled. The default is false which means the panel starts on a new page only if the previous page is filled. Data type: Boolean
Record Location	Specifies the calculation point for the properties which use formulas. Choose an option from the drop-down list. • **Default** - The values of the properties are calculated in the default location where they are located. • **Page Header** - The values of the properties are calculated in the page header. • **Page Footer** - The values of the properties are calculated in the page footer. Data type: Enumeration
Remove Blank Row	Specifies whether to remove blank rows in the panel in the exported Excel file. Data type: Boolean
Repeat	Specifies whether to make the group header appear in every page. Data type: Boolean
Repeat in Detail Panel	Specifies whether to make the objects in the panel repeated in the detail panel in the exported Excel file. Data type: Boolean
Show Bottom Line	Specifies whether to include a horizontal line at the bottom of the panel. Applies to panels in table components only. The default is false, which means that no horizontal line appears at the bottom of the panel. Data type: Boolean
Suppress	Specifies whether to show the object in the design area and in the report results. All formulas and calculations will be skipped if the property is set to true. Data type: Boolean **Note:** When both the Invisible and Suppress properties of an object are set to true, Suppress has the higher priority.
Suppress Blank Panel	Specifies whether a panel with no data is included in the report results. The default is false which means an empty panel is shown. Data type: Boolean
Suppress When No Records	Specifies whether to display the object in the report results when no record is returned to its parent data container. Data type: Boolean
Underlay	Specifies whether the panel will be layered beneath the next panel in the banded object. A panel layered beneath a subsequent panel can be used to contain a background picture or watermark. Banded page header, page footer, or group footer panels cannot be layered. The default is false which means that the panel is not layered beneath the next panel. Data type: Boolean
Border	
Border Color	Specifies the color of the border of the object. Choose a color from the drop-down list, enter a hexadecimal RGB value (for example, 0xCCAA00), choose/create a formula or edit an expression that returns a color, or select Custom from the drop-down list to specify the color. Data type: String
Border Thickness	Specifies the width of the border in inches or centimeters. Enter a numeric value to change the thickness. Data type: Float
Bottom Line	Specifies the line style of the bottom border of the object. Choose a style from the drop-down list. Data type: Enumeration
Left Line	Specifies the line style of the left border of the object. Choose a style from the drop-down list. Data type: Enumeration
Right Line	Specifies the line style of the right border of the object. Choose a style from the drop-down list. Data type: Enumeration
Top Line	Specifies the line style of the top border of the object. Choose a style from the drop-down list. Data type: Enumeration
Accessibility	
External CSS Class Selector	Specifies a class selector to be applied to the object when exported as HTML. Enter a valid class name from the CSS file. Data type: String
External ID	It is mapped to the HTML attribute *id*. This attribute specifies a name to the object, which must be unique in the report. Data type: String
External Style	It is mapped to the HTML attribute *style*. This attribute specifies style information for the object. Data type: String
Language	It is mapped to the HTML attribute *lang*. This attribute specifies the base language of the object's attribute values and text content. Data type: String

On New Page

If you want each group to start on a new page, you have two approaches. One is to configure with group header panel as follows, the other is to set the Fill Whole Page property for the group footer panel.

There is a problem when you only set On New Page to be true for the group header panel. You will find that the first page will have no details. JREngine starts the detail panel from the second page. To solve this issue, use a formula to control this property. The formula will judge the property as being false for the first group, and being true for subsequent groups for displaying on a new page. You will need to write three formulas and you will also learn how to control property value with a formula. However, if you feel this is too complicated, go to the group footer panel for the second approach.

global:
```
global integer i=0;
```

groupHD:
```
i=i+1;0
```

FormulaControl:
```
@groupHD;
if (i>1)
return true
else
return false
```

Insert formula global into the banded page header panel and assign it an initial value 0. Insert formula groupHD into the group header panel so that for each group i will be calculated as i+1. Then, select the group header panel and let formula FormulaControl control its On New Page property. This formula will only return true for the property if i>1.

Cross Page

For one detail panel which spans a large space vertically and cannot be wholly shown in a page, it will be shown in the next page if this property is set to false. If it is set to true, this panel will be split across a page break when it exceeds the page height.

Repeat

The default value for Repeat is false. For one detail panel which spans a large space vertically to be shown in several pages, we support the Repeat property if you want to see the group header information in each page. When the Repeat property is set true and group headers are visible, the group headers will duplicate on every page. The Repeat property is a Boolean variable and can be changed by a formula.

Notes:

- When a page split occurs in the bottom blank part of the detail panel (or margins in the detail panel) and the Cross Page property in the Detail panel is set true, the output will be: repeated group header + a slice of blank detail panel + group footer. It appears as though the repeated group headers are followed only by group footers without any detail panel. In order to solve this problem, you can resize the detail panel.
- If the Repeat property is set true in both outer group header and inner group header, JReport will treat the two group headers as a completely repeated object.
- If the outer and inner group headers are set to be one completely repeated object, then when the page can only hold the group headers or has no enough space for the group headers, both outer and inner group headers will not be duplicated.
- The Repeat property does not work for subreport.

A group control image can be inserted into the group header panel. See the next section for properties of a group control image.

Group control image

The properties of a group control image in a banded object of a page report are:

Property Name	Description
General	
Class Type	Indicates the class type of the object. This property is read only.
Instance Name	Shows the instance name of the object. This property is read only.
Geometry	
Height	Specifies the height of the object, in inches or centimeters. Enter a numeric value to change the height. Data type: Float
Width	Specifies the width of the object, in inches or centimeters. Enter a numeric value to change the width. Data type: Float
X	Specifies the horizontal coordinate of the top left corner of the object, relative to its parent container. This property is ignored if the Position property is set to static. Enter a numeric value to change the position in inches or centimeters. Data type: Float
Y	Specifies the vertical coordinate of the top left corner of the object, relative to its parent container. This property is ignored if the Position property is set to static. Enter a numeric value to change the position in inches or centimeters. Data type: Float
Color	
Background	Specifies the background color of the object. Choose a color from the drop-down list, enter a hexadecimal RGB value (for example, 0xff0000), choose/create a formula or edit an expression that returns a color, or select Custom from the drop-down list to specify the color. Data type: String
CSS	
Class	Specifies a CSS class to be applied to the object which is a valid class in the CSS file. Data type: String
Style	The property can be used in two ways. • Specifies a style to be applied to the object. Choose a style from the drop-down list which is available when a Style Group has been selected at the report tab level and when there are styles in the Style Group that can be applied to the object, or type in the style name. • Specifies a CSS selector to be applied to the object. Enter a valid class name from the CSS file. Data type: String
Others	
Export to Page Report Result	Specifies whether to include the object when exporting the report to Page Report Result. Data type: Boolean
Invisible	Specifies whether to show the object in the design area and in the report results. All formulas and calculations will still be performed if the property is set to true. Data type: Boolean
Position	Specifies the position of the object. Choose an option from the drop-down list. • **static** - The object is positioned at the location in which it is inserted. If selected, the X, Y and other position-related properties will be hidden or disabled. The position of the object is affected by preceding objects in the same container. • **relative** - The object is positioned at an offset to the position at which it is inserted. The offset is determined by the X and Y property values. • **absolute** - The object will be located at the position specified by dragging and dropping or by setting its X and Y property values. The position of the object is not affected by preceding objects in the same container. Data type: Enumeration
Suppress	Specifies whether to show the object in the design area and in the report results. All formulas and calculations will be skipped if the property is set to true. Data type: Boolean **Note:** When both the Invisible and Suppress properties of an object are set to true, Suppress has the higher priority.
Attribute	
Access Key	Specifies the accelerator key for the object. Data type: String
Alternate Text	Specifies the text that will be displayed if the object cannot be displayed. Data type: String
Checked	Specifies whether the object is checked. Data type: Boolean
Class	This property is typically used to associate a particular style rule in a style sheet with the object. Data type: String
Direction	Specifies a value that indicates the reading order of the object. Data type: String
Disabled	Indicates if the image is disabled. This property is read only.
ID	Specifies the identifier of the object, which must be unique in the report. The ID property can be a style sheet selector. Data type: String
Name	Specifies the name of the image. Data type: String
Size	Specifies the size of the object, in pixels. Data type: Integer
Source	Displays the URL to be loaded by the object. This property is read only.
Tab Index	Specifies the index that defines the tab order for the image. Data type: Integer
Title	Specifies the tip information about the image, which will be displayed when you hover the mouse pointer over the image after the report is exported to HTML or Page Report Result. Data type: String
Value	Specifies the text displayed in the image field. Data type: String
Event	
On Blur	Specifies the action when the object loses input focus. Data type: String
On Change	Specifies the action when the contents of the object or selection have changed. Data type: String
On Click	Specifies the action when the user clicks the left mouse button on the object. Data type: String
On Double Click	Specifies the action when the user double-clicks the object. Data type: String
On Focus	Specifies the action when the object receives focus. Data type: String
On Key Down	Specifies the action when the user presses a key. Data type: String
On Key Press	Specifies the action when the user presses an alphanumeric key. Data type: String
On Key Up	Specifies the action when the user releases a key. Data type: String
On Mouse Down	Specifies the action when the user clicks the object with either mouse button. Data type: String
On Mouse Move	Specifies the action when the user moves the mouse over the object. Data type: String
On Mouse Out	Specifies the action when the user moves the mouse pointer outside the boundaries of the object. Data type: String
On Mouse Over	Specifies the action when the user moves the mouse pointer into the object. Data type: String
On Mouse Up	Specifies the action when the user releases a mouse button while the mouse is over the object. Data type: String
Accessibility	
External CSS Class Selector	Specifies a class selector to be applied to the object when exported as HTML. Enter a valid class name from the CSS file. Data type: String
External ID	It is mapped to the HTML attribute *id*. This attribute specifies a name to the object, which must be unique in the report. Data type: String
External Style	It is mapped to the HTML attribute *style*. This attribute specifies style information for the object. Data type: String
Language	It is mapped to the HTML attribute *lang*. This attribute specifies the base language of the object's attribute values and text content. Data type: String

Detail panel

The properties of a detail panel in a banded object of a page report are:

Property Name	Description
General	
Class Type	Indicates the class type of the object. This property is read only.
Data Inherit	Indicates whether the dataset for this object is inherited from another object. This property is read only.
Dataset	Indicates the dataset used by the object. This property is read only.
Instance Name	Shows the instance name of the object. This property is read only.
Geometry	
Height	Specifies the height of the object, in inches or centimeters. Enter a numeric value to change the height. Data type: Float
Width	Specifies the width of the object, in inches or centimeters. Enter a numeric value to change the width. Data type: Float
Color	
Background	Specifies the background color and fill effect of the object. Choose a color from the drop-down list, enter a hexadecimal RGB value (for example, 0xff0000), choose/create a formula or edit an expression that returns a color, or select Custom from the drop-down list to specify the color and fill effect if necessary. Data type: String
CSS	
Class	Specifies a CSS class to be applied to the object which is a valid class in the CSS file. Data type: String
Style	The property can be used in two ways. • Specifies a style to be applied to the object. Choose a style from the drop-down list which is available when a Style Group has been selected at the report tab level and when there are styles in the Style Group that can be applied to the object, or type in the style name. • Specifies a CSS selector to be applied to the object. Enter a valid class name from the CSS file. Data type: String
Others	
Cache Section	Specifies whether to cache the objects in the detail panel. Data type: Boolean **Note:** The property doesn't work if there is crosstab, text object, chart, and so on in the panel, because these objects contain child-DBFields.
Cross Page	Specifies whether the panel can be split across a page break. This property is applied only when the report has page layout enabled. Data type: Boolean
Current Block Index	The two properties, Current Block Index and Items per Block, work together to control the data of the object to be displayed in continuous page mode. Current Block Index specifies the index of the data block that will be displayed. 0 means the first block index, and 1 the second, and so on. Data type: Integer
Export to CSV	Specifies whether to include the object when exporting the report to CSV. Data type: Boolean
Export to Page Report Result	Specifies whether to include the object when exporting the report to Page Report Result. Data type: Boolean
Export to Excel	Specifies whether to include the object when exporting the report to Excel. Data type: Boolean
Fill Whole Page	Specifies whether to make the panel extended to the bottom of the page, so that the next panel starts on a new page. Data type: Boolean
Invisible	Specifies whether to show the object in the design area and in the report results. All formulas and calculations will still be performed if the property is set to true. Data type: Boolean
Items per Block	The two properties, Current Block Index and Items per Block, work together to control the data of the object to be displayed in continuous page mode. Item per Block specifies the number of records in each data block. Data type: Integer
Label	Specifies the popup text displayed when the mouse pointer hovers over the left edge of the panel. Data type: String
On New Page	Specifies whether the panel starts on a new page. This property is applied only when the report has page layout enabled. The default is false which means the panel starts on a new page only if the previous page is filled. Data type: Boolean
Record Location	Specifies the calculation point for the properties which use formulas. Choose an option from the drop-down list. • **Default** - The values of the properties are calculated in the default location where they are located. • **Page Header** - The values of the properties are calculated in the page header. • **Page Footer** - The values of the properties are calculated in the page footer. Data type: Enumeration
Remove Blank Row	Specifies whether to remove blank rows in the panel in the exported Excel file. Data type: Boolean
Show Bottom Line	Specifies whether to include a horizontal line at the bottom of the panel. Applies to panels in table components only. The default is false, which means that no horizontal line appears at the bottom of the panel. Data type: Boolean
Suppress	Specifies whether to show the object in the design area and in the report results. All formulas and calculations will be skipped if the property is set to true. Data type: Boolean **Note:** When both the Invisible and Suppress properties of an object are set to true, Suppress has the higher priority.
Suppress Blank Panel	Specifies whether a panel with no data is included in the report results. The default is false which means an empty panel is shown. Data type: Boolean
Suppress When No Records	Specifies whether to display the object in the report results when no record is returned to its parent data container. Data type: Boolean
Tile Detail Panel	Specifies whether to tile records in the detail panel. For a vertical/horizontal banded object, you can make each set of records in the detail panel to be tiled when viewing the report if you set this property to true and set the detail panel's width/height equal to or less than 1/2 that of the banded object. The detail panel width/height defines the width/height of each record tile. Data type: Boolean
Tile Horizontal	If the Tile Detail Panel property of the detail panel is set to true, you can further specify the direction for tiling records in the detail panel. • **true** - Records will be tiled in the row direction and then in column. • **false** - Records will be tiled in the column direction and then in row. Data type: Boolean
Underlay	Specifies whether the panel will be layered beneath the next panel in the banded object. A panel layered beneath a subsequent panel can be used to contain a background picture or watermark. Banded page header, page footer, or group footer panels cannot be layered. The default is false which means that the panel is not layered beneath the next panel. Data type: Boolean
Border	
Border Color	Specifies the color of the border of the object. Choose a color from the drop-down list, enter a hexadecimal RGB value (for example, 0xCCAA00), choose/create a formula or edit an expression that returns a color, or select Custom from the drop-down list to specify the color. Data type: String
Border Thickness	Specifies the width of the border in inches or centimeters. Enter a numeric value to change the thickness. Data type: Float
Bottom Line	Specifies the line style of the bottom border of the object. Choose a style from the drop-down list. Data type: Enumeration
Left Line	Specifies the line style of the left border of the object. Choose a style from the drop-down list. Data type: Enumeration
Right Line	Specifies the line style of the right border of the object. Choose a style from the drop-down list. Data type: Enumeration
Top Line	Specifies the line style of the top border of the object. Choose a style from the drop-down list. Data type: Enumeration
Accessibility	
External CSS Class Selector	Specifies a class selector to be applied to the object when exported as HTML. Enter a valid class name from the CSS file. Data type: String
External ID	It is mapped to the HTML attribute *id*. This attribute specifies a name to the object, which must be unique in the report. Data type: String
External Style	It is mapped to the HTML attribute *style*. This attribute specifies style information for the object. Data type: String
Language	It is mapped to the HTML attribute *lang*. This attribute specifies the base language of the object's attribute values and text content. Data type: String

Tile Detail Panel

You will find this property is useful when you are making a mail-label format report.

You can insert a blank banded object in a report, then in the Report Inspector, select the detail panel, and set properties as follows:

Tile Detail Panel = true
Width = *** (width of the detail panel. You can set this value according to the width of one column.

Group footer panel

The properties of a group footer panel in a banded object of a page report are:

Property Name	Description
General	
Class Type	Indicates the class type of the object. This property is read only.
Data Inherit	Indicates whether the dataset for this object is inherited from another object. This property is read only.
Dataset	Indicates the dataset used by the object. This property is read only.
Instance Name	Shows the instance name of the object. This property is read only.
Geometry	
Height	Specifies the height of the object, in inches or centimeters. Enter a numeric value to change the height. Data type: Float
Width	Specifies the width of the object, in inches or centimeters. Enter a numeric value to change the width. Data type: Float
Color	
Background	Specifies the background color and fill effect of the object. Choose a color from the drop-down list, enter a hexadecimal RGB value (for example, 0xff0000), choose/create a formula or edit an expression that returns a color, or select Custom from the drop-down list to specify the color and fill effect. Data type: String
CSS	
Class	Specifies a CSS class to be applied to the object which is a valid class in the CSS file. Data type: String
Style	The property can be used in two ways. • Specifies a style to be applied to the object. Choose a style from the drop-down list which is available when a Style Group has been selected at the report tab level and when there are styles in the Style Group that can be applied to the object, or type in the style name. • Specifies a CSS selector to be applied to the object. Enter a valid class name from the CSS file. Data type: String
Others	
Cross Page	Specifies whether the panel can be split across a page break. This property is applied only when the report has page layout enabled. Data type: Boolean
Export to CSV	Specifies whether to include the object when exporting the report to CSV. Data type: Boolean
Export to Page Report Result	Specifies whether to include the object when exporting the report to Page Report Result. Data type: Boolean
Export to Excel	Specifies whether to include the object when exporting the report to Excel. Data type: Boolean
Fill Whole Page	Specifies whether to make the panel extended to the bottom of the page, so that the next panel starts on a new page. Data type: Boolean
Invisible	Specifies whether to show the object in the design area and in the report results. All formulas and calculations will still be performed if the property is set to true. Data type: Boolean
Label	Specifies the popup text displayed when the mouse pointer hovers over the left edge of the panel. Data type: String
Merge to Next Panel	Specifies whether to merge the panel into the next panel in the exported Excel file. Data type: Boolean
On New Page	Specifies whether the panel starts on a new page. This property is applied only when the report has page layout enabled. The default is false which means the panel starts on a new page only if the previous page is filled. Data type: Boolean
Record Location	Specifies the calculation point for the properties which use formulas. Choose an option from the drop-down list. • **Default** - The values of the properties are calculated in the default location where they are located. • **Page Header** - The values of the properties are calculated in the page header. • **Page Footer** - The values of the properties are calculated in the page footer. Data type: Enumeration
Show Bottom Line	Specifies whether to include a horizontal line at the bottom of the panel. Applies to panels in table components only. The default is false, which means that no horizontal line appears at the bottom of the panel. Data type: Boolean
Suppress	Specifies whether to show the object in the design area and in the report results. All formulas and calculations will be skipped if the property is set to true. Data type: Boolean **Note:** When both the Invisible and Suppress properties of an object are set to true, Suppress has the higher priority.
Suppress Blank Panel	Specifies whether a panel with no data is included in the report results. The default is false which means an empty panel is shown. Data type: Boolean
Suppress When No Records	Specifies whether to display the object in the report results when no record is returned to its parent data container. Data type: Boolean
Underlay	Specifies whether the panel will be layered beneath the next panel in the banded object. A panel layered beneath a subsequent panel can be used to contain a background picture or watermark. Banded page header, page footer, or group footer panels cannot be layered. The default is false which means that the panel is not layered beneath the next panel. Data type: Boolean
Border	
Border Color	Specifies the color of the border of the object. Choose a color from the drop-down list, enter a hexadecimal RGB value (0xCCAA00), choose/create a formula or edit an expression that returns a color, or select Custom from the drop-down list to specify the color. Data type: String
Border Thickness	Specifies the width of the border in inches or centimeters. Enter a numeric value to change the thickness. Data type: Float
Bottom Line	Specifies the line style of the bottom border of the object. Choose a style from the drop-down list. Data type: Enumeration
Left Line	Specifies the line style of the left border of the object. Choose a style from the drop-down list. Data type: Enumeration
Right Line	Specifies the line style of the right border of the object. Choose a style from the drop-down list. Data type: Enumeration
Top Line	Specifies the line style of the top border of the object. Choose a style from the drop-down list. Data type: Enumeration
Accessibility	
External CSS Class Selector	Specifies a class selector to be applied to the object when exported as HTML. Enter a valid class name from the CSS file. Data type: String
External ID	It is mapped to the HTML attribute *id*. This attribute specifies a name to the object, which must be unique in the report. Data type: String
External Style	It is mapped to the HTML attribute *style*. This attribute specifies style information for the object. Data type: String
Language	It is mapped to the HTML attribute *lang*. This attribute specifies the base language of the object's attribute values and text content. Data type: String

Banded page footer panel

The properties of a banded page footer panel in a page report are:

Property Name	Description
General	
Class Type	Indicates the class type of the object. This property is read only.
Data Inherit	Indicates whether the dataset for this object is inherited from another object. This property is read only.
Dataset	Indicates the dataset used by the object. This property is read only.
Instance Name	Shows the instance name of the object. This property is read only.
Geometry	
Height	Specifies the height of the object, in inches or centimeters. Enter a numeric value to change the height. Data type: Float
Width	Specifies the width of the object, in inches or centimeters. Enter a numeric value to change the width. Data type: Float
Color	
Background	Specifies the background color and fill effect of the object. Choose a color from the drop-down list, enter a hexadecimal RGB value (for example, 0xff0000), choose/create a formula or edit an expression that returns a color, or select Custom from the drop-down list to specify the color and fill effect. Data type: String
CSS	
Class	Specifies a CSS class to be applied to the object which is a valid class in the CSS file. Data type: String
Style	The property can be used in two ways. • Specifies a style to be applied to the object. Choose a style from the drop-down list which is available when a Style Group has been selected at the report tab level and when there are styles in the Style Group that can be applied to the object, or type in the style name. • Specifies a CSS selector to be applied to the object. Enter a valid class name from the CSS file. Data type: String
Others	
Export to CSV	Specifies whether to include the object when exporting the report to CSV. Data type: Boolean
Export to Page Report Result	Specifies whether to include the object when exporting the report to Page Report Result. Data type: Boolean
Export to Excel	Specifies whether to include the object when exporting the report to Excel. Data type: Boolean
Invisible	Specifies whether to show the object in the design area and in the report results. All formulas and calculations will still be performed if the property is set to true. Data type: Boolean
Label	Specifies the popup text displayed when the mouse pointer hovers over the left edge of the panel. Data type: String
Record Location	Specifies the calculation point for the properties which use formulas. Choose an option from the drop-down list. • **Default** - The values of the properties are calculated in the default location where they are located. • **Page Header** - The values of the properties are calculated in the page header. • **Page Footer** - The values of the properties are calculated in the page footer. Data type: Enumeration
Show Bottom Line	Specifies whether to include a horizontal line at the bottom of the panel. Applies to panels in table components only. The default is false, which means that no horizontal line appears at the bottom of the panel. Data type: Boolean
Suppress	Specifies whether to show the object in the design area and in the report results. All formulas and calculations will be skipped if the property is set to true. Data type: Boolean **Note:** When both the Invisible and Suppress properties of an object are set to true, Suppress has the higher priority.
Suppress Blank Panel	Specifies whether a panel with no data is included in the report results. The default is false which means an empty panel is shown. Data type: Boolean
Suppress on Last Page	If true, the footer panel on the last page will not be displayed. Data type: Boolean
Suppress When No Records	Specifies whether to display the object in the report results when no record is returned to its parent data container. Data type: Boolean
Underlay	Specifies whether the panel will be layered beneath the next panel in the banded object. A panel layered beneath a subsequent panel can be used to contain a background picture or watermark. Banded page header, page footer, or group footer panels cannot be layered. The default is false which means that the panel is not layered beneath the next panel. Data type: Boolean
Border	
Border Color	Specifies the color of the border of the object. Choose a color from the drop-down list, enter a hexadecimal RGB value (for example, 0xCCAA00), choose/create a formula or edit an expression that returns a color, or select Custom from the drop-down list to specify the color. Data type: String
Border Thickness	Specifies the width of the border in inches or centimeters. Enter a numeric value to change the thickness. Data type: Float
Bottom Line	Specifies the line style of the bottom border of the object. Choose a style from the drop-down list. Data type: Enumeration
Left Line	Specifies the line style of the left border of the object. Choose a style from the drop-down list. Data type: Enumeration
Right Line	Specifies the line style of the right border of the object. Choose a style from the drop-down list. Data type: Enumeration
Top Line	Specifies the line style of the top border of the object. Choose a style from the drop-down list. Data type: Enumeration
Accessibility	
External CSS Class Selector	Specifies a class selector to be applied to the object when exported as HTML. Enter a valid class name from the CSS file. Data type: String
External ID	It is mapped to the HTML attribute *id*. This attribute specifies a name to the object, which must be unique in the report. Data type: String
External Style	It is mapped to the HTML attribute *style*. This attribute specifies style information for the object. Data type: String
Language	It is mapped to the HTML attribute *lang*. This attribute specifies the base language of the object's attribute values and text content. Data type: String

Banded footer panel

The properties of a banded footer panel in a page report are:

Property Name	Description
General	
Class Type	Indicates the class type of the object. This property is read only.
Data Inherit	Indicates whether the dataset for this object is inherited from another object. This property is read only.
Dataset	Indicates the dataset used by the object. This property is read only.
Instance Name	Shows the instance name of the object. This property is read only.
Geometry	
Height	Specifies the height of the object, in inches or centimeters. Enter a numeric value to change the height. Data type: Float
Width	Specifies the width of the object, in inches or centimeters. Enter a numeric value to change the width. Data type: Float
Color	
Background	Specifies the background color and fill effect of the object. Choose a color from the drop-down list, enter a hexadecimal RGB value (for example, 0xff0000), choose/create a formula or edit an expression that returns a color, or select Custom from the drop-down list to specify the color and fill effect. Data type: String
CSS	
Class	Specifies a CSS class to be applied to the object which is a valid class in the CSS file. Data type: String
Style	The property can be used in two ways. • Specifies a style to be applied to the object. Choose a style from the drop-down list which is available when a Style Group has been selected at the report tab level and when there are styles in the Style Group that can be applied to the object, or type in the style name. • Specifies a CSS selector to be applied to the object. Enter a valid class name from the CSS file. Data type: String
Others	
Cross Page	Specifies whether the panel can be split across a page break. This property is applied only when the report has page layout enabled. Data type: Boolean
Export to CSV	Specifies whether to include the object when exporting the report to CSV. Data type: Boolean
Export to Page Report Result	Specifies whether to include the object when exporting the report to Page Report Result. Data type: Boolean
Export to Excel	Specifies whether to include the object when exporting the report to Excel. Data type: Boolean
Fill Whole Page	Specifies whether to make the panel extended to the bottom of the page, so that the next panel starts on a new page. Data type: Boolean
Invisible	Specifies whether to show the object in the design area and in the report results. All formulas and calculations will still be performed if the property is set to true. Data type: Boolean
Label	Specifies the popup text displayed when the mouse pointer hovers over the left edge of the panel. Data type: String
On New Page	Specifies whether the panel starts on a new page. This property is applied only when the report has page layout enabled. The default is false which means the panel starts on a new page only if the previous page is filled. Data type: Boolean
Record Location	Specifies the calculation point for the properties which use formulas. Choose an option from the drop-down list. • **Default** - The values of the properties are calculated in the default location where they are located. • **Page Header** - The values of the properties are calculated in the page header. • **Page Footer** - The values of the properties are calculated in the page footer. Data type: Enumeration
Show Bottom Line	Specifies whether to include a horizontal line at the bottom of the panel. Applies to panels in table components only. The default is false, which means that no horizontal line appears at the bottom of the panel. Data type: Boolean
Suppress	Specifies whether to show the object in the design area and in the report results. All formulas and calculations will be skipped if the property is set to true. Data type: Boolean **Note:** When both the Invisible and Suppress properties of an object are set to true, Suppress has the higher priority.
Suppress Blank Panel	Specifies whether a panel with no data is included in the report results. The default is false which means an empty panel is shown. Data type: Boolean
Suppress When No Records	Specifies whether to display the object in the report results when no record is returned to its parent data container. Data type: Boolean
To Bottom	Specifies whether to move the banded object footer panel to the bottom of the page. The default is false, which means that the banded object footer appears at the end of the banded object. Data type: Boolean
Underlay	Specifies whether the panel will be layered beneath the next panel in the banded object. A panel layered beneath a subsequent panel can be used to contain a background picture or watermark. Banded page header, page footer, or group footer panels cannot be layered. The default is false which means that the panel is not layered beneath the next panel. Data type: Boolean
Border	
Border Color	Specifies the color of the border of the object. Choose a color from the drop-down list, enter a hexadecimal RGB value (for example, 0xCCAA00), choose/create a formula or edit an expression that returns a color, or select Custom from the drop-down list to specify the color. Data type: String
Border Thickness	Specifies the width of the border in inches or centimeters. Enter a numeric value to change the thickness. Data type: Float
Bottom Line	Specifies the line style of the bottom border of the object. Choose a style from the drop-down list. Data type: Enumeration
Left Line	Specifies the line style of the left border of the object. Choose a style from the drop-down list. Data type: Enumeration
Right Line	Specifies the line style of the right border of the object. Choose a style from the drop-down list. Data type: Enumeration
Top Line	Specifies the line style of the top border of the object. Choose a style from the drop-down list. Data type: Enumeration
Accessibility	
External CSS Class Selector	Specifies a class selector to be applied to the object when exported as HTML. Enter a valid class name from the CSS file. Data type: String
External ID	It is mapped to the HTML attribute *id*. This attribute specifies a name to the object, which must be unique in the report. Data type: String
External Style	It is mapped to the HTML attribute *style*. This attribute specifies style information for the object. Data type: String
Language	It is mapped to the HTML attribute *lang*. This attribute specifies the base language of the object's attribute values and text content. Data type: String

Banded column group

If you open a table-style report created by JReport Designer of a version lower than 8, you may find that the table is actually a banded object, and the columns of the table are organized as a banded column group which may contain one or more columns. The banded column group is displayed as BandedColumnInfos in the Report Inspector by default.

The properties of a banded column group in a page report are:

Property Name	Description
General	
Class Type	Indicates the class type of the object. This property is read only.
Data Inherit	Indicates whether the dataset for this object is inherited from another object. This property is read only.
Dataset	Indicates the dataset used by the object. This property is read only.
Instance Name	Shows the instance name of the object. This property is read only.
Color	
Background	Specifies the background color of the object. Choose a color from the drop-down list, enter a hexadecimal RGB value (for example, 0xff0000), choose/create a formula or edit an expression that returns a color, or select Custom from the drop-down list to specify the color. Data type: String
CSS	
Class	Specifies a CSS class to be applied to the object which is a valid class in the CSS file. Data type: String
Style	The property can be used in two ways. • Specifies a style to be applied to the object. Choose a style from the drop-down list which is available when a Style Group has been selected at the report tab level and when there are styles in the Style Group that can be applied to the object, or type in the style name. • Specifies a CSS selector to be applied to the object. Enter a valid class name from the CSS file. Data type: String
Others	
Column Widths	Specifies the column width values, measured in inches. Type the width values delimited by comma. If you want to use the Width property of a column, leave its place blank. Data type: String
Export to CSV	Specifies whether to include the object when exporting the report to CSV. Data type: Boolean
Export to Page Report Result	Specifies whether to include the object when exporting the report to Page Report Result. Data type: Boolean
Export to Excel	Specifies whether to include the object when exporting the report to Excel. Data type: Boolean
Pattern	Specifies the style group for the columns. Data type: String
Position	Specifies the position of the object. Choose an option from the drop-down list. • **static** - The object is positioned at the location in which it is inserted. The X and Y properties are ignored. The position of the object is affected by preceding objects in the same container. • **relative** - The object is positioned at an offset to the position at which it is inserted. The offset is determined by the X and Y property values. • **absolute** - The object will be located at the position specified by dragging and dropping or by setting its X and Y property values. The position of the object is not affected by preceding objects in the same container. Data type: Enumeration
Record Location	Specifies the calculation point for the properties which use formulas. Choose an option from the drop-down list. • **Default** - The values of the properties are calculated in the default location where they are located. • **Page Header** - The values of the properties are calculated in the page header. • **Page Footer** - The values of the properties are calculated in the page footer. Data type: Enumeration
Show Columns	Specifies the columns to be shown in the report result. Type the column mapping names delimited by comma. For example: Shipping Cost, Ship Date, Ship Via, Payment Received. Data type: String
Suppress When No Records	Specifies whether to display the object in the report results when no record is returned to its parent data container. Data type: Boolean
Border	
Border Color	Specifies the color of the border of the object. Choose a color from the drop-down list, enter a hexadecimal RGB value (for example, 0xCCAA00), choose/create a formula or edit an expression that returns a color, or select Custom from the drop-down list to specify the color. Data type: String
Border Thickness	Specifies the width of the border in inches or centimeters. Enter a numeric value to change the thickness. Data type: Float
Bottom Line	Specifies the line style of the bottom border of the object. Choose a style from the drop-down list. Data type: Enumeration
Left Line	Specifies the line style of the left border of the object. Choose a style from the drop-down list. Data type: Enumeration
Right Line	Specifies the line style of the right border of the object. Choose a style from the drop-down list. Data type: Enumeration
Show Border	Specifies whether to show the border. Data type: Boolean
Top Line	Specifies the line style of the top border of the object. Choose a style from the drop-down list. Data type: Enumeration
Accessibility	
External CSS Class Selector	Specifies a class selector to be applied to the object when exported as HTML. Enter a valid class name from the CSS file. Data type: String
External ID	It is mapped to the HTML attribute *id*. This attribute specifies a name to the object, which must be unique in the report. Data type: String
Language	It is mapped to the HTML attribute *lang*. This attribute specifies the base language of the object's attribute values and text content. Data type: String

See the next section for the properties of a banded column.

Banded column

The properties of a banded column in a page report are:

Property Name	Description
General	
Class Type	Indicates the class type of the object. This property is read only.
Instance Name	Shows the instance name of the object. This property is read only.
Geometry	
Width	Specifies the width of the object, in inches or centimeters. Enter a numeric value to change the width. Data type: Float
Color	
Background	Specifies the background color of the object. Choose a color from the drop-down list, enter a hexadecimal RGB value (for example, 0xff0000), choose/create a formula or edit an expression that returns a color, or select Custom from the drop-down list to specify the color. Data type: String
CSS	
Class	Specifies a CSS class to be applied to the object which is a valid class in the CSS file. Data type: String
Style	The property can be used in two ways. • Specifies a style to be applied to the object. Choose a style from the drop-down list which is available when a Style Group has been selected at the report tab level and when there are styles in the Style Group that can be applied to the object, or type in the style name. • Specifies a CSS selector to be applied to the object. Enter a valid class name from the CSS file. Data type: String
Others	
Export to CSV	Specifies whether to include the object when exporting the report to CSV. Data type: Boolean
Export to Page Report Result	Specifies whether to include the object when exporting the report to Page Report Result. Data type: Boolean
Export to Excel	Specifies whether to include the object when exporting the report to Excel. Data type: Boolean
Position	Specifies the position of the object. Choose an option from the drop-down list. • **static** - The object is positioned at the location in which it is inserted. If selected, the X, Y and other position-related properties will be hidden or disabled. The position of the object is affected by preceding objects in the same container. • **relative** - The object is positioned at an offset to the position at which it is inserted. The offset is determined by the X and Y property values. • **absolute** - The object will be located at the position specified by dragging and dropping or by setting its X and Y property values. The position of the object is not affected by preceding objects in the same container. Data type: Enumeration
Record Location	Specifies the calculation point for the properties which use formulas. Choose an option from the drop-down list. • **Default** - The values of the properties are calculated in the default location where they are located. • **Page Header** - The values of the properties are calculated in the page header. • **Page Footer** - The values of the properties are calculated in the page footer. Data type: Enumeration
Suppress	Specifies whether to show the object in the design area and in the report results. All formulas and calculations will be skipped if the property is set to true. Data type: Boolean
Suppress When No Records	Specifies whether to display the object in the report results when no record is returned to its parent data container. Data type: Boolean
Border	
Show Border	Specifies whether to show the border. Data type: Boolean
Accessibility	
External CSS Class Selector	Specifies a class selector to be applied to the object when exported as HTML. Enter a valid class name from the CSS file. Data type: String
External ID	It is mapped to the HTML attribute *id*. This attribute specifies a name to the object, which must be unique in the report. Data type: String
External Style	It is mapped to the HTML attribute *style*. This attribute specifies style information for the object. Data type: String
External Title	It is mapped to the HTML attribute *title*. This attribute offers advisory information about the object. Data type: String
Language	It is mapped to the HTML attribute *lang*. This attribute specifies the base language of the object's attribute values and text content. Data type: String

Chart

A chart mainly consists of three parts: the platform, paper, and legend. The chart platform is a container for other chart elements, and the chart paper holds walls, axes, data markers, axis values, and category names.

The following topics are detailed introduction to properties of the elements:

- Chart platform
- Chart paper
- Chart legend
- Chart label
- A detailed chart property reference

Chart platform

A chart platform is a container for other chart elements. You can use the properties listed in the following table to control the appearances or behavior of a chart platform. In the Report Inspector, a chart platform is displayed as ChartObject by default.

The properties of a chart platform in a page report are:

Property Name	Description
General	
Class Type	Indicates the class type of the object. This property is read only.
Data Inherit	Indicates whether the dataset for this object is inherited from another object. This property is read only.
Dataset	Indicates the dataset used by the object. This property is read only.
Instance Name	Shows the instance name of the object. This property is read only.
Geometry	
Height	Specifies the height of the object, in inches or centimeters. Enter a numeric value to change the height. Data type: Float
Width	Specifies the width of the object, in inches or centimeters. Enter a numeric value to change the width. Data type: Float
X	Specifies the horizontal coordinate of the top left corner of the object, relative to its parent container. This property is ignored if the Position property is set to static. Enter a numeric value to change the position in inches or centimeters. Data type: Float
Y	Specifies the vertical coordinate of the top left corner of the object, relative to its parent container. This property is ignored if the Position property is set to static. Enter a numeric value to change the position in inches or centimeters. Data type: Float
CSS	
Class	Specifies a CSS class to be applied to the object which is a valid class in the CSS file. Data type: String
Style	The property can be used in two ways. • Specifies a style to be applied to the object. Choose a style from the drop-down list which is available when a Style Group has been selected at the report tab level and when there are styles in the Style Group that can be applied to the object, or type in the style name. • Specifies a CSS selector to be applied to the object. Enter a valid class name from the CSS file. Data type: String
Others	
Alternative Text	Specifies the alternate text of the image, which will be shown when the image cannot be displayed. This property applies only when the chart is exported to HTML as an image chart. Enter a string to serve as the alternative text. Data type: String
Cache	Specifies whether to cache the dataset for this object in the data buffer so other objects which use the same dataset can share the data rather than perform their own SQL query. Data type: Boolean
Default for Filter	Specifies whether to show the data component as the default component in the Apply to drop-down list of the Filter dialog in Page Report Studio. Data type: Boolean **Note:** In a page report tab, only one data component's Default for Filter property can be set to true.
Export to Applet	Specifies whether to include the object when exporting the report to Applet. Data type: Boolean
Export to Excel	Specifies whether to include the object when exporting the report to Excel. Data type: Boolean
Export to HTML	Specifies whether to include the object when exporting the report to HTML. Data type: Boolean
Export to PDF	Specifies whether to include the object when exporting the report to PDF. Data type: Boolean
Export to PostScript	Specifies whether to include the object when exporting the report to PostScript. Data type: Boolean
Export to Report Result	Specifies whether to include the object when exporting the report to Report Result. Data type: Boolean
Export to RTF	Specifies whether to include the object when exporting the report to RTF. Data type: Boolean
Invisible	Specifies whether to show the object in the design area and in the report results. All formulas and calculations will still be performed if the property is set to true. Data type: Boolean
Invisible for Filter Dialogs	Specifies whether to show the data component in the Apply to drop-down list of the Filter dialog in Page Report Studio. This property cannot be edited when Default for Filter is set to true. Data type: Boolean
Logic Column	Specifies whether to show the object in the next visible table cell in the same row when the column which holds the object is hidden. Choose an option from the drop-down list. Data type: Enumeration **Notes:** • The next visible table cell should be completely empty, that is, the cell should not hold any content including blank space. • When the property for several objects in the same row is set to next visible column, and the columns holding these objects are all hidden, then only the object in the rightmost column will be shown in the next visible cell.
Position	Specifies the position of the object. Choose an option from the drop-down list. • **static** - The object is positioned at the location in which it is inserted. If selected, the X, Y and other position-related properties will be hidden or disabled. The position of the object is affected by preceding objects in the same container. • **absolute** - The object will be located at the position specified by dragging and dropping or by setting its X and Y property values. The position of the object is not affected by preceding objects in the same container. Data type: Enumeration
Suppress	Specifies whether to show the object in the design area and in the report results. All formulas and calculations will be skipped if the property is set to true. Data type: Boolean **Note:** When both the Invisible and Suppress properties of an object are set to true, Suppress has the higher priority.
Suppress When No Records	Specifies whether to display the object in the report results when no record is returned to its parent data container. Data type: Boolean
Excel	
Column Index	Specifies the X coordinate of the object relative to its parent container when exported to Excel or CSV, measured in cells. The Columned property at the report tab level must be set to true for this property to take effect. Data type: Integer
Column Number	Specifies the number of columns which will be the object's width in the exported Excel or CSV file. The Columned property at the report tab level must be set to true for this property to take effect. Data type: String
Row Index	Specifies the Y coordinate of the object relative to its parent container when exported to Excel or CSV, measured in cells. The Columned property at the report tab level must be set to true for this property to take effect. Data type: Integer
Row Number	Specifies the number of rows which will be the object's height in the exported Excel or CSV file. The Columned property at the report tab level must be set to true for this property to take effect. Data type: String
TOC	
Anchor Display Value	Specifies a string or formula to display for the TOC entry for this object in the TOC Browser. By default it is the value of the specified object. The TOC Anchor property must be set to true for this property to take effect. Data type: String
TOC Anchor	Specifies whether to include the object in the TOC Browser for the report. Data type: Boolean
Platform (unavailable to org chart)	
Anti-aliasing	Specifies whether to make the edges in a chart smooth. Data type: Boolean
Color When Null	Specifies the color when the value is null. This property is available to heat map only and is applied when the heat map is colored only by a summary. Choose a color from the drop-down list, enter a hexadecimal RGB value (for example, 0xff0000), or select Custom from the drop-down list to specify the color. Data Type: String
Color When Zero	Specifies the color mapping to the zero value. Available to heat map only. It works when the heat map is colored only by a summary and the Using Color When Zero property is set to true. Choose a color from the drop-down list, enter a hexadecimal RGB value (for example, 0xff0000), or select Custom from the drop-down list to specify the color. Data Type: String
End Color	Specifies the end color of the gradient colors. This property is available to heat map only and is applied when the heat map is colored only by a summary. Choose a color from the drop-down list, enter a hexadecimal RGB value (for example, 0xff0000), or select Custom from the drop-down list to specify the color. Data type: String
Minimum Tick Mark Space	Specifies the minimum space between two adjacent tick marks. Enter a numeric value to change the space. Data type: Float
Pattern List	Specifies patterns of the data markers. Click [...] in the value cell and click the small squares in the color tray one by one to specify the patterns for the data markers. The patterns can be one or more of the following: Automatic, Color, Texture, and Gradient. For heat maps, the Pattern List property works only when using single color and using groups as the color-by fields. Data type: String
Reverse	Specifies whether to reverse the start color and the end color. This property is available to heat map only and is applied when the heat map is colored only by a summary. Data type: Boolean
Show Legend	Specifies whether to make the legend visible in a chart. Data type: Boolean
Start Color	Specifies the start color of the gradient colors. This property is available to heat map only and is applied when the heat map is colored only by a summary. Choose a color from the drop-down list, enter a hexadecimal RGB value (for example, 0xff0000), or select Custom from the drop-down list to specify the color. Data type: String
Using Color When Zero	Specifies whether to map the zero value to the color specified by the property Color When Zero. • If true and the minimum value < 0 < the maximum value, the colors will be from the start color to the color when zero first, and then from the color when zero to the end color. • If false, the colors will be from the start color to the end color directly. This property is available to heat map only and is applied when the heat map is colored only by a summary. Data type: Boolean
Background	
Border Color	Specifies the color of the platform border. Choose a color from the drop-down list, enter a hexadecimal RGB value (for example, 0xCCAA00), choose/create a formula or edit an expression that returns a color, or select Custom from the drop-down list to specify the color. Data type: String
Border End Caps	Specifies the ending style of the platform border line. Choose an option from the drop-down list. Data type: Enumeration
Border Joint	Specifies the joint style of the platform border line. Choose an option from the drop-down list. Data type: Enumeration
Border Outlined	Specifies whether to show the platform border line in Outline Path. If the property is set to true, the platform border line will be shown in Outline Path; otherwise, in Fill Path. Data type: Boolean
Border Style	Specifies the line style of the platform border. Choose an option from the drop-down list. Data type: Enumeration
Border Thickness	Specifies the width of the platform border, in inches or centimeters. Enter a numeric value to change the thickness. Data type: Float
Border Transparency	Specifies the transparency of the platform border, in percent. Enter a numeric value to change the transparency. Data type: Integer
Border Type	Specifies the border type of the platform. Choose an option from the drop-down list. Data type: Enumeration
Border Variable Dash	Specifies whether to resize the dash automatically. • **true** - If selected, the dash size will be adjusted automatically, and the option Auto Adjust Dash in the Format Platform dialog will take effect. • **false** - If selected, the dash size will be of fixed size, and the option Fixed Dash Size in the Format Platform dialog will take effect. Data type: Boolean
Color	Specifies the color of the platform background, applied when the property Fill Type is set to color. Choose a color from the drop-down list, enter a hexadecimal RGB value (for example, 0xff0000), choose/create a formula or edit an expression that returns a color, or select Custom from the drop-down list to specify the color and fill effect. Data type: String
Fill Transparency	Specifies the transparency of the platform background, in percent. Enter a numeric value to change the transparency. Data type: Integer
Fill Type	Specifies the fill pattern for the platform background. Choose an option from the drop-down list. Data type: Enumeration
Gradient End Color	Specifies the color of the point where the gradient ends, applied when the property Fill Type is set to gradient. Choose a color from the drop-down list, enter a hexadecimal RGB value (for example, 0xff0000), choose/create a formula or edit an expression that returns a color, or select Custom from the drop-down list to specify the color and fill effect. Data type: String
Gradient End X	Specifies the horizontal position where the gradient ends, measured in a percentage of the platform width. Applied when the property Fill Type is set to gradient. Data type: Integer
Gradient End Y	Specifies the vertical position where the gradient ends, measured in a percentage of the platform height. Applied when the property Fill Type is set to gradient. Data type: Integer
Gradient Start Color	Specifies the color of the point where the gradient begins, applied when the property Fill Type is set to gradient. Choose a color from the drop-down list, enter a hexadecimal RGB value (for example, 0xff0000), choose/create a formula or edit an expression that returns a color, or select Custom from the drop-down list to specify the color and fill effect. Data type: String
Gradient Start X	Specifies the horizontal position where the gradient begins, measured in a percentage of the platform width. Applied when the property Fill Type is set to gradient. Data type: Integer
Gradient Start Y	Specifies the vertical position where the gradient begins, measured in a percentage of the platform height. Applied when the property Fill Type is set to gradient. Data type: Integer
Gradient Style	Specifies the gradient style for the platform background, applied when the property Fill Type is set to gradient. Choose an option from the drop-down list. Data type: Enumeration
Image File	Specifies the file name of the image, a portion of which will be displayed as the platform background. Applied when Fill Type is set to image. Enter the file name with suffix in the value cell. If the image is outside of the current catalog, you should include the full path of the image correctly. Data type: String
Image Height	The four properties, Image X, Image Y, Image Width, and Image Height, together are used to define a portion of the image (the portion is a rectangle), which will be displayed as the platform background. They take effect when the property Fill Type is set to image. Image Height specifies the height of the image portion, measured in a percentage of the image height. Data type: Integer
Image Layout	Specifies the layout style for the image that will be displayed as the platform background, applied when the property Fill Type is set to image. Choose an option from the drop-down list. Data type: Enumeration
Image Width	The four properties, Image X, Image Y, Image Width, and Image Height, together are used to define a portion of the image (the portion is a rectangle), which will be displayed as the platform background. They take effect when the property Fill Type is set to image. Image Width specifies the width of the image portion, measured in a percentage of the image width. Data type: Integer
Image X	The four properties, Image X, Image Y, Image Width, and Image Height, together are used to define a portion of the image (the portion is a rectangle), which will be displayed as the platform background. They take effect when the property Fill Type is set to image. Image X specifies the distance between the left border of image and the portion that will contain the pattern, measured in a percentage of the image width. Data type: Integer
Image Y	The four properties, Image X, Image Y, Image Width, and Image Height, together are used to define a portion of the image (the portion is a rectangle), which will be displayed as the platform background. They take effect when the property Fill Type is set to image. Image Y specifies the distance between the top border of image and the portion that will contain the pattern, measured in a percentage of the image height. Data type: Integer
Inset Bottom	Specifies the bottom position of the background area, measured in a percentage of the chart height, from the bottom edge of the chart. Data type: Float
Inset Left	Specifies the left position of the background area, measured in a percentage of the chart width, from the left edge of the chart. Data type: Float
Inset Right	Specifies the right position of the background area, measured in a percentage of the chart width, from the right edge of the chart. Data type: Float
Inset Top	Specifies the top position of the background area, measured in a percentage of the chart height, from the top edge of the chart. Data type: Float
Radius	Specifies the radius for the joint of the platform border line. The property takes effect only when Border Joint is set to joint round. Enter a numeric value to change the radius. Data type: Float
Texture Background Color	Specifies the background color of the texture, applied when the property Fill Type is set to texture. Choose a color from the drop-down list, enter a hexadecimal RGB value (for example, 0xff0000), choose/create a formula or edit an expression that returns a color, or select Custom from the drop-down list to specify the color. Data type: String
Texture Foreground Color	Specifies the foreground color of the texture, applied when the property Fill Type is set to texture. Choose a color from the drop-down list, enter a hexadecimal RGB value (for example, 0xff0000), choose/create a formula or edit an expression that returns a color, or select Custom from the drop-down list to specify the color. Data type: String
Texture Style	Specifies the texture style of the platform background, applied when the property Fill Type is set to texture. Choose an option from the drop-down list. Data type: Enumeration
Data (unavailable to org chart)	
For heat map	
Color by Logarithm	Specifies whether to calculate the color-by summary based on logarithm function. Available to heat map only. Choose an option from the drop-down list: • **None** Returns the real value of the field instead of any logarithmic value. • **log x** Gets the logarithm of the field with base 10. • **In x** Gets the logarithm of the field with base e. Data type: String
Size by Logarithm	Specifies whether to calculate the size-by summary based on logarithm function. Available to heat map only. Choose an option from the drop-down list: • **None** Returns the real value of the field instead of any logarithmic value. • **log x** Gets the logarithm of the field with base-10. • **In x** Gets the logarithm of the field with base-e. Data type: String
For other chart types	
Category Data Mapping File	Specifies the data mapping file to the object for NLS use, which takes effect for category values shown on the category (X) axis and the hints of the data markers. For example, if the data mapping file name is Category_de_DE.properties, set the value as Category (without the language and locale part). Data type: String
Category End Offset	The four properties work together to control the range of the data being displayed on a chart: Category Start Offset, Category End Offset, Series Start Offset, and Series End Offset. Category End Offset specifies the ending offset of the categories. It does not apply to the charts which involve only one group. Data type: Integer
Category Format	Specifies the data format for the category axis (the X axis) to display the tick mark labels in the way you choose. Click [...] in the value cell to specify the format in the Data Format dialog. Data type: String
Category Start Offset	The four properties work together to control the range of the data being displayed on a chart: Category Start Offset, Category End Offset, Series Start Offset, and Series End Offset. Category Start Offset specifies the starting offset of the categories. It does not apply to the charts which involve only one group. Data type: Integer
Category Value Encoding	Specifies the encoding format for values on the category axis. Choose an option from the drop-down list. Data type: String
Go to Detail	Specifies whether to show the detailed information when you click the object in Page Report Studio. This property is available only when the object is in the group header/ footer panel of a banded object. For details about its usage, refer to Obtaining detailed information from a banded object. Data type: Boolean
Hint Percent Format	Specifies the format for each hint percentage to the total. Click [...] in the value cell to specify the format in the Data Format dialog. Data type: String
Hint Value Format	Specifies the number format for the hint message. Click [...] in the value cell to specify the format in the Data Format dialog. Data type: String
Hyperlink	Specifies whether to add a hyperlink that refers to another report or a website to the data markers of the chart. Data type: String
Hyperlink Target	Specifies the target window or frame in which to display the content the Hyperlink property specifies. Choose an option from the drop-down list. • **<Server Setting>** Loads the linked file according to setting of the Pop Up New Window for Links option in the Customize Profile panel of the Profile dialog in JReport Server. • **New Window** Loads the linked file into a new window. The window is not named. • **Whole Window** Loads the linked file into the full browser window. • **Same Frame** Loads the linked file into the same frame as the link. • **Parent Frame** Loads the linked file into the parent frame of the frame that contains the link. • **Other Frame** Loads the linked file into some other specified frame. If selected, you can also directly input the name of the frame you have defined in the text box. If the frame name does not exist, the linked file will be loaded into a new window. Data type: String
Motion Bar Format	Specifies the data format for the motion field values displayed on the motion bar. Click [...] in the value cell to specify the format in the Data Format dialog. Data type: String
Overall Series	Specifies whether to calculate the top or bottom N category values based on the series values. Data type: Boolean
Primary Data Format	Specifies the data format for the primary value axis (the Y1 axis) to display the tick mark labels in the way you choose. Click [...] in the value cell to specify the format in the Data Format dialog. Data type: String
Reverse Category	Specifies whether to reverse the sequence of the category field value. Data type: Boolean
Reverse Series	Specifies whether to reverse the sequence of the series field value. Data type: Boolean
Secondary Data Format	Specifies the data format for the secondary value axis (the Y2 axis) to display the tick mark labels in the way you choose. Click [...] in the value cell to specify the format in the Data Format dialog. Data type: String
Series Data Mapping File	Specifies the data mapping file to the object for NLS use, which takes effect for series values shown in the legend and the hints of the data markers. Not available to bubble chart. For example, if the data mapping file name is Category_de_DE.properties, set the value as Category (without the language and locale part). Data type: String
Series End Offset	The four properties work together to control the range of the data being displayed on a chart: Category Start Offset, Category End Offset, Series Start Offset, and Series End Offset. Series End Offset specifies the ending offset of the series. Data type: Integer
Series Format	Specifies the data format for the series axis (the Z axis) to display the tick mark labels in the way you choose. Click [...] in the value cell to specify the format in the Data Format dialog. Data type: String
Series Start Offset	The four properties work together to control the range of the data being displayed on a chart: Category Start Offset, Category End Offset, Series Start Offset, and Series End Offset. Series Start Offset specifies the starting offset of the series. Data type: Integer
Series Value Encoding	Specifies the encoding format for values on the series axis. Choose an option from the drop-down list. Data type: String
Sort Category	Specifies the sorting order for the category field values. Choose an option from the drop-down list. Data type: Enumeration
Sort Series	Specifies the sorting order for the series field values. Choose an option from the drop-down list. Data type: Enumeration
Swap Groups	Specifies to display values from different data fields by switching data between the category and series axes, the category and value axes of a chart. By default, this property is set to false, which means no switch will take place in the chart. • **Switching between the category and series axes** If you create a chart by adding fields to all of its axes, and set Swap Groups to true, data on the category and series axes will be switched. • **Switching between the category and value axes** If you create a chart by adding one field to the category axis and several to the value axis, and set Swap Groups to true, data on the category and value axes will be switched. Data type: Boolean **Note:** If the chart has only one field on the category axis and one on the value axis, even though the property value is set to true, there will be no switch take place on the two axes.
X HyperLink	Specifies whether to add a hyperlink that refers to another report or a website to the X axis labels. Data type: String
X Hyperlink Target	Specifies the target window or frame in which to display the content the X Hyperlink property specifies. Choose an option from the drop-down list. • **<Server Setting>** Loads the linked file according to setting of the Pop Up New Window for Links option in the Customize Profile panel of the Profile dialog in JReport Server. • **New Window** Loads the linked file into a new window. The window is not named. • **Whole Window** Loads the linked file into the full browser window. • **Same Frame** Loads the linked file into the same frame as the link. • **Parent Frame** Loads the linked file into the parent frame of the frame that contains the link. • **Other Frame** Loads the linked file into some other specified frame. If selected, you can also directly input the name of the frame you have defined in the text box. If the frame name does not exist, the linked file will be loaded into a new window. Data type: String
Z HyperLink	Specifies whether to add a hyperlink that refers to another report or a website to the Z axis labels. Data type: String
Z Hyperlink Target	Specifies the target window or frame in which to display the content the Z Hyperlink property specifies. Choose an option from the drop-down list. • **<Server Setting>** Loads the linked file according to setting of the Pop Up New Window for Links option in the Customize Profile panel of the Profile dialog in JReport Server. • **New Window** Loads the linked file into a new window. The window is not named. • **Whole Window** Loads the linked file into the full browser window. • **Same Frame** Loads the linked file into the same frame as the link. • **Parent Frame** Loads the linked file into the parent frame of the frame that contains the link. • **Other Frame** Loads the linked file into some other specified frame. If selected, you can also directly input the name of the frame you have defined in the text box. If the frame name does not exist, the linked file will be loaded into a new window. Data type: String
Accessibility	
External CSS Class Selector	Specifies a class selector to be applied to the object when exported as HTML. Enter a valid class name from the CSS file. Data type: String
External ID	A representation of the standard HTML attribute *id*. This attribute specifies a name to the object, which must be unique in the report. Data type: String
External Title	A representation of the standard HTML attribute *title*. This attribute offers advisory information about the object. Data type: String
LongDesc	A representation of the standard HTML attribute *longdesc*. This attribute specifies a link to a long description of the object. Data type: String

Chart paper

The properties of a chart paper in a page report are:

Property Name	Description
General	
Class Type	Indicates the class type of the object. This property is read only.
Data Inherit	Indicates whether the dataset for this object is inherited from another object. This property is read only.
Dataset	Indicates the dataset used by the object. This property is read only.
Instance Name	Shows the instance name of the object. This property is read only.
Geometry	
Height	Specifies the height of the object, in inches or centimeters. Enter a numeric value to change the height. Data type: Float
Width	Specifies the width of the object, in inches or centimeters. Enter a numeric value to change the width. Data type: Float
X	Specifies the horizontal coordinate of the top left corner of the object, relative to its parent container. Enter a numeric value to change the position in inches or centimeters. This property takes effect only when the Placement property for the chart legend is set to customized. Data type: Float
Y	Specifies the vertical coordinate of the top left corner of the object, relative to its parent container. Enter a numeric value to change the position in inches or centimeters. This property takes effect only when the Placement property for the chart legend is set to customized. Data type: Float
CSS	
Class	Specifies a CSS class to be applied to the object which is a valid class in the CSS file. Data type: String
Style	The property can be used in two ways. • Specifies a style to be applied to the object. Choose a style from the drop-down list which is available when a Style Group has been selected at the report tab level and when there are styles in the Style Group that can be applied to the object, or type in the style name. • Specifies a CSS selector to be applied to the object. Enter a valid class name from the CSS file. Data type: String
Accessibility	
External CSS Class Selector	Specifies a class selector to be applied to the object when exported as HTML. Enter a valid class name from the CSS file. Data type: String
Coordinate Paper (unavailable to heat map and org charts)	
Angle X	Specifies the rotation angle of the chart around the X axis, in degrees. Applies to 3-D charts only. Enter a numeric value to change the rotation. Data type: Float
Angle Y	Specifies the rotation angle of the chart around the Y axis, in degrees. Applies to 3-D charts only. Enter a numeric value to change the rotation. Data type: Float
Angle Z	Specifies the rotation angle of the chart around the Z axis, in degrees. Applies to 3-D charts only. Enter a numeric value to change the rotation. Data type: Float
Graph Position	Specifies the position of the paper contents relative to the paper background. Applies to 2-D charts only. Choose an option from the drop-down list. Data type: Enumeration
Interactive	Specifies whether the chart will be interactive in Designer view mode, in HTML result, and in Page Report Studio. When it is set to true, all the interactive actions are enabled, for example, you can drill around the chart, put the mouse pointer over a target to get its full hint, and so on. Data type: Boolean
Perspective	Specifies how to display the chart in perspective view. By default no perspective chart will be displayed. Applies to 3-D charts only. Data type: Integer
Reverse Painting Order	Specifies whether to reverse the order of graphic objects in a combo chart. Data type: Boolean
Scale X	Specifies the percentage for elongating or shortening the X axis. Enter a numeric value to change the percentage. Data type: Float
Scale Y	Specifies the percentage for elongating or shortening the Y axis. Enter a numeric value to change the percentage. Data type: Float
Scale Z	Specifies the percentage for elongating or shortening the Z axis. Applies to 3-D chart only. Enter a numeric value to change the percentage. Data type: Float
Shadow Contents	Specifies whether to add shadows to all 2-D charts. Data type: Boolean
Background	
Border Color	Specifies the color of the paper border. Choose a color from the drop-down list, enter a hexadecimal RGB value (for example, 0xCCAA00), choose/create a formula or edit an expression that returns a color, or select Custom from the drop-down list to specify the color. Data type: String
Border End Caps	Specifies the ending style of the paper border line. Choose an option from the drop-down list. Data type: Enumeration
Border Joint	Specifies the joint style of the paper border line. Choose an option from the drop-down list. Data type: Enumeration
Border Outlined	Specifies whether to show the paper border line in Outline Path. If the property is set to true, the paper border line will be shown in Outline Path; otherwise, in Fill Path. Data type: Boolean
Border Style	Specifies the line style of the paper border. Choose an option from the drop-down list. Data type: Enumeration
Border Thickness	Specifies the width of the paper border, in inches or centimeters. Enter a numeric value to change the thickness. Data type: Float
Border Transparency	Specifies the transparency of the paper border, in percent. Enter a numeric value to change the transparency. Data type: Integer
Border Type	Specifies the border type of the paper. Choose an option from the drop-down list. Data type: Enumeration
Border Variable Dash	Specifies whether to resize the dash automatically. • **true** - The dash size will be adjusted automatically. And the option Auto Adjust Dash in the Format Paper dialog will take effect. • **false** - The dash size will be of fixed size. And the option Fixed Dash Size in the Format Paper dialog will take effect. Data type: Boolean
Color	Specifies the color of the paper background, applied when the property Fill Type is set to color. Choose a color from the drop-down list, enter a hexadecimal RGB value (for example, 0xff0000), choose/create a formula or edit an expression that returns a color, or select Custom from the drop-down list to specify the color. Data type: String
Fill Transparency	Specifies the transparency of the paper background, in percent. Enter a numeric value to change the transparency. Data type: Integer
Fill Type	Specifies the fill pattern for the paper background. Choose an option from the drop-down list. Data type: Enumeration
Gradient End Color	Specifies the color of the point where the gradient ends, applied when the property Fill Type is set to gradient. Choose a color from the drop-down list, enter a hexadecimal RGB value (for example, 0xff0000), choose/create a formula or edit an expression that returns a color, or select Custom from the drop-down list to specify the color. Data type: String
Gradient End X	Specifies the horizontal position where the gradient ends, measured in a percentage of the paper width. Applied when the property Fill Type is set to gradient. Data type: Integer
Gradient End Y	Specifies the vertical position where the gradient ends, measured in a percentage of the paper height. Applied when the property Fill Type is set to gradient. Data type: Integer
Gradient Start Color	Specifies the color of the point where the gradient begins, applied when the property Fill Type is set to gradient. Choose a color from the drop-down list, enter a hexadecimal RGB value (for example, 0xff0000), choose/create a formula or edit an expression that returns a color, or select Custom from the drop-down list to specify the color. Data type: String
Gradient Start X	Specifies the horizontal position where the gradient begins, measured in a percentage of the paper width. Applied when the property Fill Type is set to gradient. Data type: Integer
Gradient Start Y	Specifies the vertical position where the gradient begins, measured in a percentage of the paper height. Applied when the property Fill Type is set to gradient. Data type: Integer
Gradient Style	Specifies the gradient style for the paper background, applied when the property Fill Type is set to gradient. Choose an option from the drop-down list. Data type: Enumeration
Image File	Specifies the file name of the image, a portion of which will be displayed as the paper background. Applied when the property Fill Type is set to image. Enter the file name with suffix in the value cell. If the image is outside of the current catalog, you should include the full path of the image correctly. Data type: String
Image Height	The four properties, Image X, Image Y, Image Width, and Image Height, together are used to define a portion of the image (the portion is a rectangle), which will be displayed as the paper background. They take effect when the property Fill Type is set to image. Image Height specifies the height of the image portion, measured in a percentage of the image height. Data type: Integer
Image Layout	Specifies the layout style for the image that will be displayed as the paper background, applied when the property Fill Type is set to image. Choose an option from the drop-down list. Data type: Enumeration
Image Width	The four properties, Image X, Image Y, Image Width, and Image Height, together are used to define a portion of the image (the portion is a rectangle), which will be displayed as the paper background. They take effect when the property Fill Type is set to image. Image Width specifies the width of the image portion, measured in a percentage of the image width. Data type: Integer
Image X	The four properties, Image X, Image Y, Image Width, and Image Height, together are used to define a portion of the image (the portion is a rectangle), which will be displayed as the paper background. They take effect when the property Fill Type is set to image. Image X specifies the distance between the left border of image and the portion that will contain the pattern, measured in a percentage of the image width. Data type: Integer
Image Y	The four properties, Image X, Image Y, Image Width, and Image Height, together are used to define a portion of the image (the portion is a rectangle), which will be displayed as the paper background. They take effect when the property Fill Type is set to image. Image Y specifies the distance between the top border of image and the portion that will contain the pattern, measured in a percentage of the image height. Data type: Integer
Inset Bottom	Specifies the bottom position of the background area, measured in a percentage of the paper height, from the bottom edge of the paper. Data type: Float
Inset Left	Specifies the left position of the background area, measured in a percentage of the paper width, from the left edge of the paper. Data type: Float
Inset Right	Specifies the right position of the background area, measured in a percentage of the paper width, from the right edge of the paper. Data type: Float
Inset Top	Specifies the top position of the background area, measured in a percentage of the paper height, from the top edge of the paper. Data type: Float
Texture Background Color	Specifies the background color of the texture, applied when the property Fill Type is set to texture. Choose a color from the drop-down list, enter a hexadecimal RGB value (for example, 0xff0000), choose/create a formula or edit an expression that returns a color, or select Custom from the drop-down list to specify the color. Data type: String
Texture Foreground Color	Specifies the foreground color of the texture, applied when the property Fill Type is set to texture. Choose a color from the drop-down list, enter a hexadecimal RGB value (for example, 0xff0000), choose/create a formula or edit an expression that returns a color, or select Custom from the drop-down list to specify the color. Data type: String
Texture Style	Specifies the texture style of the paper background, applied when the property Fill Type is set to texture. Choose an option from the drop-down list. Data type: Enumeration
Others (available to org charts only)	
Layout Mode	Specifies how the org chart tree expands from the root node. Choose an option from the drop-down list. • **top down** The org chart tree expands from top to bottom. • **bottom up** The org chart tree expands from bottom to top. It is the reverse of the top down mode. • **left right** The org chart tree expands from left to right. Data type: Enumeration
Graph (unavailable to surface, heat map and org charts)	
Border	Specifies whether to show the borders. Data type: Boolean
Border Color	Specifies the line color for the border, not applying to 2-D line charts. Choose a color from the drop-down list, enter a hexadecimal RGB value (for example, 0xff0000), choose/create a formula or edit an expression that returns a color, or select Custom from the drop-down list to specify the color. Data type: String
Border End Caps	Specifies the ending style for the border, not applying to 2-D line charts. Choose an option from the drop-down list. Data type: Enumeration
Border Joint	Specifies the line joint style for the border, not applying to 2-D line charts. Choose an option from the drop-down list. Data type: Enumeration
Border Outlined	Specifies whether to show the border in outline form, not applying to 2-D line charts. If the property is set to true, the border will be shown in outline form; otherwise, in whole form. Data type: Boolean
Border Style	Specifies the line style for the border, not applying to 2-D line charts. Choose an option from the drop-down list. Data type: Enumeration
Border Thickness	Specifies the thickness for the border, not applying to 2-D line charts. Enter a numeric value to change the thickness. Data type: Float
Border Transparency	Specifies the line transparency for the border, in percent, not applying to 2-D line charts. Enter a numeric value to change the transparency. Data type: Integer
Border Variable Dash	Specifies whether to resize the dash automatically, not applying to 2-D line charts. If the property is set to true, the dash size will be adjusted automatically; otherwise, the dash size will be of fixed size. Data type: Boolean
Suppress Label When 0	If true, the data label whose value is 0 will not be displayed. This property is available to these 2-D chart types: bar, bench, line, area, pie and radar. Data type: Boolean
For bar/bench chart	
Bar Gap	Specifies the distance between each bar/bench in a category. Applies to 2-D clustered bar/bench charts only. Enter a numeric value to change the gap. Data type: Integer
Bar Style	Specifies the style of the bars/benches. Choose an option from the drop-down list. It can be normal or cylinder, however, if you specify the style to be cylinder, it will take effect only when the property Use Depth is set to true. Data type: Enumeration
Bar Width	Specifies the total width of the bars/benches in each category, measured in a percentage of the unit width. Enter a numeric value to change the width. Data type: Integer
Depth	Specifies the depth of the bars/benches, applied when the value of the property Use Depth is set to true. Applies to 2-D bar/bench charts only. Enter a numeric value to change the depth. Data type: Float
Direction	Specifies the angle of the axis along the depth of the bars/benches. Applies to 2-D bar/bench charts only. Enter a numeric value to change the direction. Data type: Integer
Use Depth	Specifies whether to make the bars/benches visually three-dimensional. Applies to 2-D bar/bench charts only. Data type: Boolean **Note:** If the chart is a combo chart composed by bars and areas/lines, when you set the depth properties for the bars, they will be applied to the areas/lines as well, and vice versa.
Value Label Type	Specifies in which way the value labels will be displayed around the bars/benches. Applied only when Show Static Data Label is checked in the Format Bar dialog. Choose an option from the drop-down list. • **value** - Shows the value for each bar/bench. • **percent** - Shows the percentage of each bar/bench to the total. • **value and percent** - Shows the value and the percentage for each bar/bench. Data type: Enumeration
Vary Colors by Color List	Specifies whether or not to make the bar colors vary. It only takes effect on clustered bar/bench types and when the charts has no series field. Data type: Boolean
For line chart	
Area Pattern List	Specifies patterns for the areas which are formed by the chart axes and the lines. Applies to 2-D lines only. Click ... in the value cell and click the small squares in the color tray one by one to specify the patterns for the areas. The patterns can be one or more of the following: Automatic, Color, Texture, and Gradient. Data type: String
Depth	Specifies the depth of the lines, applied when the property Use Depth is set to true. Applies to 2-D charts only. Enter a numeric value to change the depth. Data type: Float
Direction	Specifies the angle of the axis along the depth of the lines. Applies to 2-D charts only. Enter a numeric value to change the direction. Data type: Integer
Ignore Null Value	Specifies whether to skip null data values in order to draw continuous lines. • **true** - When a null data value appears, it is skipped and the line is drawn from the previous data value to the next data value directly. • **false** - When a null data value appears, draw it anyway and the line is broken at the point of the null data value. Data type: Boolean
Use Depth	Specifies whether to make the lines visually three-dimensional. Applies to 2-D line charts only. Data type: Boolean **Note:** If the chart is a combo chart composed by lines and areas/bars, when you set the depth properties for the lines, they will be applied to the areas/bars as well, and vice versa.
Value Label Type	Specifies in which way the value labels will be displayed around the lines. Applied only when Show Static Data Label is checked in the Format Line dialog. Available to 2-D line charts only. Choose an option from the drop-down list. • **value** - Shows the value for each line node. • **percent** - Shows the percentage of each line node to the total. • **value and percent** - Shows the value and the percentage for each line node. Data type: Enumeration
For area chart	
Depth	Specifies the depth of the areas, applied when the property Use Depth is set to true. Applies to 2-D area charts only. Enter a numeric value to change the depth. Data type: Float
Direction	Specifies the angle of the axis along the depth of the areas. Applies to 2-D area charts only. Enter a numeric value to change the direction. Data type: Integer
Ignore Null Value	Specifies whether to skip null data values in order to draw continuous areas. • **true** - When a null data value appears, it is skipped and the area is drawn from the previous data value to the next data value directly. • **false** - When a null data value appears, draw it anyway and the area is broken at the point of the null data value. Data type: Boolean
Use Depth	Specifies whether to make the areas visually three-dimensional. Applies to 2-D area charts only. Data type: Boolean **Note:** If the chart is a combo chart composed by areas and bars/lines, when you set the depth properties for the areas, they will be applied to the bars/lines as well, and vice versa.
Use Dropline	Specifies whether to show the lines that represent the data categories. Applies to 2-D area charts only. Data type: Boolean
Value Label Type	Specifies in which way the value labels will be displayed around the areas. Applied only when Show Static Data Label is checked in the Format Area dialog. Choose an option from the drop-down list. • **value** - Shows the value for each area node. • **percent** - Shows the percentage of each area node to the total. • **value and percent** - Shows the value and the percentage for each area node. Data type: Enumeration
For pie chart	
Angle X	Specifies the rotation angle of the chart around the X axis, in degrees. Enter a numeric value to change the rotation. Data type: Float
Angle Y	Specifies the rotation angle of the chart around the Y axis, in degrees. Enter a numeric value to change the rotation. Data type: Float
Donut Hole	Specifies the radius percentage of the donut hole to the total pie circle. Applies to donut chart only. Data type: Float
Pie Gap	Specifies the distance between every two adjacent pie sections. Enter a numeric value to change the gap. Data type: Float
Section Explode Gap	Specifies the distance between each pie section and the pie center. Enter a numeric value to change the gap. Data type: Float
Show Pie/Donut Name	Specifies whether to show the name of the pie/donut. Data type: Boolean
Value Label Type	Specifies in which way the value labels will be displayed around the pie. Applied only when Show Static Data Label is checked in the Format Pie dialog. Choose an option from the drop-down list. • **value** - Shows the value for the pie section. • **category name** - Shows the category name for the pie section. • **percent** - Shows the percentage of the section to the total. • **value and percent** - Shows the value and the percentage for the pie section. Data type: Enumeration
For radar chart	
Fill Radar	Specifies whether to fill the areas formed by value nodes of the same data series in the radar chart. Data type: Boolean
Line Thickness	Specifies the thickness of the line. Enter a numeric value to change the thickness. Data type: Float
Node Height	Specifies the height of the value nodes. Enter a numeric value to change the height. Data type: Float
Node Style	Specifies the line node style for the data series lines. Applies to 2-D charts that contain lines only. Choose an option from the drop-down list. Data type: Enumeration
Node Width	Specifies the width of the value nodes. Enter a numeric value to change the width. Data type: Float
Radar Arrow Style	Specifies the arrow style for the axes. Choose an option from the drop-down list. Data type: Enumeration
Radar Fill Transparency	Specifies the transparency for the background in the radar chart, in percent. Enter a numeric value to change the transparency. Data type: Integer
Show Category Name	Specifies whether to show category names. Data type: Boolean
Value Label Type	Specifies in which way the value labels will be displayed around the value nodes. Applied only when Show Static Data Label is checked in the Format Radar dialog. Choose an option from the drop-down list. • **value** - Shows the value for each value node. • **percent** - Shows the percentage of each value node to the total. • **value and percent** - Shows the value and the percentage for each value node. Data type: Enumeration
For indicator chart	
Horizontal Margin	Specifies the distance in inches or centimeters between the left indicators and the left border of the chart paper. Data type: Float
Range Radius	Specifies the relative size of an indicator in a percentage of the total indicator size. Data type: Integer
Vertical Margin	Specifies the distance in inches or centimeters between the top indicators and the top border of the chart paper. Data type: Float
For gauge chart	
Alert Color	Specifies the color for the alert range. Choose a color from the drop-down list, enter a hexadecimal RGB value (for example, 0xff0000), choose/create a formula or edit an expression that returns a color, or select Custom from the drop-down list to specify the color. Data type: String
Alert Maximum Value	Specifies the maximum value for the alert range. Enter a numeric value to specify the value. Data type: Float
Alert Minimum Value	Specifies the minimum value for the alert range. Enter a numeric value to specify the value. Data type: Float
Arrow Style	Specifies the style for pointer arrows, applied when the property Pointer Style is set to Arrow. Applies to dial gauge charts only. Choose an option from the drop-down list. Data type: Enumeration
Curve Style	Specifies the style for the curve. Applies to dial gauge charts only. Choose an option from the drop-down list. Data type: Enumeration
Error Color	Specifies the color for the error range. Choose a color from the drop-down list, enter a hexadecimal RGB value (for example, 0xff0000), choose/create a formula or edit an expression that returns a color, or select Custom from the drop-down list to specify the color. Data type: String
Error Maximum Value	Specifies the maximum value for the error range. Enter a numeric value to specify the value. Data type: Float
Error Minimum Value	Specifies the minimum value for the error range. Enter a numeric value to specify the value. Data type: Float
Fill Curve	Specifies whether to fill the dial with colors. Applies to dial gauge charts only. Data type: Boolean
Gauge Alert Gradient End Color	Specifies the color of the point where the gradient ends for the alert range, applied when Gauge Alert Type is set to gradient. Choose a color from the drop-down list, enter a hexadecimal RGB value (for example, 0xff0000), choose/create a formula or edit an expression that returns a color, or select Custom from the drop-down list to specify the color. Data type: String
Gauge Alert Gradient End X	Specifies the horizontal position where the gradient ends, measured in a percentage of the alert range width. Applied when Gauge Alert Type is set to gradient. Data type: Integer
Gauge Alert Gradient End Y	Specifies the vertical position where the gradient ends, measured in a percentage of the alert range height. Applied when Gauge Alert Type is set to gradient. Data type: Integer
Gauge Alert Gradient Start Color	Specifies the color of the point where the gradient begins for the alert range, applied when Gauge Alert Type is set to gradient. Choose a color from the drop-down list, enter a hexadecimal RGB value (for example, 0xff0000), choose/create a formula or edit an expression that returns a color, or select Custom from the drop-down list to specify the color. Data type: String
Gauge Alert Gradient Start X	Specifies the horizontal position where the gradient begins, measured in a percentage of the alert range width. Applied when Gauge Alert Type is set to gradient. Data type: Integer
Gauge Alert Gradient Start Y	Specifies the vertical position where the gradient begins, measured in a percentage of the alert range height. Applied when Gauge Alert Type is set to gradient. Data type: Integer
Gauge Alert Gradient Style	Specifies the gradient style for the alert range, applied when Gauge Alert Type is set to gradient. Choose an option from the drop-down list. Data type: Enumeration
Gauge Alert Transparency	Specifies the transparency of the alert range background, in percent. Enter a numeric value to change the transparency. Data type: Integer
Gauge Alert Type	Specifies the fill pattern for the alert range. Choose an option from the drop-down list. Data type: Enumeration
Gauge Bar Border Color	Specifies the color of the border of the object. Applies to bar gauge charts only. Choose a color from the drop-down list, enter a hexadecimal RGB value (for example, 0xCCAA00), choose/create a formula or edit an expression that returns a color, or select Custom from the drop-down list to specify the color. Data type: String
Gauge Bar Border End Caps	Specifies the ending style of the scale border line. Applies to bar gauge charts only. Choose an option from the drop-down list. Data type: Enumeration
Gauge Bar Border Joint	Specifies the joint style of the scale border line. Applies to bar gauge charts only. Choose an option from the drop-down list. Data type: Enumeration
Gauge Bar Border Outlined	Specifies whether to show the scale border line in Outline Path. Applies to bar gauge charts only. If the property is set to true, the border line will be shown in Outline Path; otherwise, in Fill Path. Data type: Boolean
Gauge Bar Border Style	Specifies the line style of the scale borders. Applies to bar gauge charts only. Choose an option from the drop-down list. Data type: Enumeration
Gauge Bar Border Thickness	Specifies the width of the border in inches or centimeters. Applies to bar gauge charts only. Enter a numeric value to change the thickness. Data type: Float
Gauge Bar Border Transparency	Specifies the transparency of the borders, in percent. Applies to bar gauge charts only. Enter a numeric value to change the transparency. Data type: Integer
Gauge Bar Border Variable Dash	Specifies whether to resize the dash automatically. Applies to bar gauge charts only. If the property is set to true, the dash size will be adjusted automatically; otherwise, the dash size will be of fixed size. Data type: Boolean
Gauge Bar Layout	Specifies the layout for the scales. Applies to bar gauge charts only. Choose an option from the drop-down list. Data type: Enumeration
Gauge Error Gradient End Color	Specifies the color of the point where the gradient ends for the error range, applied when Gauge Error Type is set to gradient. Choose a color from the drop-down list, enter a hexadecimal RGB value (for example, 0xff0000), choose/create a formula or edit an expression that returns a color, or select Custom from the drop-down list to specify the color. Data type: String
Gauge Error Gradient End X	Specifies the horizontal position where the gradient ends, measured in a percentage of the error range width. Applied when Gauge Error Type is set to gradient. Data type: Integer
Gauge Error Gradient End Y	Specifies the vertical position where the gradient ends, measured in a percentage of the error range height. Applied when Gauge Error Type is set to gradient. Data type: Integer
Gauge Error Gradient Start Color	Specifies the color of the point where the gradient begins for the error range, applied when Gauge Error Type is set to gradient. Choose a color from the drop-down list, enter a hexadecimal RGB value (for example, 0xff0000), choose/create a formula or edit an expression that returns a color, or select Custom from the drop-down list to specify the color. Data type: String
Gauge Error Gradient Start X	Specifies the horizontal position where the gradient begins, measured in a percentage of the error range width. Applied when Gauge Error Type is set to gradient. Data type: Integer
Gauge Error Gradient Start Y	Specifies the vertical position where the gradient begins, measured in a percentage of the error range height. Applied when Gauge Error Type is set to gradient. Data type: Integer
Gauge Error Gradient Style	Specifies the gradient style for the error range, applied when Gauge Error Type is set to gradient. Choose an option from the drop-down list. Data type: Enumeration
Gauge Error Transparency	Specifies the transparency of the error range background, in percent. Enter a numeric value to change the transparency. Data type: Integer
Gauge Error Type	Specifies the fill pattern for the error range. Choose an option from the drop-down list. Data type: Enumeration
Gauge Normal Gradient End Color	Specifies the color of the point where the gradient ends for the normal range, applied when Gauge Normal Type is set to gradient. Choose a color from the drop-down list, enter a hexadecimal RGB value (for example, 0xff0000), choose/create a formula or edit an expression that returns a color, or select Custom from the drop-down list to specify the color. Data type: String
Gauge Normal Gradient End X	Specifies the horizontal position where the gradient ends, measured in a percentage of the normal range width. Applied when Gauge Normal Type is set to gradient. Data type: Integer
Gauge Normal Gradient End Y	Specifies the vertical position where the gradient ends, measured in a percentage of the normal range height. Applied when Gauge Normal Type is set to gradient. Data type: Integer
Gauge Normal Gradient Start Color	Specifies the color of the point where the gradient begins for the normal range, applied when Gauge Normal Type is set to gradient. Choose a color from the drop-down list, enter a hexadecimal RGB value (for example, 0xff0000), choose/create a formula or edit an expression that returns a color, or select Custom from the drop-down list to specify the color. Data type: String
Gauge Normal Gradient Start X	Specifies the horizontal position where the gradient begins, measured in a percentage of the normal range width. Applied when Gauge Normal Type is set to gradient. Data type: Integer
Gauge Normal Gradient Start Y	Specifies the vertical position where the gradient begins, measured in a percentage of the normal range height. Applied when Gauge Normal Type is set to gradient. Data type: Integer
Gauge Normal Gradient Style	Specifies the gradient style for the normal range, applied when Gauge Normal Type is set to gradient. Choose an option from the drop-down list. Data type: Enumeration
Gauge Normal Transparency	Specifies the transparency of the normal range background, in percent. Enter a numeric value to change the transparency. Data type: Integer
Gauge Normal Type	Specifies the fill pattern for the normal range. Choose an option from the drop-down list. Data type: Enumeration
Horizontal Margin	Specifies the distance in inches or centimeters between the left bubbles and the left border of the chart paper. Applies to bubble gauge charts only. Enter a numeric value to change the margin. Data type: Float
Normal Color	Specifies the color for the normal range. Choose a color from the drop-down list, enter a hexadecimal RGB value (for example, 0xff0000), choose/create a formula or edit an expression that returns a color, or select Custom from the drop-down list to specify the color. Data type: String
Normal Maximum Value	Specifies the maximum value for the normal range. Enter a numeric value to specify the value. Data type: Float
Normal Minimum Value	Specifies the minimum value for the normal range. Enter a numeric value to specify the value. Data type: Float
Outer Inner Gap	Specifies the width of the dial. Applies to dial gauge charts only. Enter a numeric value to change the gap. Data type: Float
Pointer Style	Specifies the style for the pointer. Applies to dial gauge charts only. Choose an option from the drop-down list. Data type: Enumeration
Range Radius	Specifies the relative size of a dial or ball. Applies to dial and bubble gauge charts only. Enter a numeric value to change the size. Data type: Integer
Start Angle	Specifies the angle of the dial beginning point. Applies to dial gauge charts only. Enter a numeric value to change the angle. Data type: Integer
Stop Angle	Specifies the angle of the dial endpoint. Applies to dial gauge charts only. Enter a numeric value to change the angle. Data type: Integer
Vertical Margin	Specifies the distance in inches or centimeters between the top bubbles and the top border of the chart paper. Applies to bubble gauge charts only. Enter a numeric value to change the margin. Data type: Float
For scatter chart	
Line Thickness	Specifies the thickness of the line. Applies to the scatter chart which contains lines. Enter a numeric value to change the thickness. Data type: Float
Node Height	Specifies the height of the value nodes. Enter a numeric value to change the height. Data type: Float
Node Style	Specifies the line node style for the data series lines. Applies to 2-D charts that contain lines only. Choose an option from the drop-down list. Data type: Enumeration
Node Width	Specifies the width of the value nodes. Enter a numeric value to change the width. Data type: Float
For bubble chart	
Use 3D Effect	Specifies whether to make the bubbles visually three-dimensional. Data type: Boolean
Cut Bubble Based on XY Area	Specifies whether to cut the bubbles when they are beyond the area formed by the X axis and Y axis. Data type: Boolean
For stock chart	
Bar Width	Specifies the width of the stock bars. Applies to Open-High-Low-Close stock charts only. Enter a numeric value to change the width. Data type: Integer
Line Thickness	Specifies the thickness of the line. Enter a numeric value to change the thickness. Data type: Float
Stock Down Bar Color	Specifies the color to indicate that the opening price is higher than the closing price. Applies to Open-High-Low-Close stock charts only. Choose a color from the drop-down list, enter a hexadecimal RGB value (for example, 0xff0000), choose/create a formula or edit an expression that returns a color, or select Custom from the drop-down list to specify the color. Data type: String
Stock Line Color	Specifies the color of the stock line. Choose a color from the drop-down list, enter a hexadecimal RGB value (for example, 0xff0000), choose/create a formula or edit an expression that returns a color, or select Custom from the drop-down list to specify the color. Data type: String
Stock Up Bar Color	Specifies the color to indicate that the opening price is lower than the closing price. Applies to Open-High-Low-Close stock charts only. Choose a color from the drop-down list, enter a hexadecimal RGB value (for example, 0xff0000), choose/create a formula or edit an expression that returns a color, or select Custom from the drop-down list to specify the color. Data type: String
For bullet chart	
Comparative Measure Color List	Specifies the fill effect of the comparative measures in the same data series. Click ... in the value cell to specify the effect. Data type: String
Comparative Measure Width	Specifies the width of the comparative measures. Enter a numeric value to change the width. Data type: Integer
Featured Measure Width	Specifies the width of the featured measures. Enter a numeric value to change the width. Data type: Integer
Graph Direction	Specifies the direction to draw the bullet chart, horizontally or vertically. Choose an option from the drop-down list. Data type: Enumeration
Qualitative Ranges Color List	Specifies the fill effect of the qualitative ranges in the same data series. Click ... in the value cell to specify the effect. Data type: String
Qualitative Ranges Width	Specifies the width of the qualitative ranges. Enter a numeric value to change the width. Data type: Integer
Special Featured Measure Graph Type	Specifies the graph type for the featured measures when the value (Y) axis does not start from the value zero. Choose an option from the drop-down list. Data type: Enumeration
Vary Colors by Color List	Specifies whether or not to make colors for the featured measures vary. Data type: Boolean
Hint (unavailable to org charts)	
Color	Specifies the color of the hint text, applied when the property Fill Type is set to color. Choose a color from the drop-down list, enter a hexadecimal RGB value (for example, 0xff0000), choose/create a formula or edit an expression that returns a color, or select Custom from the drop-down list to specify the color. Data type: String
Fill Transparency	Specifies the transparency of the background, in percent. Enter a numeric value to change the transparency. Data type: Integer
Fill Type	Specifies the fill pattern for the hint text. Choose an option from the drop-down list. Data type: Enumeration
Font Effect	Specifies the special effect for the hint. Choose an option from the drop-down list. Data type: Integer
Font Face	Specifies the font face for the hint. Choose an option from the drop-down list. Data type: Enumeration
Font Rotation	Specifies the rotation angle of the hint around its center, in degrees. The default value is 0. Enter a numeric value to change the rotation. Data type: Float
Font Script	Specifies whether the hint will be in superscript or subscript form, or neither of them. Choose an option from the drop-down list. Data type: Enumeration
Font Shearing	Specifies the shearing transformation of the hint around its center. The default value is 0. Enter a numeric value to change the shearing. Data type: Float
Font Size	Specifies the font size for the hint. Enter an integer value to change the size. Data type: Integer
Font Strikethrough	Specifies the style for the line by which the hint text is struck through. Choose an option from the drop-down list. Data type: Enumeration
Font Style	Specifies the font style for the hint. Choose an option from the drop-down list. Data type: Enumeration
Font Underline	Specifies the underline style for the hint. Choose an option from the drop-down list. Data type: Enumeration
Gradient End Color	Specifies the color of the point where the gradient ends, applied when the property Fill Type is set to gradient. Choose a color from the drop-down list, enter a hexadecimal RGB value (for example, 0xff0000), choose/create a formula or edit an expression that returns a color, or select Custom from the drop-down list to specify the color. Data type: String
Gradient End X	Specifies the horizontal position where the gradient ends, measured in a percentage of the hint width. Applied when the property Fill Type is set to gradient. Data type: Integer
Gradient End Y	Specifies the vertical position where the gradient ends, measured in a percentage of the hint height. Applied when the property Fill Type is set to gradient. Data type: Integer
Gradient Start Color	Specifies the color of the point where the gradient begins, applied when the property Fill Type is set to gradient. Choose a color from the drop-down list, enter a hexadecimal RGB value (for example, 0xff0000), choose/create a formula or edit an expression that returns a color, or select Custom from the drop-down list to specify the color. Data type: String
Gradient Start X	Specifies the horizontal position where the gradient begins, measured in a percentage of the hint width. Applied when the property Fill Type is set to gradient. Data type: Integer
Gradient Start Y	Specifies the vertical position where the gradient begins, measured in a percentage of the hint height. Applied when the property Fill Type is set to gradient. Data type: Integer
Gradient Style	Specifies the gradient style for the hint text, applied when the property Fill Type is set to gradient. Choose an option from the drop-down list. Data type: Enumeration
Line Color	Specifies the color of the thin lines that are used to point to the static data labels. Applies to pie charts only. Choose a color from the drop-down list, enter a hexadecimal RGB value (for example, 0xff0000), choose/create a formula or edit an expression that returns a color, or select Custom from the drop-down list to specify the color. Data type: String
Outline Color	Specifies the color for the hint text outline. Choose a color from the drop-down list, enter a hexadecimal RGB value (for example, 0xff0000), choose/create a formula or edit an expression that returns a color, or select Custom from the drop-down list to specify the color. Data type: String
Outline End Caps	Specifies the ending style for the hint text outline. Choose an option from the drop-down list. Data type: Enumeration
Outline Joint	Specifies the joint style for the hint text outline. Choose an option from the drop-down list. Data type: Enumeration
Outline Outlined	Specifies whether to show the hint text outline in outline form. If the property is set to true, the outline will be shown in outline form; otherwise, in whole form. Data type: Boolean
Outline Style	Specifies the style for the hint text outline. Choose an option from the drop-down list. Data type: Enumeration
Outline Thickness	Specifies the thickness of the hint text outline. Enter a numeric value to change the thickness. Data type: Float
Outline Transparency	Specifies the transparency for the hint text outline, in percent. Enter a numeric value to change the transparency. Data type: Integer
Outline Variable Dash	Specifies whether to resize the dash automatically. If the property is set to true, the dash size will be adjusted automatically; otherwise, the dash size will be of fixed size. Data type: Boolean
Show Category and Series	Specifies whether to include the category and series values in the hints of the data markers. Data type: Boolean
Show Fields	Specifies the fields you want to show in the hint. Available to heat map only. Choose an option from the drop-down list: • **Auto** The fields shown in the hint are all the groups, size-by fields, and color-by fields. • **Customized** Opens the Show Fields dialog to customize the fields shown in the hint flexibly. The available fields are the same as the fields which can be inserted into the innermost rectangle. Data type: String
Show Tips	Specifies whether to show the hint that displays the value of a data marker when the mouse pointer points at the data marker in Designer view mode, in HTML result, or at runtime on JReport Server. If this property is set to false, all the other hint properties will be ignored. Data type: Boolean
Texture Background Color	Specifies the background color of the texture, applied when the property Fill Type is set to texture. Choose a color from the drop-down list, enter a hexadecimal RGB value (for example, 0xff0000), choose/create a formula or edit an expression that returns a color, or select Custom from the drop-down list to specify the color. Data type: String
Texture Foreground Color	Specifies the foreground color of the texture, applied when the property Fill Type is set to texture. Choose a color from the drop-down list, enter a hexadecimal RGB value (for example, 0xff0000), choose/create a formula or edit an expression that returns a color, or select Custom from the drop-down list to specify the color. Data type: String
Texture Style	Specifies the texture style for the hint text, applied when the property Fill Type is set to texture. Choose an option from the drop-down list. Data type: Enumeration
Category (X) Axis (unavailable to heat map and org charts)	
Axis Color	Specifies the color for the axis. Choose a color from the drop-down list, enter a hexadecimal RGB value (for example, 0xff0000), choose/create a formula or edit an expression that returns a color, or select Custom from the drop-down list to specify the color. Data type: String
Axis End Caps	Specifies the ending style for the axis. Choose an option from the drop-down list. Data type: Enumeration
Axis Joint	Specifies the joint style for the axis. Choose an option from the drop-down list. Data type: Enumeration
Axis Outlined	Specifies whether to show the axis in outline form. If the property is set to true, the axis will be shown in outline form; otherwise, in whole form. Data type: Boolean
Axis Placement	Specifies the position of X axis across the Y axis. Choose an option from the drop-down list. • **origin** - The X axis will cross the Y axis at the origin. • **bottom** - The X axis will cross the Y axis at its minimum value point. Data type: Enumeration
Axis Style	Specifies the style for the axis. Choose an option from the drop-down list. Data type: Enumeration
Axis Thickness	Specifies the thickness for the axis. Enter a numeric value to change the thickness. Data type: Float
Axis Transparency	Specifies the transparency for the axis, in percent. Enter a numeric value to change the transparency. Data type: Integer
Axis Variable Dash	Specifies whether to resize the dash automatically. If the property is set to true, the dash size will be adjusted automatically; otherwise, the dash size will be of fixed size. Data type: Boolean
Gridline Color	Specifies the color for the gridlines. Choose a color from the drop-down list, enter a hexadecimal RGB value (for example, 0xff0000), choose/create a formula or edit an expression that returns a color, or select Custom from the drop-down list to specify the color. Data type: String
Gridline End Caps	Specifies the ending style for the gridlines. Choose an option from the drop-down list. Data type: Enumeration
Gridline Joint	Specifies the joint style for the gridlines. Choose an option from the drop-down list. Data type: Enumeration
Gridline Outlined	Specifies whether to show the gridlines in outline form. If the property is set to true, the gridlines will be shown in outline form; otherwise, in whole form. Data type: Boolean
Gridline Style	Specifies the style for the gridlines. Choose an option from the drop-down list. Data type: Integer
Gridline Thickness	Specifies the thickness for the gridlines. Enter a numeric value to change the thickness. Data type: Float
Gridline Transparency	Specifies the transparency for the gridlines, in percent. Enter a numeric value to change the transparency. Data type: Integer
Gridline Variable Dash	Specifies whether to resize the dash automatically. If the property is set to true, the dash size will be adjusted automatically; otherwise, the dash size will be of fixed size. Data type: Boolean
Increment	Specifies the increased amount between two adjacent values on the axis. The default is to let the increased amount be determined by JReport Designer. Applies only to bubble charts which use the axis to show numeric data and scatter charts. Data type: Integer
Label Axis Gap	Specifies the distance between the labels and the axis. Enter a numeric value to change the gap. Data type: Float
Label Color	Specifies the color for the label text. Choose a color from the drop-down list, enter a hexadecimal RGB value (for example, 0xff0000), choose/create a formula or edit an expression that returns a color, or select Custom from the drop-down list to specify the color. Data type: String
Label Every N Tick Marks	Specifies the frequency at which the tick marks will be labeled. Type 1 to label every tick mark, type 2 to label every two tick marks, type 3 to label every three tick marks, and so on. Data type: Integer
Label Fill Transparency	Specifies the transparency for the label text, in percent. Enter a numeric value to change the transparency. Data type: Integer
Label Fill Type	Specifies the fill pattern for the label text. Choose an option from the drop-down list. Data type: Enumeration
Label Font Automatic Orientation	Specifies whether to adjust the rotation angle of the label text on the axis automatically. • **true** - The rotation angle of the label text will be adjusted automatically according to the length of the text, in degrees. If the text can be completely displayed horizontally, the default rotation angle will be 0; if the text cannot be completely displayed horizontally, the default rotation angle will be 30 anticlockwise, and the cut off part will be shown as suspension points. • **false** - If selected, the rotation angle of the label text will be determined by value of the property Label Font Rotation. Data type: Boolean **Note:** The property Label Font Automatic Orientation takes higher priority over the property Label Word Wrap.
Label Font Effect	Specifies the special effect for the label text. Choose an option from the drop-down list. Data type: Enumeration
Label Font Face	Specifies the font face for the label text. Choose an option from the drop-down list. Data type: Enumeration
Label Font Rotation	Specifies the rotation angle of the label text around its center, in degrees. The default value is 0. Enter a numeric value to change the rotation. Data type: Float
Label Font Script	Specifies whether the label text will be in superscript or subscript form, or neither of them. Choose an option from the drop-down list. Data type: Enumeration
Label Font Shearing	Specifies the shearing transformation of the label text around its center. The default value is 0. Enter a numeric value to change the shearing. Data type: Float
Label Font Size	Specifies the font size for the label text. Enter an integer value to change the size. Data type: Integer
Label Font Strikethrough	Specifies the style for the line by which the label text is struck through. Choose an option from the drop-down list. Data type: Enumeration
Label Font Style	Specifies the font style for the label text. Choose an option from the drop-down list. Data type: Enumeration
Label Font Underline	Specifies the underline style for the label text. Choose an option from the drop-down list. Data type: Enumeration
Label Gradient End Color	Specifies the color of the point where the gradient ends, applied when the property Label Fill Type is set to gradient. Choose a color from the drop-down list, enter a hexadecimal RGB value (for example, 0xff0000), choose/create a formula or edit an expression that returns a color, or select Custom from the drop-down list to specify the color. Data type: String
Label Gradient End X	Specifies the horizontal position where the gradient ends, measured in a percentage of the label text width. Applied when the property Label Fill Type is set to gradient. Data type: Integer
Label Gradient End Y	Specifies the vertical position of the gradient endpoint, measured in a percentage of the label text height. Applied when the property Label Fill Type is set to gradient. Data type: Integer
Label Gradient Start Color	Specifies the color of the point where the gradient begins, applied when the property Label Fill Type is set to gradient. Choose a color from the drop-down list, enter a hexadecimal RGB value (for example, 0xff0000), choose/create a formula or edit an expression that returns a color, or select Custom from the drop-down list to specify the color. Data type: String
Label Gradient Start X	Specifies the horizontal position where the gradient begins, measured in a percentage of the label text width. Applied when the property Label Fill Type is set to gradient. Data type: Integer
Label Gradient Start Y	Specifies the vertical position where the gradient begins, measured in a percentage of the label text height. Applied when the property Label Fill Type is set to gradient. Data type: Integer
Label Gradient Style	Specifies the gradient style for the label text, applied when the property Label Fill Type is set to gradient. Choose an option from the drop-down list. Data type: Enumeration
Label Outline Color	Specifies the color for the label text outline. Choose a color from the drop-down list, enter a hexadecimal RGB value (for example, 0xff0000), choose/create a formula or edit an expression that returns a color, or select Custom from the drop-down list to specify the color. Data type: String
Label Outline End Caps	Specifies the ending style for the label text outline. Choose an option from the drop-down list. Data type: Enumeration
Label Outline Joint	Specifies the joint style for the label text outline. Choose an option from the drop-down list. Data type: Enumeration
Label Outline Outlined	Specifies whether to show the label text outline in outline form. If the property is set to true, the outline will be shown in outline form; otherwise, in whole form. Data type: Boolean
Label Outline Style	Specifies the style for the label text outline. Choose an option from the drop-down list. Data type: Enumeration
Label Outline Thickness	Specifies the thickness for the label text outline. Enter a numeric value to change the thickness. Data type: Float
Label Outline Transparency	Specifies the transparency for the label text outline, in percent. Enter a numeric value to change the transparency. Data type: Integer
Label Outline Variable Dash	Specifies whether to resize the dash automatically. If the property is set to true, the dash size will be adjusted automatically; otherwise, the dash size will be of fixed size. Data type: Boolean
Label Position	Specifies whether to show the label text outside or inside the axis. Applies to 2-D chart only, and takes effect only when Use Depth is set to false. Data type: Enumeration
Label Texture Background Color	Specifies the background color of the texture, applied when the property Label Fill Type is set to texture. Choose a color from the drop-down list, enter a hexadecimal RGB value (for example, 0xff0000), choose/create a formula or edit an expression that returns a color, or select Custom from the drop-down list to specify the color. Data type: String
Label Texture Foreground Color	Specifies the foreground color of the texture, applied when the property Label Fill Type is set to texture. Choose a color from the drop-down list, enter a hexadecimal RGB value (for example, 0xff0000), choose/create a formula or edit an expression that returns a color, or select Custom from the drop-down list to specify the color. Data type: String
Label Texture Style	Specifies the texture style for the label text, applied when the property Label Fill Type is set to texture. Choose an option from the drop-down list. Data type: Enumeration
Label Word Wrap	Specifies whether to enable the word wrap function for the label text, not applying to radar charts, bubble charts, scatter charts and gauge charts. If the text contains special characters (such as ",", "-" and space), it will be broken at the position of one of the special characters. Data type: Boolean
Major Tick Mark Length	Specifies the length of the major tick marks. Enter a numeric value to change the length. Data type: Float
Major Tick Mark Line Color	Specifies the color of the major tick marks. Choose a color from the drop-down list, enter a hexadecimal RGB value (for example, 0xff0000), choose/create a formula or edit an expression that returns a color, or select Custom from the drop-down list to specify the color. Data type: String
Major Tick Mark Line End Caps	Specifies the ending style for the major tick marks. Choose an option from the drop-down list. Data type: Enumeration
Major Tick Mark Line Joint	Specifies the joint style for the major tick marks. Choose an option from the drop-down list. Data type: Enumeration
Major Tick Mark Line Outlined	Specifies whether to show the major tick marks in outline form. If the property is set to true, the tick marks will be shown in outline form; otherwise, in whole form. Data type: Boolean
Major Tick Mark Line Style	Specifies the line type for the major tick marks. Choose an option from the drop-down list. Data type: Enumeration
Major Tick Mark Line Thickness	Specifies the thickness of the major tick marks. Enter a numeric value to change the thickness. Data type: Float

Major Tick Mark Line Transparency	Specifies the transparency of the major tick marks, in percent. Enter a numeric value to change the transparency. Data type: Integer
Major Tick Mark Line Variable Dash	Specifies whether to adjust the dash size automatically. Applied when Major Tick Mark Line Style is not set to solid line. • **true** - The dash size will be adjusted automatically. • **false** - The dash size will be of fixed size. Data type: Boolean
Major Tick Mark Type	Specifies the position of the major tick marks relative to the axis. Choose an option from the drop-down list. Data type: Enumeration
Maximum Value	Specifies the maximum value of the data appearing on the X axis. Applies only to bubble charts which use the axis to show numeric data and scatter charts. Data type: Float
Minimum Value	Specifies the minimum value of the data appearing on the X axis. Applies only to bubble charts which use the axis to show numeric data and scatter charts. Data type: Float
Minor Tick Mark Length	Specifies the length of the minor tick marks. Enter a numeric value to change the length. Data type: Float
Minor Tick Mark Type	Specifies the position of the minor tick marks relative to the axis. Choose an option from the drop-down list. Data type: Enumeration
Number of Labels	Specifies the number of tick mark labels that will be shown. The default is to let JReport Designer determine the number of labels to be shown. Data type: Integer
Number of Tick Marks	Specifies the number of tick marks that will be shown. The default is to let the number be determined by JReport Designer. Applies to scatter charts and bubble charts only. Data type: Integer
Position Axis	Specifies the position at which the data point and label will be displayed on the X axis. • **on tick marks** - The data point and label will be aligned with each tick mark on the X axis. • **between tick marks** - The data point and label will be displayed between each two tick marks on the X axis. Data type: Enumeration
Show Axis	Specifies whether to show the axis. Data type: Boolean
Show Axis Label Tips	Specifies whether to show the complete label text when the mouse pointer points at a label on the axis in Designer view mode, in HTML result, or at runtime on JReport Server. Data type: Boolean
Show Gridline	Specifies whether to show the gridlines perpendicular to the axis. Data type: Boolean
Show Tick Mark Labels	Specifies whether to show the tick mark labels. Data type: Boolean
Start Value	Specifies the position where the X axis label starts to be shown, applied when the property Use Start Value is set to true. Applies to 2-D charts that have a wall only. Data type: Float
Use Best Effect	Specifies whether to hide the label which would overlap with the one in front of it. Data type: Boolean
Use Start Value	Specifies whether to set the start value of X axis. This property applies to scatter and bubble charts, and only if the value on the category axis is numeric. Data type: Boolean
Value (Y) Axis (unavailable to heat map and org charts)	
Axis Color	Specifies the color for the axis. Choose a color from the drop-down list, enter a hexadecimal RGB value (for example, 0xff0000), choose/create a formula or edit an expression that returns a color, or select Custom from the drop-down list to specify the color. Data type: String
Axis End Caps	Specifies the ending style for the axis. Choose an option from the drop-down list. Data type: Enumeration
Axis Joint	Specifies the joint style for the axis. Choose an option from the drop-down list. Data type: Enumeration
Axis Outlined	Specifies whether to show the axis in outline form. If the property is set to true, the axis will be shown in outline form; otherwise, in whole form. Data type: Boolean
Axis Placement	Specifies the position of the Y axis across the X axis. Choose an option from the drop-down list. • **origin** - The Y axis will cross the X axis at the origin. • **bottom** - The Y axis will cross the X axis at its minimum value point. Data type: Enumeration
Axis Style	Specifies the style for the axis. Choose an option from the drop-down list. Data type: Enumeration
Axis Thickness	Specifies the thickness for the axis. Enter a numeric value to change the thickness. Data type: Float
Axis Transparency	Specifies the transparency for the axis, in percent. Enter a numeric value to change the transparency. Data type: Integer
Axis Variable Dash	Specifies whether to resize the dash automatically. If the property is set to true, the dash size will be adjusted automatically; otherwise, the dash size will be of fixed size. Data type: Boolean
Gridline Color	Specifies the color for the gridlines. Choose a color from the drop-down list, enter a hexadecimal RGB value (for example, 0xff0000), choose/create a formula or edit an expression that returns a color, or select Custom from the drop-down list to specify the color. Data type: String
Gridline End Caps	Specifies the ending style for the gridlines. Choose an option from the drop-down list. Data type: Enumeration
Gridline Joint	Specifies the joint style for the gridlines. Choose an option from the drop-down list. Data type: Enumeration
Gridline Outlined	Specifies whether to show the gridlines in outline form. If the property is set to true, the gridlines will be shown in outline form; otherwise, in whole form. Data type: Boolean
Gridline Style	Specifies the style for the gridlines. Choose an option from the drop-down list. Data type: Enumeration
Gridline Thickness	Specifies the thickness for the gridlines. Enter a numeric value to change the thickness. Data type: Float
Gridline Transparency	Specifies the transparency for the gridlines, in percent. Enter a numeric value to change the transparency. Data type: Integer
Gridline Variable Dash	Specifies whether to resize the dash automatically. If the property is set to true, the dash size will be adjusted automatically; otherwise, the dash size will be of fixed size. Data type: Boolean
Increment	Specifies the increased amount between two adjacent values on the axis. The default is to let the increased amount be determined by JReport Designer. Data type: Integer
Label Axis Gap	Specifies the distance between the labels and the axis. Enter a numeric value to change the gap. Data type: Float
Label Color	Specifies the color for the label text. Choose a color from the drop-down list, enter a hexadecimal RGB value (for example, 0xff0000), choose/create a formula or edit an expression that returns a color, or select Custom from the drop-down list to specify the color. Data type: String
Label Every N Tick Marks	Specifies the frequency at which the tick marks will be labeled. Type 1 to label every tick mark, type 2 to label every two tick marks, type 3 to label every three tick marks, and so on. Data type: Integer
Label Fill Transparency	Specifies the transparency for the label text, in percent. Enter a numeric value to change the transparency. Data type: Integer
Label Fill Type	Specifies the fill pattern for the label text. Choose an option from the drop-down list. Data type: Enumeration
Label Font Automatic Orientation	Specifies whether to adjust the rotation angle of the label text on the axis automatically. • **true** - The rotation angle of the label text will be adjusted automatically according to the length of the text, in degrees. If the text can be completely displayed horizontally, the default rotation angle will be 0; if the text cannot be completely displayed horizontally, the default rotation angle will be 30 anticlockwise, and the cut off part will be shown as suspension points. • **false** - If selected, the rotation angle of the label text will be determined by value of the property Label Font Rotation. Data type: Boolean **Note:** The property Label Font Automatic Orientation takes higher priority over the property Label Word Wrap.
Label Font Effect	Specifies the special effect for the label text. Choose an option from the drop-down list. Data type: Enumeration
Label Font Face	Specifies the font face for the label text. Choose an option from the drop-down list. Data type: Enumeration
Label Font Rotation	Specifies the rotation angle of the label text around its center, in degrees. The default value is 0. Enter a numeric value to change the rotation. Data type: Float
Label Font Script	Specifies whether the label text will in superscript or subscript form, or neither of them. Choose an option from the drop-down list. Data type: Enumeration
Label Font Shearing	Specifies the shearing transformation of the label text around its center. The default value is 0. Enter a numeric value to change the shearing. Data type: Float
Label Font Size	Specifies the font size for the label text. Enter an integer value to change the size. Data type: Integer
Label Font Strikethrough	Specifies the style for the line by which the label text is struck through. Choose an option from the drop-down list. Data type: Enumeration
Label Font Style	Specifies the font style for the label text. Choose an option from the drop-down list. Data type: Enumeration
Label Font Underline	Specifies the underline style for the label text. Choose an option from the drop-down list. Data type: Enumeration
Label Gradient End Color	Specifies the color of the point where the gradient ends, applied when the property Label Fill Type is set to gradient. Choose a color from the drop-down list, enter a hexadecimal RGB value (for example, 0xff0000), choose/create a formula or edit an expression that returns a color, or select Custom from the drop-down list to specify the color. Data type: String
Label Gradient End X	Specifies the horizontal position where the gradient ends, measured in a percentage of the label text width. Applied when the property Label Fill Type is set to gradient. Data type: Integer
Label Gradient End Y	Specifies the vertical position of the gradient endpoint, measured in a percentage of the label text height. Applied when the property Label Fill Type is set to gradient. Data type: Integer
Label Gradient Start Color	Specifies the color of the point where the gradient begins, applied when the property Label Fill Type is set to gradient. Choose a color from the drop-down list, enter a hexadecimal RGB value (for example, 0xff0000), choose/create a formula or edit an expression that returns a color, or select Custom from the drop-down list to specify the color. Data type: String
Label Gradient Start X	Specifies the horizontal position where the gradient begins, measured in a percentage of the label text width. Applied when the property Label Fill Type is set to gradient. Data type: Integer
Label Gradient Start Y	Specifies the vertical position where the gradient begins, measured in a percentage of the label text height. Applied when the property Label Fill Type is set to gradient. Data type: Integer
Label Gradient Style	Specifies the gradient style for the label text, applied when the property Label Fill Type is set to gradient. Choose an option from the drop-down list. Data type: Integer
Label Outline Color	Specifies the color for the label text outline. Choose a color from the drop-down list, enter a hexadecimal RGB value (for example, 0xff0000), choose/create a formula or edit an expression that returns a color, or select Custom from the drop-down list to specify the color. Data type: String
Label Outline End Caps	Specifies the ending style for the label text outline. Choose an option from the drop-down list. Data type: Enumeration
Label Outline Joint	Specifies the joint style for the label text outline. Choose an option from the drop-down list. Data type: Enumeration
Label Outline Outlined	Specifies whether to show the label text outline in outline form. If the property is set to true, the outline will be shown in outline form; otherwise, in whole form. Data type: Boolean
Label Outline Style	Specifies the style for the label text outline. Choose an option from the drop-down list. Data type: Enumeration
Label Outline Thickness	Specifies the thickness for the label text outline. Enter a numeric value to change the thickness. Data type: Float
Label Outline Transparency	Specifies the transparency for the label text outline, in percent. Enter a numeric value to change the transparency. Data type: Integer
Label Outline Variable Dash	Specifies whether to resize the dash automatically. If the property is set to true, the dash size will be adjusted automatically; otherwise, the dash size will be of fixed size. Data type: Boolean
Label Position	Specifies whether to show the label text outside or inside the axis. Applies to 2-D chart only, and takes effect only when the property Use Depth is set to false. Data type: Enumeration
Label Texture Background Color	Specifies the background color of the texture, applied when the property Label Fill Type is set to texture. Choose a color from the drop-down list, enter a hexadecimal RGB value (for example, 0xff0000), choose/create a formula or edit an expression that returns a color, or select Custom from the drop-down list to specify the color. Data type: String
Label Texture Foreground Color	Specifies the foreground color of the texture, applied when the property Label Fill Type is set to texture. Choose a color from the drop-down list, enter a hexadecimal RGB value (for example, 0xff0000), choose/create a formula or edit an expression that returns a color, or select Custom from the drop-down list to specify the color. Data type: String
Label Texture Style	Specifies the texture style for the label text, applied when the property Label Fill Type is set to texture. Choose an option from the drop-down list. Data type: Enumeration
Label Word Wrap	Specifies whether to enable the word wrap function for the label text, not applying to radar charts, bubble charts, scatter charts and gauge charts. If the text contains special characters (such as ",", "." and space), it will be broken at the position of one of the special characters. Data type: Boolean
Major Tick Mark Length	Specifies the length of the major tick marks. Enter a numeric value to change the length. Data type: Float
Major Tick Mark Line Color	Specifies the color of the major tick marks. Choose a color from the drop-down list, enter a hexadecimal RGB value (for example, 0xff0000), choose/create a formula or edit an expression that returns a color, or select Custom from the drop-down list to specify the color. Data type: String
Major Tick Mark Line End Caps	Specifies the ending style for the major tick marks. Choose an option from the drop-down list. Data type: Enumeration
Major Tick Mark Line Joint	Specifies the joint style for the major tick marks. Choose an option from the drop-down list. Data type: Enumeration
Major Tick Mark Line Outlined	Specifies whether to show the major tick marks in outline form. If the property is set to true, the tick marks will be shown in outline form; otherwise, in whole form. Data type: Boolean
Major Tick Mark Line Style	Specifies the line type for the major tick marks. Choose an option from the drop-down list. Data type: Enumeration
Major Tick Mark Line Thickness	Specifies the thickness of the major tick marks. Enter a numeric value to change the thickness. Data type: Float
Major Tick Mark Line Transparency	Specifies the transparency of the major tick marks, in percent. Enter a numeric value to change the transparency. Data type: Integer
Major Tick Mark Line Variable Dash	Specifies whether to adjust the dash size automatically. Applied when Major Tick Mark Line Style is not set to solid line. • **true** - The dash size will be adjusted automatically. • **false** - The dash size will be of fixed size. Data type: Boolean
Major Tick Mark Type	Specifies the position of the major tick marks relative to the axis. Choose an option from the drop-down list. Data type: Enumeration
Maximum Value	Specifies the maximum value of the data appearing on the Y axis. Data type: Float
Minimum Value	Specifies the minimum value of the data appearing on the Y axis. Data type: Float
Minor Tick Mark Length	Specifies the length of the minor tick marks. Enter a numeric value to change the length. Data type: Float
Minor Tick Mark Type	Specifies the position of the minor tick marks relative to the axis. Choose an option from the drop-down list. Data type: Enumeration
Number of Labels	Specifies the number of tick mark labels that will be shown. The default is to let JReport Designer determine the number of labels to be shown. Data type: Integer
Number of Tick Marks	Specifies the number of tick marks that will be shown. The default is to let the number be determined by JReport Designer. Data type: Integer
Show Axis	Specifies whether to show the axis. Data type: Boolean
Show Axis Label Tips	Specifies whether to show the complete label text when the mouse pointer points at a label on the axis in Designer view mode, in HTML result, or at runtime on JReport Server. Data type: Boolean
Show Gridline	Specifies whether to show the gridlines perpendicular to the axis. Data type: Boolean
Show Percent	Specifies whether to show the value labels on the axis in percent. Only applies to radar, bullet, bar/bench, line, and area chart that are not 100% stacked type. Data type: Boolean
Show Tick Mark Labels	Specifies whether to show the tick mark labels. Data type: Boolean
Start Value	Specifies the position where the axis label starts to be shown, applied when the property Use Start Value is set to true. Applies to 2-D charts that have a wall only. Data type: Float
Use Best Effect	Specifies whether to hide the label which would overlap with the one in front of it. Data type: Boolean
Use Start Value	Specifies whether to set the start value of the axis. Applies to 2-D charts that have a wall only. Data type: Boolean
Series (Z) Axis (available to 3-D bar, bench, line, surface and bubble gauge charts only)	
Axis Color	Specifies the color for the axis. Choose a color from the drop-down list, enter a hexadecimal RGB value (for example, 0xff0000), choose/create a formula or edit an expression that returns a color, or select Custom from the drop-down list to specify the color. Data type: String
Axis End Caps	Specifies the ending style for the axis. Choose an option from the drop-down list. Data type: Enumeration
Axis Joint	Specifies the joint style for the axis. Choose an option from the drop-down list. Data type: Enumeration
Axis Outlined	Specifies whether to show the axis in outline form. If the property is set to true, the axis will be shown in outline form; otherwise, in whole form. Data type: Boolean
Axis Placement	Specifies the position of the Z axis. Choose an option from the drop-down list. Data type: Enumeration
Axis Style	Specifies the style for the axis. Choose an option from the drop-down list. Data type: Enumeration
Axis Thickness	Specifies the thickness for the axis. Enter a numeric value to change the thickness. Data type: Float
Axis Transparency	Specifies the transparency for the axis, in percent. Enter a numeric value to change the transparency. Data type: Integer
Axis Variable Dash	Specifies whether to resize the dash automatically. If the property is set to true, the dash size will be adjusted automatically; otherwise, the dash size will be of fixed size. Data type: Boolean
Gridline Color	Specifies the color of the gridlines. Choose a color from the drop-down list, enter a hexadecimal RGB value (for example, 0xff0000), choose/create a formula or edit an expression that returns a color, or select Custom from the drop-down list to specify the color. Data type: String
Gridline End Caps	Specifies the ending style for the gridlines. Choose an option from the drop-down list. Data type: Enumeration
Gridline Joint	Specifies the joint style for the gridlines. Choose an option from the drop-down list. Data type: Enumeration
Gridline Outlined	Specifies whether to show the gridlines in outline form. If the property is set to true, the gridlines will be shown in outline form; otherwise, in whole form. Data type: Boolean
Gridline Style	Specifies the style for the gridlines. Choose an option from the drop-down list. Data type: Enumeration
Gridline Thickness	Specifies the thickness for the gridlines. Enter a numeric value to change the thickness. Data type: Float
Gridline Transparency	Specifies the gridline transparency, in percent. Enter a numeric value to change the transparency. Data type: Integer
Gridline Variable Dash	Specifies whether to resize the dash automatically. If the property is set to true, the dash size will be adjusted automatically; otherwise, the dash size will be of fixed size. Data type: Boolean
Label Axis Gap	Specifies the distance between the labels and the axis. Enter a numeric value to change the gap. Data type: Float
Label Color	Specifies the color for the label text. Choose a color from the drop-down list, enter a hexadecimal RGB value (for example, 0xff0000), choose/create a formula or edit an expression that returns a color, or select Custom from the drop-down list to specify the color. Data type: String
Label Every N Tick Marks	Specifies the frequency at which the tick marks will be labeled. Type 1 to label every tick mark, type 2 to label every two tick marks, type 3 to label every three tick marks, and so on. Data type: Integer
Label Fill Transparency	Specifies the transparency for the label text, in percent. Enter a numeric value to change the transparency. Data type: Integer
Label Fill Type	Specifies the fill pattern for the label text. Choose an option from the drop-down list. Data type: Enumeration
Label Font Automatic Orientation	Specifies whether to adjust the rotation angle of the label text on the axis automatically. • **true** - The rotation angle of the label text will be adjusted automatically according to the length of the text, in degrees. If the text can be completely displayed horizontally, the default rotation angle will be 0; if the text cannot be completely displayed horizontally, the default rotation angle will be 30 anticlockwise, and the cut off part will be shown as suspension points. • **false** - If selected, the rotation angle of the label text will be determined by value of the property Label Font Rotation. Data type: Boolean **Note:** The property Label Font Automatic Orientation takes higher priority over the property Label Word Wrap.
Label Font Effect	Specifies the special effect for the label text. Choose an option from the drop-down list. Data type: Enumeration
Label Font Face	Specifies the font face for the label text. Choose an option from the drop-down list. Data type: Enumeration
Label Font Rotation	Specifies the rotation angle of the label text around its center, in degrees. The default value is 0. Data type: Float
Label Font Script	Specifies whether the label text will in superscript or subscript form, or neither of them. Choose an option from the drop-down list. Data type: Enumeration
Label Font Shearing	Specifies the shearing transformation of the label text around its center. The default value is 0. Data type: Float
Label Font Size	Specifies the font size for the label text. Enter an integer value to change the size. Data type: Integer
Label Font Strikethrough	Specifies the style for the line by which the label text is struck through. Choose an option from the drop-down list. Data type: Enumeration
Label Font Style	Specifies the font style for the label text. Choose an option from the drop-down list. Data type: Enumeration
Label Font Underline	Specifies the underline style for the label text. Choose an option from the drop-down list. Data type: Enumeration
Label Gradient End Color	Specifies the color of the point where the gradient ends, applied when the property Label Fill Type is set to gradient. Choose a color from the drop-down list, enter a hexadecimal RGB value (for example, 0xff0000), choose/create a formula or edit an expression that returns a color, or select Custom from the drop-down list to specify the color. Data type: String
Label Gradient End X	Specifies the horizontal position where the gradient ends, measured in a percentage of the label text width. Applied when the property Label Fill Type is set to gradient. Data type: Integer
Label Gradient End Y	Specifies the vertical position of the gradient endpoint, measured in a percentage of the label text height. Applied when the property Label Fill Type is set to gradient. Data type: Integer
Label Gradient Start Color	Specifies the color of the point where the gradient begins, applied when the property Label Fill Type is set to gradient. Choose a color from the drop-down list, enter a hexadecimal RGB value (for example, 0xff0000), choose/create a formula or edit an expression that returns a color, or select Custom from the drop-down list to specify the color. Data type: String
Label Gradient Start X	Specifies the horizontal position where the gradient begins, measured in a percentage of the label text width. Applied when the property Label Fill Type is set to gradient. Data type: Integer
Label Gradient Start Y	Specifies the vertical position where the gradient begins, measured in a percentage of the label text height. Applied when the property Label Fill Type is set to gradient. Data type: Integer
Label Gradient Style	Specifies the gradient style for the label text, applied when the property Label Fill Type is set to gradient. Choose an option from the drop-down list. Data type: Enumeration
Label Outline Color	Specifies the color for the label text outline. Choose a color from the drop-down list, enter a hexadecimal RGB value (for example, 0xff0000), choose/create a formula or edit an expression that returns a color, or select Custom from the drop-down list to specify the color. Data type: String
Label Outline End Caps	Specifies the ending style for the label text outline. Choose an option from the drop-down list. Data type: Enumeration
Label Outline Joint	Specifies the joint style for the label text outline. Choose an option from the drop-down list. Data type: Enumeration
Label Outline Outlined	Specifies whether to show the label text outline in outline form. If the property is set to true, the outline will be shown in outline form; otherwise, in whole form. Data type: Boolean
Label Outline Style	Specifies the style for the label text outline. Choose an option from the drop-down list. Data type: Enumeration
Label Outline Thickness	Specifies the thickness for the label text outline. Enter a numeric value to change the thickness. Data type: Float
Label Outline Transparency	Specifies the transparency for the label text outline, in percent. Enter a numeric value to change the transparency. Data type: Integer
Label Outline Variable Dash	Specifies whether to resize the dash automatically. If the property is set to true, the dash size will be adjusted automatically; otherwise, the dash size will be of fixed size. Data type: Boolean
Label Texture Background Color	Specifies the background color of the texture, applied when the property Label Fill Type is set to texture. Choose a color from the drop-down list, enter a hexadecimal RGB value (for example, 0xff0000), choose/create a formula or edit an expression that returns a color, or select Custom from the drop-down list to specify the color. Data type: String
Label Texture Foreground Color	Specifies the foreground color of the texture, applied when the property Label Fill Type is set to texture. Choose a color from the drop-down list, enter a hexadecimal RGB value (for example, 0xff0000), choose/create a formula or edit an expression that returns a color, or select Custom from the drop-down list to specify the color. Data type: String
Label Texture Style	Specifies the texture style for the label text, applied when the property Label Fill Type is set to texture. Choose an option from the drop-down list. Data type: Enumeration
Label Word Wrap	Specifies whether to enable the word wrap function for the label text, not applying to radar charts, bubble charts, scatter charts and gauge charts. If the text contains special characters (such as ",", "." and space), it will be broken at the position of one of the special characters. Data type: Boolean
Major Tick Mark Length	Specifies the length of the major tick marks. Enter a numeric value to change the length. Data type: Float
Major Tick Mark Line Color	Specifies the color of the major tick marks. Choose a color from the drop-down list, enter a hexadecimal RGB value (for example, 0xff0000), choose/create a formula or edit an expression that returns a color, or select Custom from the drop-down list to specify the color. Data type: String
Major Tick Mark Line End Caps	Specifies the ending style for the major tick marks. Choose an option from the drop-down list. Data type: Enumeration
Major Tick Mark Line Joint	Specifies the joint style for the major tick marks. Choose an option from the drop-down list. Data type: Enumeration
Major Tick Mark Line Outlined	Specifies whether to show the major tick marks in outline form. If the property is set to true, the tick marks will be shown in outline form; otherwise, in whole form. Data type: Boolean
Major Tick Mark Line Style	Specifies the line type for the major tick marks. Choose an option from the drop-down list. Data type: Enumeration
Major Tick Mark Line Thickness	Specifies the thickness of the major tick marks. Enter a numeric value to change the thickness. Data type: Float
Major Tick Mark Line Transparency	Specifies the transparency of the major tick marks, in percent. Enter a numeric value to change the transparency. Data type: Integer
Major Tick Mark Line Variable Dash	Specifies whether to adjust the dash size automatically. Applied when Major Tick Mark Line Style is not set to solid line. • **true** - The dash size will be adjusted automatically. • **false** - The dash size will be of fixed size. Data type: Boolean
Major Tick Mark Type	Specifies the position of the major tick marks relative to the axis. Choose an option from the drop-down list. Data type: Enumeration
Minor Tick Mark Length	Specifies the length of the minor tick marks. Enter a numeric value to change the length. Data type: Float
Minor Tick Mark Type	Specifies the position of the minor tick marks relative to the axis. Choose an option from the drop-down list. Data type: Enumeration
Number of Labels	Specifies the number of tick mark labels that will be shown. The default is to let JReport Designer determine the number of labels to be shown. Data type: Integer
Show Axis	Specifies whether to show the axis. Data type: Boolean
Show Axis Label Tips	Specifies whether to show the complete label text when the mouse pointer points at a label on the axis in Designer view mode, in HTML result, or at runtime on JReport Server. Data type: Boolean
Show Gridline	Specifies whether to show the gridlines perpendicular to the axis. Data type: Boolean
Show Tick Mark Labels	Specifies whether to show the major tick mark labels. Data type: Boolean
Use Best Effect	Specifies whether to hide the label which would overlap with the one in front of it. Data type: Boolean
Wall (unavailable to radar, gauge, heat map and org charts)	
Border Color	Specifies the color of the border of the object. Choose a color from the drop-down list, enter a hexadecimal RGB value (for example, 0xCCAA00), choose/create a formula or edit an expression that returns a color, or select Custom from the drop-down list to specify the color. Data type: String
Border End Caps	Specifies the ending style of the wall border line. Choose an option from the drop-down list. Data type: Enumeration
Border Joint	Specifies the joint style of the wall border line. Choose an option from the drop-down list. Data type: Enumeration
Border Outlined	Specifies whether to show the wall border line in Outline Path. If the property is set to true, the wall border line will be shown in Outline Path; otherwise, in Fill Path. Data type: Boolean
Border Style	Specifies the line style of the wall borders. Choose an option from the drop-down list. Data type: Enumeration
Border Thickness	Specifies the width of the border in inches or centimeters. Enter a numeric value to change the thickness. Data type: Float
Border Transparency	Specifies the transparency of the wall borders, in percent. Enter a numeric value to change the transparency. Data type: Integer
Border Variable Dash	Specifies whether to resize the dash automatically. If the property is set to true, the dash size will be adjusted automatically; otherwise, the dash size will be of fixed size. Data type: Boolean
Bound Color	Specifies the color of the bound wall, applied when the property Bound Fill Type is set to color. Choose a color from the drop-down list, enter a hexadecimal RGB value (for example, 0xff0000), choose/create a formula or edit an expression that returns a color, or select Custom from the drop-down list to specify the color. Data type: String
Bound Fill Transparency	Specifies the transparency of the background, in percent. Enter a numeric value to change the transparency. Data type: Integer
Bound Fill Type	Specifies the fill pattern for the bound wall. Choose an option from the drop-down list. Data type: Enumeration
Bound Gradient End Color	Specifies the color of the point where the gradient ends, applied when the property Bound Fill Type is set to gradient. Choose a color from the drop-down list, enter a hexadecimal RGB value (for example, 0xff0000), choose/create a formula or edit an expression that returns a color, or select Custom from the drop-down list to specify the color. Data type: String
Bound Gradient End X	Specifies the horizontal position where the gradient ends, measured in a percentage of the bound wall width. Applied when the property Bound Fill Type is set to gradient. Data type: Integer
Bound Gradient End Y	Specifies the vertical position where the gradient ends, measured in a percentage of the bound wall height. Applied when the property Bound Fill Type is set to gradient. Data type: Integer
Bound Gradient Start Color	Specifies the color of the point where the gradient begins, applied when the property Bound Fill Type is set to gradient. Choose a color from the drop-down list, enter a hexadecimal RGB value (for example, 0xff0000), choose/create a formula or edit an expression that returns a color, or select Custom from the drop-down list to specify the color. Data type: String
Bound Gradient Start X	Specifies the horizontal position where the gradient begins, measured in a percentage of the bound wall width. Applied when the property Bound Fill Type is set to gradient. Data type: Integer
Bound Gradient Start Y	Specifies the vertical position where the gradient begins, measured in a percentage of the bound wall height. Applied when the property Bound Fill Type is set to gradient. Data type: Integer
Bound Gradient Style	Specifies the gradient style for the bound wall, applied when the property Bound Fill Type is set to gradient. Choose an option from the drop-down list. Data type: Enumeration
Bound Texture Background Color	Specifies the background color of the texture, applied when the property Bound Fill Type is set to texture. Choose a color from the drop-down list, enter a hexadecimal RGB value (for example, 0xff0000), choose/create a formula or edit an expression that returns a color, or select Custom from the drop-down list to specify the color. Data type: String
Bound Texture Foreground Color	Specifies the foreground color of the texture, applied when the property Bound Fill Type is set to texture. Choose a color from the drop-down list, enter a hexadecimal RGB value (for example, 0xff0000), choose/create a formula or edit an expression that returns a color, or select Custom from the drop-down list to specify the color. Data type: String
Bound Texture Style	Specifies the texture style for the bound wall, applied when the property Bound Fill Type is set to texture. Choose an option from the drop-down list. Data type: Enumeration
Color	Specifies the color of the wall, applied when the property Fill Type is set to color. Choose a color from the drop-down list, enter a hexadecimal RGB value (for example, 0xff0000), choose/create a formula or edit an expression that returns a color, or select Custom from the drop-down list to specify the color. Data type: String
Fill Transparency	Specifies the transparency of the background, in percent. Enter a numeric value to change the transparency. Data type: Integer
Fill Type	Specifies the fill pattern for the wall. Choose an option from the drop-down list. Data type: Enumeration
Gradient End Color	Specifies the color of the point where the gradient ends, applied when the property Fill Type is set to gradient. Choose a color from the drop-down list, enter a hexadecimal RGB value (for example, 0xff0000), choose/create a formula or edit an expression that returns a color, or select Custom from the drop-down list to specify the color. Data type: String
Gradient End X	Specifies the horizontal position where the gradient ends, measured in a percentage of the wall width. Applied when the property Fill Type is set to gradient. Data type: Integer
Gradient End Y	Specifies the vertical position where the gradient ends, measured in a percentage of the wall height. Applied when the property Fill Type is set to gradient. Data type: Integer
Gradient Start Color	Specifies the color of the point where the gradient begins, applied when the property Fill Type is set to gradient. Choose a color from the drop-down list, enter a hexadecimal RGB value (for example, 0xff0000), choose/create a formula or edit an expression that returns a color, or select Custom from the drop-down list to specify the color. Data type: String
Gradient Start X	Specifies the horizontal position where the gradient begins, measured in a percentage of the wall width. Applied when the property Fill Type is set to gradient. Data type: Integer
Gradient Start Y	Specifies the vertical position where the gradient begins, measured in a percentage of the wall height. Applied when the property Fill Type is set to gradient. Data type: Integer
Gradient Style	Specifies the gradient style for the wall, applied when the property Fill Type is set to gradient. Choose an option from the drop-down list. Data type: Enumeration
Image File	Specifies the file name of the image, a portion of which will be displayed as the wall background. Applied when the property Fill Type is set to image. Enter the file name with suffix in the value cell. If the image is outside of the current catalog, you should include the full path of the image correctly. Data type: String
Image Height	The four properties, Image X, Image Y, Image Width and Image Height, together are used to define a portion of the image (the portion is a rectangle), which will be displayed as the wall background. They take effect when the property Fill Type is set to image. Image Height specifies the height of the image portion, measured in a percentage of the image height. Data type: Integer
Image Layout	Specifies the layout style for the image that will be displayed as the wall background, applied when the property Fill Type is set to image. Choose an option from the drop-down list. Data type: Enumeration
Image Width	The four properties, Image X, Image Y, Image Width and Image Height, together are used to define a portion of the image (the portion is a rectangle), which will be displayed as the wall background. They take effect when the property Fill Type is set to image. Image Width specifies the width of the image portion, measured in a percentage of the image width. Data type: Integer
Image X	The four properties, Image X, Image Y, Image Width and Image Height, together are used to define a portion of the image (the portion is a rectangle), which will be displayed as the wall background. They take effect when the property Fill Type is set to image. Image X specifies the distance between the left border of image and the portion that will contain the pattern, measured in a percentage of the image width. Data type: Integer
Image Y	The four properties, Image X, Image Y, Image Width and Image Height, together are used to define a portion of the image (the portion is a rectangle), which will be displayed as the wall background. They take effect when the property Fill Type is set to image. Image Y specifies the distance between the top border of image and the portion that will contain the pattern, measured in a percentage of the image height. Data type: Integer
Show Wall	Specifies whether to show the wall. Data type: Boolean
Texture Background Color	Specifies the background color of the texture, applied when the property Fill Type is set to texture. Choose a color from the drop-down list, enter a hexadecimal RGB value (for example, 0xff0000), choose/create a formula or edit an expression that returns a color, or select Custom from the drop-down list to specify the color. Data type: String
Texture Foreground Color	Specifies the foreground color of the texture, applied when the property Fill Type is set to texture. Choose a color from the drop-down list, enter a hexadecimal RGB value (for example, 0xff0000), choose/create a formula or edit an expression that returns a color, or select Custom from the drop-down list to specify the color. Data type: String
Texture Style	Specifies the texture style for the wall, applied when the property Fill Type is set to texture. Choose an option from the drop-down list. Data type: Enumeration
Floor (available to 3-D bar, bench, line, area, surface charts only)	
Border Color	Specifies the color of the border of the object. Choose a color from the drop-down list, enter a hexadecimal RGB value (for example, 0xCCAA00), choose/create a formula or edit an expression that returns a color, or select Custom from the drop-down list to specify the color. Data type: String
Border End Caps	Specifies the ending style of the floor border line. Choose an option from the drop-down list. Data type: Enumeration
Border Joint	Specifies the joint style of the floor border line. Choose an option from the drop-down list. Data type: Enumeration
Border Outlined	Specifies whether to show the floor border line in Outline Path. If the property is set to true, the floor border line will be shown in Outline Path; otherwise, in Fill Path. Data type: Boolean
Border Style	Specifies the line style of the floor borders. Choose an option from the drop-down list. Data type: Enumeration
Border Thickness	Specifies the width of the border in inches or centimeters. Enter a numeric value to change the thickness. Data type: Float
Border Transparency	Specifies the transparency of the floor borders, in percent. Enter a numeric value to change the transparency. Data type: Integer
Border Variable Dash	Specifies whether to resize the dash automatically. If the property is set to true, the dash size will be adjusted automatically; otherwise, the dash size will be of fixed size. Data type: Boolean
Color	Specifies the color of the floor, applied when the property Fill Type is set to color. Choose a color from the drop-down list, enter a hexadecimal RGB value (for example, 0xff0000), choose/create a formula or edit an expression that returns a color, or select Custom from the drop-down list to specify the color. Data type: String
Fill Transparency	Specifies the transparency of the background, in percent. Enter a numeric value to change the transparency. Data type: Integer
Fill Type	Specifies the fill pattern for the floor. Choose an option from the drop-down list. Data type: Enumeration
Gradient End Color	Specifies the color of the point where the gradient ends, applied when the property Fill Type is set to gradient. Choose a color from the drop-down list, enter a hexadecimal RGB value (for example, 0xff0000), choose/create a formula or edit an expression that returns a color, or select Custom from the drop-down list to specify the color. Data type: String
Gradient End X	Specifies the horizontal position where the gradient ends, measured in a percentage of the floor width. Applied when the property Fill Type is set to gradient. Data type: Integer
Gradient End Y	Specifies the vertical position where the gradient ends, measured in a percentage of the floor height. Applied when the property Fill Type is set to gradient. Data type: Integer
Gradient Start Color	Specifies the color of the point where the gradient begins, applied when the property Fill Type is set to gradient. Choose a color from the drop-down list, enter a hexadecimal RGB value (for example, 0xff0000), choose/create a formula or edit an expression that returns a color, or select Custom from the drop-down list to specify the color. Data type: Integer
Gradient Start X	Specifies the horizontal position where the gradient begins, measured in a percentage of the floor width. Applied when the property Fill Type is set to gradient. Data type: Integer
Gradient Start Y	Specifies the vertical position where the gradient begins, measured in a percentage of the floor height. Applied when the property Fill Type is set to gradient. Data type: Integer
Gradient Style	Specifies the gradient style for the floor, applied when the property Fill Type is set to gradient. Choose an option from the drop-down list. Data type: Enumeration
Image File	Specifies the file name of the image, a portion of which will be displayed as the floor background. Applied when the property Fill Type is set to image. Enter the file name with suffix in the value cell. If the image is outside of the current catalog, you should include the full path of the image correctly. Data type: String
Image Height	The four properties, Image X, Image Y, Image Width and Image Height, together are used to define a portion of the image (the portion is a rectangle), which will be displayed as the floor background. They take effect when the property Fill Type is set to image. Image Height specifies the height of the image portion, measured in a percentage of the image height. Data type: Integer
Image Layout	Specifies the layout style for the image that will be displayed as the floor background, applied when the property Fill Type is set to image. Choose an option from the drop-down list. Data type: Enumeration
Image Width	The four properties, Image X, Image Y, Image Width and Image Height, together are used to define a portion of the image (the portion is a rectangle), which will be displayed as the floor background. They take effect when the property Fill Type is set to image. Image Width specifies the width of the image portion, measured in a percentage of the image width. Data type: Integer
Image X	The four properties, Image X, Image Y, Image Width and Image Height, together are used to define a portion of the image (the portion is a rectangle), which will be displayed as the floor background. They take effect when the property Fill Type is set to image. Image X specifies the distance between the left border of image and the portion that will contain the pattern, measured in a percentage of the image width. Data type: Integer
Image Y	The four properties, Image X, Image Y, Image Width and Image Height, together are used to define a portion of the image (the portion is a rectangle), which will be displayed as the floor background. They take effect when the property Fill Type is set to image. Image Y specifies the distance between the top border of image and the portion that will contain the pattern, measured in a percentage of the image height. Data type: Integer
Show Floor	Specifies whether to show the floor. Applies to 3-D charts only. Data type: Boolean
Texture Background Color	Specifies the background color of the texture, applied when the property Fill Type is set to texture. Choose a color from the drop-down list, enter a hexadecimal RGB value (for example, 0xff0000), choose/create a formula or edit an expression that returns a color, or select Custom from the drop-down list to specify the color. Data type: String
Texture Foreground Color	Specifies the foreground color of the texture, applied when the property Fill Type is set to texture. Choose a color from the drop-down list, enter a hexadecimal RGB value (for example, 0xff0000), choose/create a formula or edit an expression that returns a color, or select Custom from the drop-down list to specify the color. Data type: String
Texture Style	Specifies the texture style for the floor, applied when the property Fill Type is set to texture. Choose an option from the drop-down list. Data type: Enumeration
Threshold Line (unavailable to radar, gauge, scatter, bubble, stock, heat map and org charts)	
Fill Threshold1	Specifies whether to fill the first threshold area with a semi-transparent color. Applies to 2-D charts (except stock charts) only. Data type: Boolean
Fill Threshold2	Specifies whether to fill the second threshold area with a semi-transparent color. Applies to 2-D charts (except stock charts) only. Data type: Boolean
Fill Transparency	Specifies the transparency of the background, in percent. Enter a numeric value to change the transparency. Data type: Integer
Show Threshold Line1	Specifies whether to show the first threshold line. Applies to 2-D charts (except stock charts), 3-D bar charts, 3-D area charts, 3-D line charts and 3-D bench charts. Data type: Boolean
Show Threshold Line2	Specifies whether to show the second threshold line. Applies to 2-D charts (except stock charts), 3-D bar charts, 3-D area charts, 3-D line charts and 3-D bench charts. Data type: Boolean
Threshold Line Color1	Specifies the color for the first threshold line. Applies to 2-D charts (except stock charts), 3-D bar charts, 3-D area charts, 3-D line charts and 3-D bench charts. Choose a color from the drop-down list, enter a hexadecimal RGB value (for example, 0xff0000), choose/create a formula or edit an expression that returns a color, or select Custom from the drop-down list to specify the color. Data type: String
Threshold Line Color2	Specifies the color for the second threshold line. Applies to 2-D charts (except stock charts), 3-D bar charts, 3-D area charts, 3-D line charts and 3-D bench charts. Choose a color from the drop-down list, enter a hexadecimal RGB value (for example, 0xff0000), choose/create a formula or edit an expression that returns a color, or select Custom from the drop-down list to specify the color. Data type: String
Threshold Line Style	Specifies the style for the threshold lines. Applies to 2-D charts (except stock charts). Choose an option from the drop-down list. Data type: Enumeration
Threshold Value1	Specifies the value for the first threshold line. Applies to 2-D charts (except stock charts), 3-D bar charts, 3-D area charts, 3-D line charts and 3-D bench charts. Data type: Float
Threshold Value2	Specifies the value for the second threshold line. Applies to 2-D charts (except stock charts), 3-D bar charts, 3-D area charts, 3-D line charts and 3-D bench charts. Data type: Float
MeanLine (unavailable to radar, gauge, surface, scatter, bubble, stock, heat map and org charts)	
Color	Specifies the color of the mean line. Choose a color from the drop-down list, enter a hexadecimal RGB value (for example, 0xff0000), choose/create a formula or edit an expression that returns a color, or select Custom from the drop-down list to specify the color. Data type: String
Node Style	Specifies the style of the nodes that represent the average values. Choose an option from the drop-down list. Data type: Enumeration
Show	Specifies whether to show a line that represents the average value of each category group. Applies to 2-D charts (except stock charts, scatter charts, radar charts and bubble charts) only. Data type: Boolean
Type	Specifies the type of the mean line. Choose an option from the drop-down list. Data type: Enumeration

Notes:

- For combo chart, the graph properties will respectively apply to the corresponding chart types.
- A combo chart may contain two value axes, namely, the Y1 and Y2 axes. While a 3-D chart only contains one value axis. The properties of the Y1 or Y2 axis are the same as those of the Y axis.

The following are the properties of the two child objects of the chart paper in an org chart:

- Org chart node
- Org chart line

The following are the properties of the two child objects of the chart paper in a heat map:

- Heat map rectangle
- Heat map rectangle title

Org chart node

The properties of each node in an org chart of a page report are:

Property Name	Description
General	
Class Type	Indicates the class type of the object. This property is read only.
Data Inherit	Indicates whether the dataset for this object is inherited from another object. This property is read only.
Dataset	Indicates the dataset used by the object. This property is read only.
Instance Name	Shows the instance name of the object. This property is read only.
Geometry	
Height	Specifies the height of the object, in inches or centimeters. Enter a numeric value to change the height. Data type: Float
Width	Specifies the width of the object, in inches or centimeters. Enter a numeric value to change the width. Data type: Float
Color	
Background	Specifies the background color of the object. Choose a color from the drop-down list, enter a hexadecimal RGB value (for example, 0xff0000), choose/create a formula or edit an expression that returns a color, or select Custom from the drop-down list to specify the color. Data type: String
CSS	
Class	Specifies a CSS class to be applied to the object which is a valid class in the CSS file. Data type: String
Style	The property can be used in two ways. • Specifies a style to be applied to the object. Choose a style from the drop-down list which is available when a Style Group has been selected at the report tab level and when there are styles in the Style Group that can be applied to the object, or type in the style name. • Specifies a CSS selector to be applied to the object. Enter a valid class name from the CSS file. Data type: String
Others	
Export to CSV	Specifies whether to include the object when exporting the report to CSV. Data type: Boolean
Export to Excel	Specifies whether to include the object when exporting the report to Excel. Data type: Boolean
Export to Page Report Result	Specifies whether to include the object when exporting the report to Page Report Result. Data type: Boolean
Minimum Leaf Distance	Specifies the minimum distance between two adjacent leaf nodes on the same level. For top-down and bottom-up layouts, it means horizontal distance. For left-right layout, it means vertical distance. Enter a numeric value to specify the distance. Data type: Float
Minimum Level Distance	Specifies the minimum distance between two adjacent levels. For top-down and bottom-up layouts, it means vertical distance. For left-right layout, it means horizontal distance. At runtime, if the vlaue of Minimum Level Distance is less than that of Line Thickness on org chart line, the former will be overridden by the latter. Data type: Float
Record Location	Specifies the calculation point for the properties which use formulas. Choose an option from the drop-down list. • **Default** - The values of the properties are calculated in the default location where they are located. • **Page Header** - The values of the properties are calculated in the page header. • **Page Footer** - The values of the properties are calculated in the page footer. Data type: Enumeration
Vertical Alignment	Specifies the vertical justification of the objects in each node. Choose an option from the drop-down list. Data type: Enumeration
Border	
Border Color	Specifies the color of the border of the object. Choose a color from the drop-down list, enter a hexadecimal RGB value (for example, 0xCCAA00), choose/create a formula or edit an expression that returns a color, or select Custom from the drop-down list to specify the color. Data type: String
Border Thickness	Specifies the width of the border in inches or centimeters. Enter a numeric value to change the thickness. Data type: Float
Bottom Line	Specifies the line style of the bottom border of the object. Choose a style from the drop-down list. Data type: Enumeration
Left Line	Specifies the line style of the left border of the object. Choose a style from the drop-down list. Data type: Enumeration
Right Line	Specifies the line style of the right border of the object. Choose a style from the drop-down list. Data type: Enumeration
Top Line	Specifies the line style of the top border of the object. Choose a style from the drop-down list. Data type: Enumeration
Accessibility	
External CSS Class Selector	Specifies a class selector to be applied to the object when exported as HTML. Enter a valid class name from the CSS file. Data type: String

Org chart line

The properties of the line object in an org chart of a page report are:

Property Name	Description
General	
Class Type	Indicates the class type of the object. This property is read only.
Instance Name	Shows the instance name of the object. This property is read only.
CSS	
Class	Specifies a CSS class to be applied to the object which is a valid class in the CSS file. Data type: String
Style	The property can be used in two ways. • Specifies a style to be applied to the object. Choose a style from the drop-down list which is available when a Style Group has been selected at the report tab level and when there are styles in the Style Group that can be applied to the object, or type in the style name. • Specifies a CSS selector to be applied to the object. Enter a valid class name from the CSS file. Data type: String
Line Property	
Line Color	Specifies the color of the line. Choose a color from the drop-down list, enter a hexadecimal RGB value (for example, 0xff0000), choose/create a formula or edit an expression that returns a color, or select Custom from the drop-down list to specify the color and fill effect. Data type: String
Line Style	Specifies the style of the line. Choose an option from the drop-down list. Data type: Enumeration
Line Thickness	Specifies the thickness of the line. Enter a numeric value to change the thickness. Data type: Float

Heat map rectangle

The properties of the rectangles in a heat map of a page report are:

Property	Description
General	
Class Type	Indicates the class type of the object. This property is read only.
Group By	Indicates the group-by field related to the rectangle. This property is read only.
Instance Name	Shows the instance name of the object. This property is read only.
CSS	
Class	Specifies a CSS class to be applied to the object which is a valid class in the CSS file. Data type: String
Style	The property can be used in two ways. • Specifies a style to be applied to the object. Choose a style from the drop-down list which is available when a Style Group has been selected at the report tab level and when there are styles in the Style Group that can be applied to the object, or type in the style name. • Specifies a CSS selector to be applied to the object. Enter a valid class name from the CSS file. Data type: String
Others	
Link	Specifies to link the object to another report, a website, an e-mail address or a Blob data type field. Click ... in the value cell to set the link target. For details, see Binding a link to a label. Data type: String
Vertical Alignment	Specifies the vertical justification of the text in the object. Choose an option from the drop-down list. Data type: Enumeration
Separator	
Color	Specifies the color of the separator line. Choose a color from the drop-down list, enter a hexadecimal RGB value (for example, 0xff0000), choose/create a formula or edit an expression that returns a color, or select Custom from the drop-down list to specify the color. Data type: String
Line Style	Specifies the line style of separator. Choose an option from the drop-down list. Data type: Enumeration
Thickness	Specifies the line width of the separator, in inches or centimeters. Enter a numeric value to change the thickness. Data type: Float
Transparency	Specifies the transparency of separators, in percent. Enter a numeric value to change the transparency. Data type: Integer
TOC	
Anchor Display Value	Specifies a string or formula to display for the TOC entry for this object in the TOC Browser. By default it is the value of the specified object. The TOC Anchor property must be set to true for this property to take effect. Data type: String
TOC Anchor	Specifies whether to include the object in the TOC Browser for the report. Data type: Boolean
Accessibility	
External CSS Class Selector	Specifies a class selector to be applied to the object when exported as HTML. Enter a valid class name from the CSS file. Data type: String

Heat map rectangle title

The properties of the rectangle titles in a heat map of a page report are:

Property Name	Description
General	
Class Type	Indicates the class type of the object. This property is read only.
Instance Name	Shows the instance name of the object. This property is read only.
Text Format	
Bold	Specifies whether to make the text bold. Data type: Boolean
Font Face	Specifies the font of the text. Choose an option from the drop-down list. Data type: Enumeration
Font Size	Specifies the font size of the text. Enter an integer value to change the size. Data type: Integer
Format	Indicates the data format depending on the data type of the group field.
Horizontal Alignment	Specifies the horizontal justification of the text in the object. Choose an option from the drop-down list. Data type: Enumeration
Ignore HTML Tag	Specifies whether or not to ignore the HTML tags when exporting the report results to HTML format. • **true** - The text will appear in the HTML file the same as that in JReport Designer (HTML tag elements in the text, if any, will not be parsed). • **false** - HTML tag elements in the text will be parsed when exporting the report results to HTML format. This property is useful for exporting the report results to HTML format. You may want to write the label contents in HTML tags in your report, and for the exported HTML files, the tags will be transferred as they are translated into HTML. Data type: Boolean
Italic	Specifies whether to make the text italic. Data type: Boolean
Maximum Width	Specifies the maximum width of the text displayed. Enter a numeric value to change the width. Data type: Float
Strikethrough	Specifies whether to add a strikethrough line to the text. Data type: Boolean
Underline	Specifies whether to underline the text. Data type: Boolean
Vertical Alignment	Specifies the vertical justification of the text in the object. Choose an option from the drop-down list. Data type: Enumeration
Word Wrap	Specifies whether to wrap the text to the width. Data type: Boolean
Geometry	
Height	Specifies the height of the object, in inches or centimeters. Enter a numeric value to change the height. Enabled when it is a horizontal banded object. Data type: Float
Color	
Background	Specifies the background color of the object. Choose a color from the drop-down list, enter a hexadecimal RGB value (for example, 0xff0000), choose/create a formula or edit an expression that returns a color, or select Custom from the drop-down list to specify the color. Data type: String
Foreground	Specifies the foreground color of the object. Choose a color from the drop-down list, enter a hexadecimal RGB value (for example, 0xff0000), choose/create a formula or edit an expression that returns a color, or select Custom from the drop-down list to specify the color. Data type: String
CSS	
Class	Specifies a CSS class to be applied to the object which is a valid class in the CSS file. Data type: String
ID	Specifies the identifier of the object, which must be unique in the report. The ID property can be a style sheet selector. Data type: String
Style	The property can be used in two ways. • Specifies a style to be applied to the object. Choose a style from the drop-down list which is available when a Style Group has been selected at the report tab level and when there are styles in the Style Group that can be applied to the object, or type in the style name. • Specifies a CSS selector to be applied to the object. Enter a valid class name from the CSS file. Data type: String
Excel	
Column Index	Specifies the X coordinate of the object relative to its parent container when exported to Excel or CSV, measured in cells. The Columned property at the report tab level must be set to true for this property to take effect. Data type: Integer
Row Index	Specifies the Y coordinate of the object relative to its parent container when exported to Excel or CSV, measured in cells. The Columned property at the report tab level must be set to true for this property to take effect. Data type: Integer
Padding	
Bottom Padding	Specifies the space in inches or centimeters between the text and the bottom border of the object. Enter a numeric value to change the padding. Data type: Float
Left Padding	Specifies the space in inches or centimeters between the text and the left border of the object. Enter a numeric value to change the padding. Data type: Float
Right Padding	Specifies the space in inches or centimeters between the text and the right border of the object. Enter a numeric value to change the padding. Data type: Float
Top Padding	Specifies the space in inches or centimeters between the text and the top border of the object. Enter a numeric value to change the padding. Data type: Float
Border	
Border Color	Specifies the color of the border of the object. Choose a color from the drop-down list, enter a hexadecimal RGB value (for example, 0xCCAA00), choose/create a formula or edit an expression that returns a color, or select Custom from the drop-down list to specify the color. Data type: String
Border Thickness	Specifies the width of the border in inches or centimeters. Enter a numeric value to change the thickness. Data type: Float
Bottom Line	Specifies the line style of the bottom border of the object. Choose a style from the drop-down list. Data type: Enumeration
Left Line	Specifies the line style of the left border of the object. Choose a style from the drop-down list. Data type: Enumeration
Right Line	Specifies the line style of the right border of the object. Choose a style from the drop-down list. Data type: Enumeration
Shadow	Specifies whether to draw two shadowed borders, beneath and to the right of the object. Data type: Boolean
Shadow Color	Specifies the color of the shadow. Choose a color from the drop-down list, enter a hexadecimal RGB value (for example, 0xff0000), choose/create a formula or edit an expression that returns a color, or select Custom from the drop-down list to specify the color and fill effect. Data type: String
Top Line	Specifies the line style of the top border of the object. Choose a style from the drop-down list. Data type: Enumeration
Pattern	
Pattern Color	Specifies the color in which to draw a pattern to fill the object. Choose a color from the drop-down list, enter a hexadecimal RGB value (for example, 0xff0000), choose/create a formula or edit an expression that returns a color, or select Custom from the drop-down list to specify the color and fill effect. Data type: String
Pattern Style	Specifies the style of the pattern. Choose an option from the drop-down list. • **none** - No pattern will be applied to the object. • **50%** - The object will be filled in the Pattern Color of 50%-transparency. • **horizontal** - The pattern will be made of horizontal lines of the Pattern Color. • **vertical** - The pattern will be made of vertical lines of the Pattern Color. • **grid** - The pattern will be an overlapping of horizontal and vertical lines of the Pattern Color. • **diagonal** - The pattern will be made of diagonal lines of the Pattern Color. Data type: Enumeration
Others	
Invisible	Specifies whether to show the object in the design area and in the report results. All formulas and calculations will still be performed if the property is set to true. Data type: Boolean
TOC	
Anchor Display Value	Specifies a string or formula to display for the TOC entry for this object in the TOC Browser. By default it is the value of the specified object. The TOC Anchor property must be set to true for this property to take effect. Data type: String
TOC Anchor	Specifies whether to include the object in the TOC Browser for the report. Data type: Boolean
Accessibility	
External AccessKey	It is mapped to the HTML attribute *accesskey*. This attribute specifies an access key to the object. Data type: String
External CSS Class Selector	Specifies a class selector to be applied to the object when exported as HTML. Enter a valid class name from the CSS file. Data type: String
External Dir	It is mapped to the HTML attribute *dir*. This attribute specifies the base direction of directionally neutral text in the object's content and attribute values. It also specifies the directionality of tables. Possible values: • LTR - Left-to-right text or table. • RTL - Right-to-left text or table. Data type: String
External ID	It is mapped to the HTML attribute *id*. This attribute specifies a name to the object, which must be unique in the report. Data type: String
External Style	It is mapped to the HTML attribute *style*. This attribute specifies style information for the object. Data type: String
External TabIndex	It is mapped to the HTML attribute *tabindex*. This attribute specifies the position of the object in the tabbing order for the current report. Enter an integer equal to or greater than 0 and less than 65535. Data type: Integer
External Title	It is mapped to the HTML attribute *title*. This attribute offers advisory information about the object. Data type: String
HrefLang	It is mapped to the HTML attribute *hreflang*. This attribute specifies the base language of the resource designated by a link and may only be used when a link is specified. Data type: String
Language	It is mapped to the HTML attribute *lang*. This attribute specifies the base language of the object's attribute values and text content. Data type: String
LongDesc	It is mapped to the HTML attribute *longdesc*. This attribute specifies a link to a long description of the object. Data type: String

Chart legend

The properties of a chart legend in a page report are:

Property	Description
General	
Class Type	Indicates the class type of the object. This property is read only.
Data Inherit	Indicates whether the dataset for this object is inherited from another object. This property is read only.
Dataset	Indicates the dataset used by the object. This property is read only.
Instance Name	Shows the instance name of the object. This property is read only.
Geometry	
Height	Specifies the height of the object, in inches or centimeters. Enter a numeric value to change the height. Data type: Float
Width	Specifies the width of the object, in inches or centimeters. Enter a numeric value to change the width. Data type: Float
X	Specifies the horizontal coordinate of the top left corner of the object, relative to its parent container. Enter a numeric value to change the position in inches or centimeters. This property takes effect only when the Placement property is set to customized. Data type: Float
Y	Specifies the vertical coordinate of the top left corner of the object, relative to its parent container. Enter a numeric value to change the position in inches or centimeters. This property takes effect only when the Placement property is set to customized. Data type: Float
CSS	
Class	Specifies a CSS class to be applied to the object which is a valid class in the CSS file. Data type: String
Style	The property can be used in two ways. • Specifies a style to be applied to the object. Choose a style from the drop-down list which is available when a Style Group has been selected at the report tab level and when there are styles in the Style Group that can be applied to the object, or type in the style name. • Specifies a CSS selector to be applied to the object. Enter a valid class name from the CSS file. Data type: String
Accessibility	
External CSS Class Selector	Specifies a class selector to be applied to the object when exported as HTML. Enter a valid class name from the CSS file. Data type: String
Legend	
Auto Size	Specifies whether to resize the legend automatically. Data type: Boolean
Bottom Margin	Specifies the distance in inches or centimeters between the legend labels and the bottom border of the legend. Enter a numeric value to change the margin. Data type: Float
Data Mapping File	Specifies the data mapping file to the object for NLS use. For example, if the data mapping file name is Category_de_DE.properties, set the value as Category (without the language and locale part). Data type: String
Label Format Source	Specifies the data format option for the chart legend labels. Choose an option from the drop-down list. • **Axis Data Format** - If selected, the data format of the legend labels will follow that of the axis data: if the chart has series axis, the legend label format will be the same as the series axis data format; if not, the category axis data format will be used to the legend labels. • **Legend Data Format** - If selected, the data format of the legend labels will be controlled by the property Legend Data Format. Unavailable to heat maps that are colored only by summaries. Data type: Enumeration
Label Horizontal Spacing	Specifies the horizontal distance between two adjacent legend labels. Enter a numeric value to change the distance. Unavailable to heat maps that are colored only by summaries. Data type: Float
Label Vertical Spacing	Specifies the vertical distance between two adjacent legend labels. Enter a numeric value to change the distance. Unavailable to heat maps that are colored only by summaries. Data type: Float
Left Margin	Specifies the distance in inches or centimeters between the legend labels and the left border of the legend. Enter a numeric value to change the margin. Data type: Float
Legend Data Format	Specifies the data format of the legend labels, applied when Label Format Source is set to Legend Data Format. Click ... in the value cell to specify the format in the Data Format dialog. Data type: String
Placement	Specifies the position of the legend in the platform. Choose an option from the drop-down list. Data type: Enumeration
Right Margin	Specifies the distance in inches or centimeters between the legend labels and the right border of the legend. Enter a numeric value to change the margin. Data type: Float
Secondary Placement	Specifies the position of the legend further based on the setting of the property Placement. Choose an option from the drop-down list. Data type: Enumeration
Show Scrollbar	Specifies whether to show a scrollbar on the legend to fully view the legend content when the content does not fit into the legend. Data type: Boolean
Show Tips	Specifies whether to show the corresponding data information when the mouse pointer points at a target in the chart legend in Designer view mode, in HTML result, or at server runtime. Data type: Boolean
Top Margin	Specifies the distance in inches or centimeters between the legend labels and the top border of the legend. Enter a numeric value to change the margin. Data type: Float
Truncate	Specifies whether to truncate the legend entry label text when the text overflow the labels. Data type: Boolean
Background	
Border Color	Specifies the color of the legend border. Choose a color from the drop-down list, enter a hexadecimal RGB value (for example, 0xCCAA00), choose/create a formula or edit an expression that returns a color, or select Custom from the drop-down list to specify the color. Data type: String
Border End Caps	Specifies the ending style of the legend border line. Choose an option from the drop-down list. Data type: Enumeration
Border Joint	Specifies the joint style of the legend border line. Choose an option from the drop-down list. Data type: Enumeration
Border Outlined	Specifies whether to show the legend border line in Outline Path. If the property is set to true, the legend border line will be shown in Outline Path; otherwise, in Fill Path. Data type: Boolean
Border Style	Specifies the line style of the legend border. Choose an option from the drop-down list. Data type: Enumeration
Border Thickness	Specifies the width of the legend border, in inches or centimeters. Enter a numeric value to change the thickness. Data type: Float
Border Transparency	Specifies the transparency of the legend border, in percent. Enter a numeric value to change the transparency. Data type: Integer
Border Type	Specifies the border type of the legend. Choose an option from the drop-down list. Data type: Enumeration
Border Variable Dash	Specifies whether to resize the dash automatically. • **true** - If selected, the dash size will be adjusted automatically, and the option Auto Adjust Dash in the Format Legend dialog will take effect. • **false** - If selected, the dash size will be of fixed size, and the option Fixed Dash Size in the Format Legend dialog will take effect. Data type: Boolean
Color	Specifies the color of the legend background, applied when the property Fill Type is set to color. Choose a color from the drop-down list, enter a hexadecimal RGB value (for example, 0xff0000), choose/create a formula or edit an expression that returns a color, or select Custom from the drop-down list to specify the color. Data type: String
Fill Transparency	Specifies the transparency of the legend background, in percent. Enter a numeric value to change the transparency. Data type: Integer
Fill Type	Specifies the fill pattern for the legend background. Choose an option from the drop-down list. Data type: Enumeration
Gradient End Color	Specifies the color of the point where the gradient ends, applied when the property Fill Type is set to gradient. Choose a color from the drop-down list, enter a hexadecimal RGB value (for example, 0xff0000), choose/create a formula or edit an expression that returns a color, or select Custom from the drop-down list to specify the color. Data type: String
Gradient End X	Specifies the horizontal position where the gradient ends, measured in a percentage of the legend width. Applied when the property Fill Type is set to gradient. Data type: Integer
Gradient End Y	Specifies the vertical position where the gradient ends, measured in a percentage of the legend height. Applied when the property Fill Type is set to gradient. Data type: Integer
Gradient Start Color	Specifies the color of the point where the gradient begins, applied when the property Fill Type is set to gradient. Choose a color from the drop-down list, enter a hexadecimal RGB value (for example, 0xff0000), choose/create a formula or edit an expression that returns a color, or select Custom from the drop-down list to specify the color. Data type: String
Gradient Start X	Specifies the horizontal position where the gradient begins, measured in a percentage of the legend width. Applied when the property Fill Type is set to gradient. Data type: Integer
Gradient Start Y	Specifies the vertical position where the gradient begins, measured in a percentage of the legend height. Applied when the property Fill Type is set to gradient. Data type: Integer
Gradient Style	Specifies the gradient style for the legend background, applied when the property Fill Type is set to gradient. Choose an option from the drop-down list. Data type: Enumeration
Image File	Specifies the file name of the image, a portion of which will be displayed as the legend background. Applied when the property Fill Type is set to image. Enter the file name with suffix in the value cell. If the image is outside of the current catalog, you should include the full path of the image correctly. Data type: String
Image Height	The four properties, Image X, Image Y, Image Width, and Image Height, together are used to define a portion of the image (the portion is a rectangle), which will be displayed as the legend background. They take effect when the property Fill Type is set to image. Image Height specifies the height of the image portion, measured in a percentage of the image height. Data type: Integer
Image Layout	Specifies the layout style for the image that will be displayed as the legend background, applied when the property Fill Type is set to image. Choose an option from the drop-down list. Data type: Enumeration
Image Width	The four properties, Image X, Image Y, Image Width, and Image Height, together are used to define a portion of the image (the portion is a rectangle), which will be displayed as the legend background. They take effect when the property Fill Type is set to image. Image Width specifies the width of the image portion, measured in a percentage of the image width. Data type: Integer
Image X	The four properties, Image X, Image Y, Image Width, and Image Height, together are used to define a portion of the image (the portion is a rectangle), which will be displayed as the legend background. They take effect when the property Fill Type is set to image. Image X specifies the distance between the left border of image and the portion that will contain the pattern, measured in a percentage of the image width. Data type: Integer
Image Y	The four properties, Image X, Image Y, Image Width, and Image Height, together are used to define a portion of the image (the portion is a rectangle), which will be displayed as the legend background. They take effect when the property Fill Type is set to image. Image Y specifies the distance between the top border of image and the portion that will contain the pattern, measured in a percentage of the image height. Data type: Integer
Inset Bottom	Specifies the bottom position of the background area, measured in a percentage of the legend height, from the bottom edge of the legend. Data type: Float
Inset Left	Specifies the left position of the background area, measured in a percentage of the legend width, from the left edge of the legend. Data type: Float
Inset Right	Specifies the right position of the background area, measured in a percentage of the legend width, from the right edge of the legend. Data type: Float
Inset Top	Specifies the top position of the background area, measured in a percentage of the legend height, from the top edge of the legend. Data type: Float
Texture Background Color	Specifies the background color of the texture, applied when the property Fill Type is set to texture. Choose a color from the drop-down list, enter a hexadecimal RGB value (for example, 0xff0000), choose/create a formula or edit an expression that returns a color, or select Custom from the drop-down list to specify the color. Data type: String
Texture Foreground Color	Specifies the foreground color of the texture, applied when the property Fill Type is set to texture. Choose a color from the drop-down list, enter a hexadecimal RGB value (for example, 0xff0000), choose/create a formula or edit an expression that returns a color, or select Custom from the drop-down list to specify the color. Data type: String
Texture Style	Specifies the texture style of the legend background, applied when the property Fill Type is set to texture. Choose an option from the drop-down list. Data type: Enumeration
Label	
Color	Specifies the color of the label, applied when the property Fill Type is set to color. Choose a color from the drop-down list, enter a hexadecimal RGB value (for example, 0xff0000), choose/create a formula or edit an expression that returns a color, or select Custom from the drop-down list to specify the color. Data type: String
Fill Transparency	Specifies the transparency of the background, in percent. Enter a numeric value to change the transparency. Data type: Integer
Fill Type	Specifies the fill pattern for the label. Choose an option from the drop-down list. Data type: Enumeration
Font Effect	Specifies the special effect for the label text. Choose an option from the drop-down list. Data type: Enumeration
Font Face	Specifies the font face for the legend labels. Choose an option from the drop-down list. Data type: Enumeration
Font Rotation	Specifies the rotation angle of each legend label around its center, in degrees. The default value is 0. Enter a numeric value to change the rotation. Data type: Float
Font Script	Specifies whether the label text will in superscript or subscript form, or neither of them. Choose an option from the drop-down list. Data type: Enumeration
Font Shearing	Specifies the shearing transformation of each legend label around its center. The default value is 0. Enter a numeric value to change the shearing. Data type: Float
Font Size	Specifies the font size for the legend labels. Enter an integer value to change the size. Data type: Integer
Font Strikethrough	Specifies the style of the line by which the label texts are struck through. Choose an option from the drop-down list. Data type: Enumeration
Font Style	Specifies the font style for the legend labels. Choose an option from the drop-down list. Data type: Enumeration
Font Underline	Specifies the underline style for the legend labels. Choose an option from the drop-down list. Data type: Enumeration
Gradient End Color	Specifies the color of the point where the gradient ends, applied when the property Fill Type is set to gradient. Choose a color from the drop-down list, enter a hexadecimal RGB value (for example, 0xff0000), choose/create a formula or edit an expression that returns a color, or select Custom from the drop-down list to specify the color. Data type: String
Gradient End X	Specifies the horizontal position where the gradient ends, measured in a percentage of the label text width. Applied when the property Fill Type is set to gradient. Data type: Integer
Gradient End Y	Specifies the vertical position where the gradient ends, measured in a percentage of the label text height. Applied when the property Fill Type is set to gradient. Data type: Integer
Gradient Start Color	Specifies the color of the point where the gradient begins, applied when the property Fill Type is set to gradient. Choose a color from the drop-down list, enter a hexadecimal RGB value (for example, 0xff0000), choose/create a formula or edit an expression that returns a color, or select Custom from the drop-down list to specify the color. Data type: String
Gradient Start X	Specifies the horizontal position where the gradient begins, measured in a percentage of the label text width. Applied when the property Fill Type is set to gradient. Data type: Integer
Gradient Start Y	Specifies the vertical position where the gradient begins, measured in a percentage of the label text height. Applied when the property Fill Type is set to gradient. Data type: Integer
Gradient Style	Specifies the gradient style for the label, applied when the property Fill Type is set to gradient. The image fill type does not apply to the legend label. Choose an option from the drop-down list. Data type: Enumeration
Outline Color	Specifies the color for the label text outline. Choose a color from the drop-down list, enter a hexadecimal RGB value (for example, 0xff0000), choose/create a formula or edit an expression that returns a color, or select Custom from the drop-down list to specify the color. Data type: String
Outline End Caps	Specifies the ending style for the label text outline. Choose an option from the drop-down list. Data type: Enumeration
Outline Joint	Specifies the joint style for the label text outline. Choose an option from the drop-down list. Data type: Enumeration
Outline Outlined	Specifies whether to show the label text outline in outline form. If the property is set to true, the outline will be shown in outline form; otherwise, in whole form. Data type: Boolean
Outline Style	Specifies the style for the label text outline. Choose an option from the drop-down list. Data type: Enumeration
Outline Thickness	Specifies the thickness for the label text outline. Enter a numeric value to change the thickness. Data type: Float
Outline Transparency	Specifies the transparency for the label text outline, in percent. Enter a numeric value to change the transparency. Data type: Integer
Outline Variable Dash	Specifies whether to resize the dash automatically. If the property is set to true, the dash size will be adjusted automatically; otherwise, the dash size will be of fixed size. Data type: Boolean
Percent Format	Specifies the format of the percentage values on the legend. Applied when Show Percent is set to true. Click ... in the value cell to specify the format in the Data Format dialog. Data type: String
Reverse Labels	Specifies whether to reverse the order of the labels in the legend object. Data type: Boolean
Show Percent	Specifies whether to show the percentage of each entry value to the total on the legend. Applies to pie, bar and bench charts that have only one value axis. Data type: Boolean
Show Values	Specifies whether to show the values for the legend entries. Applies to pie, bar and bench charts that have only one value axis. Data type: Boolean
Text of Total Value	Specifies the text of the total value label on the legend. Applies to pie, bar and bench charts that have only one value axis. Data type: String
Texture Background Color	Specifies the background color of the texture, applied when the property Fill Type is set to texture. Choose a color from the drop-down list, enter a hexadecimal RGB value (for example, 0xff0000), choose/create a formula or edit an expression that returns a color, or select Custom from the drop-down list to specify the color. Data type: String
Texture Foreground Color	Specifies the foreground color of the texture, applied when the property Fill Type is set to texture. Choose a color from the drop-down list, enter a hexadecimal RGB value (for example, 0xff0000), choose/create a formula or edit an expression that returns a color, or select Custom from the drop-down list to specify the color. Data type: String
Texture Style	Specifies the texture style for the label, applied when the property Fill Type is set to texture. Choose an option from the drop-down list. Data type: Enumeration
Value Format	Specifies the number format of the legend entry values. Applied when Show Values is set to true. Click ... in the value cell to specify the format in the Data Format dialog. Data type: String
Word Wrap	Specifies whether to enable the word wrap function for the label text, not applying to radar charts, bubble charts, scatter charts and gauge charts. If the text contains special characters (such as ",", "." and space), it will be broken at the position of one of the special characters. Data type: Boolean
Icon	
Border Color	Specifies the color of the icon borders. Choose a color from the drop-down list, enter a hexadecimal RGB value (for example, 0xCCAA00), choose/create a formula or edit an expression that returns a color, or select Custom from the drop-down list to specify the color. Data type: String
Border End Caps	Specifies the ending style of the icon border line. Choose an option from the drop-down list. Data type: Enumeration
Border Joint	Specifies the joint style of the icon border line. Choose an option from the drop-down list. Data type: Enumeration
Border Outlined	Specifies whether to show the icon border line in Outline Path. If the property is set to true, the icon border line will be shown in Outline Path; otherwise, in Fill Path. Data type: Boolean
Border Style	Specifies the line style of the icon borders. Choose an option from the drop-down list. Data type: Enumeration
Border Thickness	Specifies the width of the icon borders in inches or centimeters. Enter a numeric value to change the thickness. Data type: Float
Border Transparency	Specifies the transparency of the icon borders, in percent. Enter a numeric value to change the transparency. Data type: Integer
Border Variable Dash	Specifies whether to resize the dash automatically. If the property is set to true, the dash size will be adjusted automatically; otherwise, the dash size will be of fixed size. Data type: Boolean
Height	Specifies the height of the icon, in inches or centimeters. Enter a numeric value to change the width. Data type: Float
Icon Alignment	Specifies the relative position of the icon to the legend. Choose an option from the drop-down list. Data type: Enumeration
Icon Text Gap	Specifies the distance between each legend label and icon. Enter a numeric value to change the gap. Data type: Float
Width	Specifies the width of the icon, in inches or centimeters. Enter a numeric value to change the width. Data type: Float

Chart label

The properties of a chart label in a page report are:

Property Name	Description
General	
Class Type	Indicates the class type of the object. This property is read only.
Data Inherit	Indicates whether the dataset for this object is inherited from another object. This property is read only.
Dataset	Indicates the dataset used by the object. This property is read only.
Instance Name	Shows the instance name of the object. This property is read only.
Geometry	
Height	Specifies the height of the object, in inches or centimeters. Enter a numeric value to change the height. Data type: Float
Width	Specifies the width of the object, in inches or centimeters. Enter a numeric value to change the width. Data type: Float
X	Specifies the horizontal coordinate of the top left corner of the object, relative to its parent container. Enter a numeric value to change the position in inches or centimeters. Data type: Float
Y	Specifies the vertical coordinate of the top left corner of the object, relative to its parent container. Enter a numeric value to change the position in inches or centimeters. Data type: Float
CSS	
Class	Specifies a CSS class to be applied to the object which is a valid class in the CSS file. Data type: String
Style	The property can be used in two ways. • Specifies a style to be applied to the object. Choose a style from the drop-down list which is available when a Style Group has been selected at the report tab level and when there are styles in the Style Group that can be applied to the object, or type in the style name. • Specifies a CSS selector to be applied to the object. Enter a valid class name from the CSS file. Data type: String
Accessibility	
External CSS Class Selector	Specifies a class selector to be applied to the object when exported as HTML. Enter a valid class name from the CSS file. Data type: String
Label	
Bottom Margin	Specifies the distance in inches or centimeters between the label text and the bottom border of the label. The four properties, Bottom Margin, Left Margin, Right Margin and Top Margin work together with the property Text Alignment. Enter a numeric value to change the margin. Data type: Float
Color	Specifies the color of the label, applied when the property Fill Type is set to color. Choose a color from the drop-down list, enter a hexadecimal RGB value (for example, 0xff0000), choose/create a formula or edit an expression that returns a color, or select Custom from the drop-down list to specify the color. Data type: String
Fill Transparency	Specifies the transparency of the background, in percent. Enter a numeric value to change the transparency. Data type: Integer
Fill Type	Specifies the fill pattern for the label. Choose an option from the drop-down list. Data type: Integer
Font Effect	Specifies the special effect for the label text. Choose an option from the drop-down list. Data type: Enumeration
Font Face	Specifies the font face of the label text. Choose an option from the drop-down list. Data type: Enumeration
Font Rotation	Specifies the rotation angle of the label text around its center, in degrees. The default value is 0. Enter a numeric value to change the rotation. Data type: Float
Font Script	Specifies whether the label text will be in superscript or subscript form, or neither of them. Choose an option from the drop-down list. Data type: Enumeration
Font Shearing	Specifies the shearing transformation of the label text around its center. The default value is 0. Enter a numeric value to change the shearing. Data type: Float
Font Size	Specifies the font size of the label text. Enter an integer value to change the size. Data type: Integer
Font Strikethrough	Specifies the style of the line by which the label text is struck through. Choose an option from the drop-down list. Data type: Enumeration
Font Style	Specifies the font style for the label text. Choose an option from the drop-down list. Data type: Enumeration
Font Underline	Specifies the underline style for the label text. Choose an option from the drop-down list. Data type: Enumeration
Gradient End Color	Specifies the color of the point where the gradient ends, applied when the property Fill Type is set to gradient. Choose a color from the drop-down list, enter a hexadecimal RGB value (for example, 0xff0000), choose/create a formula or edit an expression that returns a color, or select Custom from the drop-down list to specify the color. Data type: String
Gradient End X	Specifies the horizontal position where the gradient ends, measured in a percentage of the label text width. Applied when the property Fill Type is set to gradient. Data type: Integer
Gradient End Y	Specifies the vertical position where the gradient ends, measured in a percentage of the label text height. Applied when the property Fill Type is set to gradient. Data type: Integer
Gradient Start Color	Specifies the color of the point where the gradient begins, applied when the property Fill Type is set to gradient. Choose a color from the drop-down list, enter a hexadecimal RGB value (for example, 0xff0000), choose/create a formula or edit an expression that returns a color, or select Custom from the drop-down list to specify the color. Data type: String
Gradient Start X	Specifies the horizontal position where the gradient begins, measured in a percentage of the label text width. Applied when the property Fill Type is set to gradient. Data type: Integer
Gradient Start Y	Specifies the vertical position where the gradient begins, measured in a percentage of the label text height. Applied when the property Fill Type is set to gradient. Data type: Integer
Gradient Style	Specifies the gradient style for the label, applied when the property Fill Type is set to gradient. Choose an option from the drop-down list. Data type: Enumeration
Left Margin	Specifies the distance in inches or centimeters between the label text and the left border of the label. The four properties, Bottom Margin, Left Margin, Right Margin and Top Margin work together with the property Text Alignment. Enter a numeric value to change the margin. Data type: Float
Outline Color	Specifies the color for the label text outline. Choose a color from the drop-down list, enter a hexadecimal RGB value (for example, 0xff0000), choose/create a formula or edit an expression that returns a color, or select Custom from the drop-down list to specify the color. Data type: String
Outline End Caps	Specifies the ending style for the label text outline. Choose an option from the drop-down list. Data type: Integer
Outline Joint	Specifies the joint style for the label text outline. Choose an option from the drop-down list. Data type: Enumeration
Outline Outlined	Specifies whether to show the label text outline in outline form. If the property is set to true, the outline will be shown in outline form; otherwise, in whole form. Data type: Boolean
Outline Style	Specifies the style for the label text outline. Choose an option from the drop-down list. Data type: Enumeration
Outline Thickness	Specifies the thickness for the label text outline. Data type: Float
Outline Transparency	Specifies the transparency for the label text outline, in percent. Enter a numeric value to change the transparency. Data type: Integer
Outline Variable Dash	Specifies whether to resize the dash automatically. If the property is set to true, the dash size will be adjusted automatically; otherwise, the dash size will be of fixed size. Data type: Boolean
Right Margin	Specifies the distance in inches or centimeters between the label text and the right border of the label. The four properties, Bottom Margin, Left Margin, Right Margin and Top Margin work together with the property Text Alignment. Enter a numeric value to change the margin. Data type: Float
Text	Specifies the text of the label. Enter a string to change the text. Data type: String
Text Alignment	Specifies the alignment style of the label text. It works together with the four properties, Bottom Margin, Left Margin, Right Margin and Top Margin. Choose an option from the drop-down list. Data type: Enumeration
Texture Background Color	Specifies the background color of the texture, applied when the property Fill Type is set to texture. Choose a color from the drop-down list, enter a hexadecimal RGB value (for example, 0xff0000), choose/create a formula or edit an expression that returns a color, or select Custom from the drop-down list to specify the color. Data type: String
Texture Foreground Color	Specifies the foreground color of the texture, applied when the property Fill Type is set to texture. Choose a color from the drop-down list, enter a hexadecimal RGB value (for example, 0xff0000), choose/create a formula or edit an expression that returns a color, or select Custom from the drop-down list to specify the color. Data type: String
Texture Style	Specifies the texture style for the label, applied when the property Fill Type is set to texture. Choose an option from the drop-down list. Data type: Enumeration
Top Margin	Specifies the distance in inches or centimeters between the label text and the top border of the label. The four properties, Bottom Margin, Left Margin, Right Margin and Top Margin work together with the property Text Alignment. Enter a numeric value to change the margin. Data type: Float
Word Wrap	Specifies whether to enable the word wrap function for the label text, not applying to radar charts, bubble charts, scatter charts and gauge charts. If the text contains special characters (such as ",", "." and space), it will be broken at the position of one of the special characters. Data type: Boolean
Background	
Border Color	Specifies the color of the label border. Choose a color from the drop-down list, enter a hexadecimal RGB value (for example, 0xCCAA00), choose/create a formula or edit an expression that returns a color, or select Custom from the drop-down list to specify the color. Data type: String
Border End Caps	Specifies the ending style of the label border line. Choose an option from the drop-down list. Data type: Enumeration
Border Joint	Specifies the joint style of the label border line. Choose an option from the drop-down list. Data type: Enumeration
Border Outlined	Specifies whether to show the label border line in Outline Path. If the property is set to true, the label border line will be shown in Outline Path; otherwise, in Fill Path. Data type: Boolean
Border Style	Specifies the line style of the label border. Choose an option from the drop-down list. Data type: Enumeration
Border Thickness	Specifies the width of the label border, in inches or centimeters. Enter a numeric value to change the thickness. Data type: Float
Border Transparency	Specifies the transparency of the label border, in percent. Enter a numeric value to change the transparency. Data type: Integer
Border Type	Specifies the border type of the label. Choose an option from the drop-down list. Data type: Enumeration
Border Variable Dash	Specifies whether to resize the dash automatically. • **true** - If selected, the dash size will be adjusted automatically, and the option Auto Adjust Dash in the Format Label dialog will take effect. • **false** - If selected, the dash size will be of fixed size, and the option Fixed Dash Size in the Format Label dialog will take effect. Data type: Boolean
Color	Specifies the color of the label background, applied when the property Fill Type is set to color. Choose a color from the drop-down list, enter a hexadecimal RGB value (for example, 0xff0000), choose/create a formula or edit an expression that returns a color, or select Custom from the drop-down list to specify the color. Data type: String
Fill Transparency	Specifies the transparency of the label background, in percent. Enter a numeric value to change the transparency. Data type: Integer
Fill Type	Specifies the fill pattern for the label background. Choose an option from the drop-down list. Data type: Enumeration
Gradient End Color	Specifies the color of the point where the gradient ends, applied when the property Fill Type is set to gradient. Choose a color from the drop-down list, enter a hexadecimal RGB value (for example, 0xff0000), choose/create a formula or edit an expression that returns a color, or select Custom from the drop-down list to specify the color. Data type: String
Gradient End X	Specifies the horizontal position where the gradient ends, measured in a percentage of the label width. Applied when the property Fill Type is set to gradient. Data type: Integer
Gradient End Y	Specifies the vertical position where the gradient ends, measured in a percentage of the label height. Applied when the property Fill Type is set to gradient. Data type: Integer
Gradient Start Color	Specifies the color of the point where the gradient begins, applied when the property Fill Type is set to gradient. Choose a color from the drop-down list, enter a hexadecimal RGB value (for example, 0xff0000), choose/create a formula or edit an expression that returns a color, or select Custom from the drop-down list to specify the color. Data type: String
Gradient Start X	Specifies the horizontal position where the gradient begins, measured in a percentage of the label width. Applied when the property Fill Type is set to gradient. Data type: Integer
Gradient Start Y	Specifies the vertical position where the gradient begins, measured in a percentage of the label height. Applied when the property Fill Type is set to gradient. Data type: Integer
Gradient Style	Specifies the gradient style for the label background, applied when the property Fill Type is set to gradient. Choose an option from the drop-down list. Data type: Enumeration
Image File	Specifies the file name of the image, a portion of which will be displayed as the label background. Applied when the property Fill Type is set to image. Enter the file name with suffix in the value cell. If the image is outside of the current catalog, you should include the full path of the image correctly. Data type: String
Image Height	The four properties, Image X, Image Y, Image Width, and Image Height, together are used to define a portion of the image (the portion is a rectangle), which will be displayed as the label background. They take effect when the property Fill Type is set to image. Image Height specifies the height of the image portion, measured in a percentage of the image height. Data type: Integer
Image Layout	Specifies the layout style for the image that will be displayed as the label background, applied when the property Fill Type is set to image. Choose an option from the drop-down list. Data type: Integer
Image Width	The four properties, Image X, Image Y, Image Width, and Image Height, together are used to define a portion of the image (the portion is a rectangle), which will be displayed as the label background. They take effect when the property Fill Type is set to image. Image Width specifies the width of the image portion, measured in a percentage of the image width. Data type: Integer
Image X	The four properties, Image X, Image Y, Image Width, and Image Height, together are used to define a portion of the image (the portion is a rectangle), which will be displayed as the label background. They take effect when the property Fill Type is set to image. Image X specifies the distance between the left border of image and the portion that will contain the pattern, measured in a percentage of the image width. Data type: Integer
Image Y	The four properties, Image X, Image Y, Image Width, and Image Height, together are used to define a portion of the image (the portion is a rectangle), which will be displayed as the label background. They take effect when the property Fill Type is set to image. Image Y specifies the distance between the top border of image and the portion that will contain the pattern, measured in a percentage of the image height. Data type: Integer
Inset Bottom	Specifies the bottom position of the background area, measured in a percentage of the label height, from the bottom edge of the label. Data type: Float
Inset Left	Specifies the left position of the background area, measured in a percentage of the label width, from the left edge of the label. Data type: Float
Inset Right	Specifies the right position of the background area, measured in a percentage of the label width, from the right edge of the label. Data type: Float
Inset Top	Specifies the top position of the background area, measured in a percentage of the label height, from the top edge of the label. Data type: Float
Texture Background Color	Specifies the background color of the texture, applied when the property Fill Type is set to texture. Choose a color from the drop-down list, enter a hexadecimal RGB value (for example, 0xff0000), choose/create a formula or edit an expression that returns a color, or select Custom from the drop-down list to specify the color. Data type: String
Texture Foreground Color	Specifies the foreground color of the texture, applied when the property Fill Type is set to texture. Choose a color from the drop-down list, enter a hexadecimal RGB value (for example, 0xff0000), choose/create a formula or edit an expression that returns a color, or select Custom from the drop-down list to specify the color. Data type: String
Texture Style	Specifies the texture style for the label background, applied when the property Fill Type is set to texture. Choose an option from the drop-down list. Data type: Enumeration
Icon	
Border Color	Specifies the color of the border of the object. Choose a color from the drop-down list, enter a hexadecimal RGB value (for example, 0xCCAA00), choose/create a formula or edit an expression that returns a color, or select Custom from the drop-down list to specify the color. Data type: String
Border End Caps	Specifies the ending style of the icon border line. Choose an option from the drop-down list. Data type: Enumeration
Border Joint	Specifies the joint style of the icon border line. Choose an option from the drop-down list. Data type: Enumeration
Border Outlined	Specifies whether to show the icon border line in Outline Path. If the property is set to true, the icon border line will be shown in Outline Path; otherwise, in Fill Path. Data type: Boolean
Border Style	Specifies the line style of the icon borders. Choose an option from the drop-down list. Data type: Enumeration
Border Thickness	Specifies the width of the border in inches or centimeters. Enter a numeric value to change the thickness. Data type: Float
Border Transparency	Specifies the transparency of the icon borders, in percent. Enter a numeric value to change the transparency. Data type: Integer
Border Variable Dash	Specifies whether to resize the dash automatically. If the property is set to true, the dash size will be adjusted automatically; otherwise, the dash size will be of fixed size. Data type: Boolean
Color	Specifies the color of the icon background, applied when the property Fill Type is set to color. Choose a color from the drop-down list, enter a hexadecimal RGB value (for example, 0xff0000), choose/create a formula or edit an expression that returns a color, or select Custom from the drop-down list to specify the color. Data type: String
Fill Transparency	Specifies the transparency of the icon background, in percent. Enter a numeric value to change the transparency. Data type: Integer
Fill Type	Specifies the fill pattern for the icon. Choose an option from the drop-down list. Data type: Enumeration
Gradient End Color	Specifies the color of the point where the gradient ends, applied when the property Fill Type is set to gradient. Choose a color from the drop-down list, enter a hexadecimal RGB value (for example, 0xff0000), choose/create a formula or edit an expression that returns a color, or select Custom from the drop-down list to specify the color. Data type: String
Gradient End X	Specifies the horizontal position where the gradient ends, measured in a percentage of the icon width. Applied when the property Fill Type is set to gradient. Data type: Integer
Gradient End Y	Specifies the vertical position where the gradient ends, measured in a percentage of the icon height. Applied when the property Fill Type is set to gradient. Data type: Integer
Gradient Start Color	Specifies the color of the point where the gradient begins, applied when the property Fill Type is set to gradient. Choose a color from the drop-down list, enter a hexadecimal RGB value (for example, 0xff0000), choose/create a formula or edit an expression that returns a color, or select Custom from the drop-down list to specify the color. Data type: String
Gradient Start X	Specifies the horizontal position where the gradient begins, measured in a percentage of the icon width. Applied when the property Fill Type is set to gradient. Data type: Integer
Gradient Start Y	Specifies the vertical position where the gradient begins, measured in a percentage of the icon height. Applied when the property Fill Type is set to gradient. Data type: Integer
Gradient Style	Specifies the gradient style for the icon, applied when the property Fill Type is set to gradient. Choose an option from the drop-down list. Data type: Enumeration
Height	Specifies the height of the icon, in inches or centimeters. Enter a numeric value to change the width. Data type: Float
Icon Alignment	Specifies the relative position of the icon to the label. Choose an option from the drop-down list. Data type: Enumeration
Icon Style	Specifies the style of the icon attached to the label. Choose an option from the drop-down list. Data type: Enumeration
Icon Text Gap	Specifies the distance between the icon and the label. Data type: Float
Image File	Specifies the file name of the image, a portion of which will be displayed as the icon background. Applied when the property Fill Type is set to image. Enter the file name with suffix in the value cell. If the image is outside of the current catalog, you should include the full path of the image correctly. Data type: String
Image Height	The four properties, Image X, Image Y, Image Width, and Image Height, together are used to define a portion of the image (the portion is a rectangle), which will be displayed as the icon background. They take effect when the property Fill Type is set to image. Image Height specifies the height of the image portion, measured in a percentage of the image height. Data type: Integer
Image Layout	Specifies the layout style for the image that will be displayed as the icon background, applied when the property Fill Type is set to image. Choose an option from the drop-down list. Data type: Enumeration
Image Width	The four properties, Image X, Image Y, Image Width, and Image Height, together are used to define a portion of the image (the portion is a rectangle), which will be displayed as the icon background. They take effect when the property Fill Type is set to image. Image Width specifies the width of the image portion, measured in a percentage of the image width. Data type: Integer
Image X	The four properties, Image X, Image Y, Image Width, and Image Height, together are used to define a portion of the image (the portion is a rectangle), which will be displayed as the icon background. They take effect when the property Fill Type is set to image. Image X specifies the distance between the left border of image and the portion that will contain the pattern, measured in a percentage of the image width. Data type: Integer
Image Y	The four properties, Image X, Image Y, Image Width, and Image Height, together are used to define a portion of the image (the portion is a rectangle), which will be displayed as the icon background. They take effect when the property Fill Type is set to image. Image Y specifies the distance between the top border of image and the portion that will contain the pattern, measured in a percentage of the image height. Data type: Integer
Texture Background Color	Specifies the background color of the texture, applied when the property Fill Type is set to texture. Choose a color from the drop-down list, enter a hexadecimal RGB value (for example, 0xff0000), choose/create a formula or edit an expression that returns a color, or select Custom from the drop-down list to specify the color. Data type: String
Texture Foreground Color	Specifies the foreground color of the texture, applied when the property Fill Type is set to texture. Choose a color from the drop-down list, enter a hexadecimal RGB value (for example, 0xff0000), choose/create a formula or edit an expression that returns a color, or select Custom from the drop-down list to specify the color. Data type: String
Texture Style	Specifies the texture style for the icon, applied when the property Fill Type is set to texture. Choose an option from the drop-down list. Data type: Enumeration
Width	Specifies the width of the icon, in inches or centimeters. Enter a numeric value to change the width. Data type: Float

Crosstab

The properties of a crosstab in a page report are:

Property Name	Description
General	
Class Type	Indicates the class type of the object. This property is read only.
Data Inherit	Indicates whether the dataset for this object is inherited from another object. This property is read only.
Dataset	Indicates the dataset used by the object. This property is read only.
Instance Name	Shows the instance name of the object. This property is read only.
Geometry	
Height	Displays the height of the object, in inches or centimeters. This property is read only.
Width	Displays the width of the object, in inches or centimeters. This property is read only.
X	Specifies the horizontal coordinate of the top left corner of the object, relative to its parent container. This property is ignored if the Position property is set to static. Enter a numeric value to change the position in inches or centimeters. Data type: Float
Y	Specifies the vertical coordinate of the top left corner of the object, relative to its parent container. This property is ignored if the Position property is set to static. Enter a numeric value to change the position in inches or centimeters. Data type: Float
Color	
Background	Specifies the background color of the object. Choose a color from the drop-down list, enter a hexadecimal RGB value (for example, 0xff0000), choose/create a formula or edit an expression that returns a color, or select Custom from the drop-down list to specify the color. Data type: String
Alternating Line Color	
Direction	Specifies whether the color defined by the property Pattern List will apply to the fields in the same rows or columns in the crosstab. • **vertical** - The value of the property Pattern List will take effect on the fields in the same columns, except for the total columns. • **horizontal** - The value of the property Pattern List will take effect on the fields in the same rows, except for the total rows. Data type: Enumeration
Enable	Specifies whether to enable setting alternating line color for the crosstab. Only when it is set to true, the settings for the properties Direction and Pattern List can take effect. Data type: Boolean **Note:** The setting of the alternating line color function for crosstab can only take effect when all the aggregate fields in the crosstab have transparent background.
Pattern List	Specifies the color pattern for the fields in the same rows or columns in the crosstab. Click in the value cell and click the small squares in the color tray one by one to specify the patterns. The patterns can be one or more of the following: Automatic, Color, Texture, and Gradient. Data type: String
CSS	
Class	Specifies a CSS class to be applied to the object which is a valid class in the CSS file. Data type: String
Style	The property can be used in two ways. • Specifies a style to be applied to the object. Choose a style from the drop-down list which is available when a Style Group has been selected at the report tab level and when there are styles in the Style Group that can be applied to the object, or type in the style name. • Specifies a CSS selector to be applied to the object. Enter a valid class name from the CSS file. Data type: String
Others	
Cache	Specifies whether to cache the dataset for this object in the data buffer so other objects which use the same dataset can share the data rather than perform their own SQL query. Data type: Boolean
Current Column Block Index	Specifies the horizontal index of the data block that will be displayed. 0 means the first block index, and 1 the second, and so on. The four properties work together to control the data of the crosstab to be displayed in continuous page mode: Current Row Block Index, Current Column Block Index, Items per Row Block, and Items per Column Block. Data type: Integer
Current Row Block Index	Specifies the vertical index of the data block that will be displayed. 0 means the first block index, and 1 the second, and so on. The four properties work together to control the data of the crosstab to be displayed in continuous page mode: Current Row Block Index, Current Column Block Index, Items per Row Block, and Items per Column Block. Data type: Integer
Default for Filter	Specifies whether to show the data component as the default component in the Apply to drop-down list of the Filter dialog in Page Report Studio. Data type: Boolean **Note:** In a page report tab, only one data component's Default for Filter property can be set to true.
Export to Applet	Specifies whether to include the object when exporting the report to Applet. Data type: Boolean
Export to CSV	Specifies whether to include the object when exporting the report to CSV. Data type: Boolean
Export to Excel	Specifies whether to include the object when exporting the report to Excel. Data type: Boolean
Export to HTML	Specifies whether to include the object when exporting the report to HTML. Data type: Boolean
Export to PDF	Specifies whether to include the object when exporting the report to PDF. Data type: Boolean
Export to PostScript	Specifies whether to include the object when exporting the report to PostScript. Data type: Boolean
Export to Report Result	Specifies whether to include the object when exporting the report to Report Result. Data type: Boolean
Export to RTF	Specifies whether to include the object when exporting the report to RTF. Data type: Boolean
Export to Text	Specifies whether to include the object when exporting the report to Text. Data type: Boolean
Export to XML	Specifies whether to include the object when exporting the report to XML. Data type: Boolean
Invisible	Specifies whether to show the object in the design area and in the report results. All formulas and calculations will still be performed if the property is set to true. Data type: Boolean
Invisible for Filter Dialogs	Specifies whether to show the data component in the Apply to drop-down list of the Filter dialog in Page Report Studio. This property cannot be edited when Default for Filter is set to true. Data type: Boolean
Items per Column Block	Specifies the number of columns of records in each data block. The four properties work together to control the data of the crosstab to be displayed in continuous page mode: Current Row Block Index, Current Column Block Index, Items per Row Block, and Items per Column Block. Data type: Integer
Items per Row Block	Specifies the number of rows of records in each data block. The four properties work together to control the data of the crosstab to be displayed in continuous page mode: Current Row Block Index, Current Column Block Index, Items per Row Block, and Items per Column Block. Data type: Integer
Position	Specifies the position of the object. Choose an option from the drop-down list. • **static** - The object is positioned at the location in which it is inserted. If selected, the X, Y and other position-related properties will be hidden or disabled. The position of the object is affected by preceding objects in the same container. • **absolute** - The object will be located at the position specified by dragging and dropping or by setting its X and Y property values. The position of the object is not affected by preceding objects in the same container. Data type: Enumeration
Record Location	Specifies the calculation point for the properties which use formulas. Choose an option from the drop-down list. • **Default** - The values of the properties are calculated in the default location where they are located. • **Page Header** - The values of the properties are calculated in the page header. • **Page Footer** - The values of the properties are calculated in the page footer. Data type: Enumeration
Suppress	Specifies whether to show the object in the design area and in the report results. All formulas and calculations will be skipped if the property is set to true. Data type: Boolean **Note:** When both the Invisible and Suppress properties of an object are set to true, Suppress has the higher priority.
Suppress When Empty	Specifies whether to display the object in the report results when no record is returned to it. Data type: Boolean
Suppress When No Records	Specifies whether to display the object in the report results when no record is returned to its parent data container. Data type: Boolean
Excel	
Column Index	Specifies the X coordinate of the object relative to its parent container when exported to Excel or CSV, measured in cells. The Columned property at the report tab level must be set to true for this property to take effect. Data type: Integer
Row Index	Specifies the Y coordinate of the object relative to its parent container when exported to Excel or CSV, measured in cells. The Columned property at the report tab level must be set to true for this property to take effect. Data type: Integer
Border	
Border Color	Specifies the color of the border of the object. Choose a color from the drop-down list, enter a hexadecimal RGB value (for example, 0xCCA000), choose/create a formula or edit an expression that returns a color, or select Custom from the drop-down list to specify the color. Data type: String
Has Border	Specifies whether to show the cell borders. Data type: Boolean
Crosstab Property	
Avoid Orphan Header	Specifies whether to make the column header be together with the data. Sometimes the column header happens to be at the bottom of a page. To keep the column header together with the data in the next page, set this property to true. Data type: Boolean
Block Gap	Specifies the space between each part if the crosstab is split into more than one part. Data type: Float
Boundary Value	Specifies the number of columns in one aggregate cell when the crosstab is displayed horizontally, or rows when displayed vertically. Data type: Integer
Column Totals on Left	Specifies whether to display the total columns on the left of the aggregates. Data type: Boolean
Expand Data	Specifies whether to enable Page Report Studio users to expand/collapse dimensions in the crosstab. This property works only in continuous page mode. Data type: Boolean
Horizontal Gap	Specifies the space between the left/right edge of a crosstab cell and the contents in it. Data type: Float
Outside Aggregate Title	Specifies to place the summary labels to the left of the row values if true, right of the row values if false. Data type: Boolean
Repeat Aggregate	Specifies whether to repeat the crosstab for different aggregate fields. Data type: Boolean
Repeat Column Header	Specifies whether to repeat column headings on every page. Data type: Boolean
Row Totals on Top	Specifies whether to display the total rows on the top of the aggregates. Data type: Boolean
Suppress Column Header	Specifies whether to suppress the column header in view mode. Data type: Boolean
Suppress Row Header	Specifies whether to suppress the row header in view mode. Data type: Boolean
Use Table Style	Specifies whether to add headers to the Total rows and columns. Data type: Boolean
Vertical Gap	Specifies the space between the top/bottom edge of a crosstab cell and the contents in it. Data type: Float
Vertical Layout	Indicates whether the aggregate fields are displayed vertically. The property is read only.
TOC	
Anchor Display Value	Specifies a string or formula to display for the TOC entry for this object in the TOC Browser. By default it is the value of the specified object. The TOC Anchor property must be set to true for this property to take effect. Data type: String
TOC Anchor	Specifies whether to include the object in the TOC Browser for the report. Data type: Boolean
Accessibility	
External CSS Class Selector	Specifies a class selector to be applied to the object when exported as HTML. Enter a valid class name from the CSS file. Data type: String
External ID	It is mapped to the HTML attribute *id*. This attribute specifies a name to the object, which must be unique in the report. Data type: String
External Style	It is mapped to the HTML attribute *style*. This attribute specifies style information for the object. Data type: String
Language	It is mapped to the HTML attribute *lang*. This attribute specifies the base language of the object's attribute values and text content. Data type: String
Summary	It is mapped to the HTML attribute *summary*. This attribute provides a summary of the object's purpose and structure. Data type: String

A crosstab has some child objects, the properties of which will be shown in the following topics:

- DBField
- Formula field
- Label

DBField

The properties of a DBField in a crosstab of a page report are:

Property Name	Description
General	
Column	Shows the column name of the field in the raw database. This property is read only.
DBField Name	Shows the mapping name of the field in the catalog. This property is read only.
Data Type	Shows the data type of the field. This property is read only.
Field Type	Shows what kind of field it is. This property is read only.
Instance Name	Shows the instance name of the object. This property is read only.
Query	Shows the query name. This property is read only.
Table	Shows the table name of the field in the raw database. This property is read only.
Geometry (only for aggregate field)	
Height	Specifies the height of the object, in inches or centimeters. Enter a numeric value to change the height. Data type: Float
Width	Specifies the width of the object, in inches or centimeters. Enter a numeric value to change the width. Data type: Float
X	Specifies the horizontal coordinate of the top left corner of the object, relative to its parent container. Enter a numeric value to change the position in inches or centimeters. Data type: Float
Y	Specifies the vertical coordinate of the top left corner of the object, relative to its parent container. Enter a numeric value to change the position in inches or centimeters. Data type: Float
Color	
Background	Specifies the background color of the object. Choose a color from the drop-down list, enter a hexadecimal RGB value (for example, 0xff0000), choose/create a formula or edit an expression that returns a color, or select Custom from the drop-down list to specify the color. Data type: String
Foreground	Specifies the foreground color of the object. Choose a color from the drop-down list, enter a hexadecimal RGB value (for example, 0xff0000), choose/create a formula or edit an expression that returns a color, or select Custom from the drop-down list to specify the color. Data type: String
CSS	
Class	Specifies a CSS class to be applied to the object which is a valid class in the CSS file. Data type: String
ID	Specifies the identifier of the object, which must be unique in the report. The ID property value can be a style sheet selector. Data type: String
Style	The property can be used in two ways. • Specifies a style to be applied to the object. Choose a style from the drop-down list which is available when a Style Group has been selected at the report tab level and when there are styles in the Style Group that can be applied to the object, or type in the style name. • Specifies a CSS selector to be applied to the object. Enter a valid class name from the CSS file. Data type: String
Padding	
Bottom Padding	Specifies the space in inches or centimeters between the text and the bottom border of the object. Enter a numeric value to change the padding. Data type: Float
Left Padding	Specifies the space in inches or centimeters between the text and the left border of the object. Enter a numeric value to change the padding. Data type: Float
Right Padding	Specifies the space in inches or centimeters between the text and the right border of the object. Enter a numeric value to change the padding. Data type: Float
Top Padding	Specifies the space in inches or centimeters between the text and the top border of the object. Enter a numeric value to change the padding. Data type: Float
Border	
Border Color	Specifies the color of the border of the object. Choose a color from the drop-down list, enter a hexadecimal RGB value (for example, 0xff0000), choose/create a formula or edit an expression that returns a color, or select Custom from the drop-down list to specify the color. Data type: String
Border Thickness	Specifies the width of the border in inches or centimeters. Enter a numeric value to change the thickness. Data type: Float
Bottom Line	Specifies the line style of the bottom border of the object. Choose a style from the drop-down list. Data type: Enumeration
Left Line	Specifies the line style of the left border of the object. Choose a style from the drop-down list. Data type: Enumeration
Right Line	Specifies the line style of the right border of the object. Choose a style from the drop-down list. Data type: Enumeration
Shadow	Specifies whether to draw two shadowed borders, beneath and to the right of the object. Data type: Boolean
Shadow Color	Specifies the color of the shadow. Choose a color from the drop-down list, enter a hexadecimal RGB value (for example, 0xff0000), choose/create a formula or edit an expression that returns a color, or select Custom from the drop-down list to specify the color and fill effect. Data type: String
Top Line	Specifies the line style of the top border of the object. Choose a style from the drop-down list. Data type: Enumeration
Pattern	
Pattern Color	Specifies the color in which to draw a pattern to fill the object. Choose a color from the drop-down list, enter a hexadecimal RGB value (for example, 0xff0000), choose/create a formula or edit an expression that returns a color, or select Custom from the drop-down list to specify the color and fill effect. Data type: String
Pattern Style	Specifies the style of the pattern. Choose an option from the drop-down list. • **none** - No pattern will be applied to the object. • **50 %** - The object will be filled in the Pattern Color of 50%-transparency. • **horizontal** - The pattern will be made of horizontal lines of the Pattern Color. • **vertical** - The pattern will be made of vertical lines of the Pattern Color. • **grid** - The pattern will be an overlapping of horizontal and vertical lines of the Pattern Color. • **diagonal** - The pattern will be made of diagonal lines of the Pattern Color. Data type: Enumeration
Text Format	
Auto Fit	Specifies whether to adjust the width of the object according to the contents. Data type: Boolean
Bold	Specifies whether to make the text bold. Data type: Boolean
Font Face	Specifies the font of the text. Choose an option from the drop-down list. Data type: Enumeration
Font Size	Specifies the font size of the text. Enter an integer value to change the size. Data type: Integer
Format	Specifies the display format of the text in the report result. It varies with data type and can be manually defined. Data type: String **Note**: For the BigDecimal type, to avoid precision loss, you should enter a prefix JRD when setting the format property value.
Horizontal Alignment	Specifies the horizontal justification of the text in the object. Choose an option from the drop-down list. Data type: Enumeration
Ignore HTML Tag	Specifies whether or not to ignore the HTML tags when exporting the report results to HTML format. • **true** - The text will appear in the HTML file the same as that in JReport Designer (HTML tag elements in the text, if any, will not be parsed). • **false** - HTML tag elements in the text will be parsed when exporting the report results to HTML format. This property is useful for exporting the report results to HTML format. You may want to ignore the elements in HTML tags in your report, and for the exported HTML files, the tags will be transferred as they are translated into HTML. Data type: Boolean
Italic	Specifies whether to make the text italic. Data type: Boolean
Maximum Width	Specifies the maximum width of the text displayed. Enter a numeric value to change the width. Data type: Float
Strikethrough	Specifies whether to add a strikethrough line to the text. Data type: Boolean
Underline	Specifies whether to underline the text. Data type: Boolean
Vertical Alignment	Specifies the vertical justification of the text in the object. Choose an option from the drop-down list. Data type: Enumeration
Word Wrap	Specifies whether to wrap the text to the width. Data type: Boolean
Others	
Cache Value	Specifies whether to cache the value of the field instead of obtaining it repeatedly. Data type: Boolean
Column Name	Substitutes the current field with another field by selecting from the drop-down list. To make the property editable, uncheck Forbid changing column in the Options dialog (**File** > **Options** > **Panel** > **Forbid changing column**). Data type: String
Data Mapping File	Specifies the data mapping file to the object for NLS use. For example, if the data mapping file name is Category_de_DE.properties, set the value as Category (without the language and locale part). Data type: String
Detail Target Frame	Specifies the target window or frame in which to display the detailed information. This property is enabled only when Go To Detail is set to true. Choose an option from the drop-down list. • **<Server Setting>** Loads the detailed information according to setting of the Pop Up New Window for Links option in the Customize Profile panel of the Profile dialog in JReport Server. • **New Window** Loads the detailed information into a new window. The window is not named. • **Whole Window** Loads the detailed information into the full browser window. • **Same Frame** Loads the detailed information into the same frame as the object. • **Parent Frame** Loads the detailed information into the parent frame of the frame in which the object is. • **Other Frame** Loads the detailed information into some other specified frame. If selected, you can also directly input the name of the frame you have defined in the text box. If the frame name does not exist, the detailed information will be loaded into a new window. Data type: String
Display Null	Specifies the string to be displayed if the field value is null. Data type: String
Enable Hyperlink in Excel	Specifies whether to enable the link defined on the object when exporting the report to an Excel file. Data type: Boolean
Enable Hyperlink in HTML	Specifies whether to enable the link defined on the object when exporting the report to an HTML file. Data type: Boolean
Enable Hyperlink in PDF	Specifies whether to enable the link defined on the object when exporting the report to a PDF file. Data type: Boolean
Expand Data	Specifies whether to enable Page Report Studio users to expand/collapse the details of the field which is as dimension (group level) in the crosstab. This property works only in continuous page mode and when the Expand Data property of the crosstab is set to true. It is not available for aggregate field. Data type: Boolean
Expand Detail Data	Specifies whether to display the details of the field by default in Page Report Studio. This property works only in continuous page mode, and is not available for aggregate field. It is meaningful to enable the property only when the field's Expand Data property is set to true. Data type: Boolean
Export to CSV	Specifies whether to include the object when exporting the report to CSV. Data type: Boolean
Export to Excel	Specifies whether to include the object when exporting the report to Excel. Data type: Boolean
Export to Page Report Result	Specifies whether to include the object when exporting the report to Page Report Result. Data type: Boolean
Export to PDF	Specifies whether to include the object when exporting the report to PDF. Data type: Boolean
Filter Options	Specifies the filter-related options that will be displayed on the shortcut menu when right-clicking the object in Page Report Studio. If Default is selected, the filter options displayed are decided by the setting in the Profile dialog in JReport Server (Profile > Customize Profile > Page Report Studio > Properties > Default > Filter Menu). Data type: Integer
Go to Detail	Specifies whether to show the detailed information when you click the object in Page Report Studio. This property is available only when the object is in the group header/footer panel of a banded object. For details about its usage, refer to Obtaining detailed information from a banded object. Data type: Boolean
Link	Specifies to link the object to another report, a website, an e-mail address or a Blob data type field. Click [...] in the value cell to set the link target. For details, see Binding a link to a label. Data type: String
Order By	Specifies the field by which to sort the data. Not available for aggregate field. Data type: String
Record Location	Specifies the calculation point for the properties which use formulas. Choose an option from the drop-down list. • **Default** - The values of the properties are calculated in the default location where they are located. • **Page Header** - The values of the properties are calculated in the page header. • **Page Footer** - The values of the properties are calculated in the page footer. Data type: Enumeration
Suppress	Specifies whether to show the object in the design area and in the report results. All formulas and calculations will be skipped if the property is set to true. Data type: Boolean
Suppress When No Records	Specifies whether to display the object in the report results when no record is returned to its parent data container. Data type: Boolean
Suppress When Null	If true and its value is null, the field will not be displayed. Data type: Boolean
Transfer Style	Specifies whether the style group of the primary report will be applied to the linked report if the object is linked to a report. Data type: Boolean
Value Delimiter	Specifies the separator for the data whose type is DBArray. The default value space means that the elements will be displayed in a horizontal line and separated by a space. Data type: String
Accessibility	
Abbr	It is mapped to the HTML attribute abbr. This attribute specifies an abbreviated form of the content of the object. The abbreviated text may be rendered by user agents when appropriate in place of the object's content. Data type: String
Axis	It is mapped to the HTML attribute axis. This attribute is used to place a cell into conceptual categories that can be considered to form axes in an n-dimensional space. The value of this attribute is a comma-separated list of category names. Data type: String
External AccessKey	It is mapped to the HTML attribute accesskey. This attribute specifies an access key to the object. Data type: String
External CSS Class Selector	Specifies a class selector to be applied to the object when exported as HTML. Enter a valid class name from the CSS file. Data type: String
External Dir	It is mapped to the HTML attribute dir. This attribute specifies the base direction of directionally neutral text in the object's content and attribute values. It also specifies the directionality of tables. Possible values: • LTR - Left-to-right text or table. • RTL - Right-to-left text or table. Data type: String
External ID	It is mapped to the HTML attribute id. This attribute specifies a name to the object, which must be unique in the report. Data type: String
External Style	It is mapped to the HTML attribute style. This attribute specifies style information for the object. Data type: String
External TabIndex	It is mapped to the HTML attribute tabindex. This attribute specifies the position of the object in tabbing order for the current report. Enter an integer equal to or greater than 0 and less than 65535. Data type: Integer
External Title	It is mapped to the HTML attribute title. This attribute offers advisory information about the object. Data type: String
Headers	It is mapped to the HTML attribute headers. This attribute specifies the list of header cells that provide header information for the current data cell. The value of this attribute is a space-separated list of cell names; those cells must be named by setting their id attribute. Data type: String
HrefLang	It is mapped to the HTML attribute hreflang. This attribute specifies the base language of the resource designated by a link and may only be used when a link is specified. Data type: String
Language	It is mapped to the HTML attribute lang. This attribute specifies the base language of the object's attribute values and text content. Data type: String
LongDesc	It is mapped to the HTML attribute longdesc. This attribute specifies a link to a long description of the object. Data type: String
Scope	It is mapped to the HTML attribute scope. This attribute specifies the set of data cells for which the current header cell provides header information. Available values are: • **Row** - The current cell provides header information for the rest of the row that contains it. • **Column** - The current cell provides header information for the rest of the column that contains it. • **none** - The scope attribute will not be generated when exporting to HTML. Data type: String

Formula field

The properties of a formula field in a crosstab of a page report are:

Property Name	Description
General	
Data Type	Shows the data type of the formula. This property is read only.
Expression	Shows the expression of the formula. This property is read only.
Field Type	Shows what kind of field is in. This property is read only.
Formula Editor	Edits the formula. Click [...] to display the Formula Editor, in which you can edit the formula as required.
Formula Name	Shows the name of the formula. This property is read only.
Instance Name	Shows the instance name of the object. This property is read only.
Geometry (only for aggregate field)	
Height	Specifies the height of the object, in inches or centimeters. Enter a numeric value to change the height. Data type: Float
Width	Specifies the width of the object, in inches or centimeters. Enter a numeric value to change the width. Data type: Float
X	Displays the horizontal coordinate of the top left corner of the object, relative to its parent container. This property is read only.
Y	Displays the vertical coordinate of the top left corner of the object, relative to its parent container. This property is read only.
Color	
Background	Specifies the background color of the object. Choose a color from the drop-down list, enter a hexadecimal RGB value (for example, 0xff0000), choose/create a formula or edit an expression that returns a color, or select Custom from the drop-down list to specify the color. Data type: String
Foreground	Specifies the foreground color of the object. Choose a color from the drop-down list, enter a hexadecimal RGB value (for example, 0xff0000), choose/create a formula or edit an expression that returns a color, or select Custom from the drop-down list to specify the color. Data type: String
CSS	
Class	Specifies a CSS class to be applied to the object which is a valid class in the CSS file. Data type: String
ID	Specifies the identifier of the object, which must be unique in the report. The ID property can be a style sheet selector. Data type: String
Style	The property can be used in two ways. • Specifies a style to be applied to the object. Choose a style from the drop-down list which is available when a Style Group has been selected at the report tab level and when there are styles in the Style Group that can be applied to the object, or type in the style name. • Specifies a CSS selector to be applied to the object. Enter a valid class name from the CSS file. Data type: String
Padding	
Bottom Padding	Specifies the space in inches or centimeters between the text and the bottom border of the object. Enter a numeric value to change the padding. Data type: Float
Left Padding	Specifies the space in inches or centimeters between the text and the left border of the object. Enter a numeric value to change the padding. Data type: Float
Right Padding	Specifies the space in inches or centimeters between the text and the right border of the object. Enter a numeric value to change the padding. Data type: Float
Top Padding	Specifies the space in inches or centimeters between the text and the top border of the object. Enter a numeric value to change the padding. Data type: Float
Border	
Border Color	Specifies the color of the border of the object. Choose a color from the drop-down list, enter a hexadecimal RGB value (for example, 0xCCAA00), choose/create a formula or edit an expression that returns a color, or select Custom from the drop-down list to specify the color. Data type: String
Border Thickness	Specifies the width of the border in inches or centimeters. Enter a numeric value to change the thickness. Data type: Float
Bottom Line	Specifies the line style of the bottom border of the object. Choose a style from the drop-down list. Data type: Enumeration
Left Line	Specifies the line style of the left border of the object. Choose a style from the drop-down list. Data type: Enumeration
Right Line	Specifies the line style of the right border of the object. Choose a style from the drop-down list. Data type: Enumeration
Shadow	Specifies whether to draw two shadowed borders, beneath and to the right of the object. Data type: Boolean
Shadow Color	Specifies the color of the shadow. Choose a color from the drop-down list, enter a hexadecimal RGB value (for example, 0xff0000), choose/create a formula or edit an expression that returns a color, or select Custom from the drop-down list to specify the color and fill effect. Data type: String
Top Line	Specifies the line style of the top border of the object. Choose a style from the drop-down list. Data type: Enumeration
Pattern	
Pattern Color	Specifies the color in which to draw a pattern to fill the object. Choose a color from the drop-down list, enter a hexadecimal RGB value (for example, 0xff0000), choose/create a formula or edit an expression that returns a color, or select Custom from the drop-down list to specify the color and fill effect. Data type: String
Pattern Style	Specifies the style of the pattern. Choose an option from the drop-down list. • **none** - No pattern will be applied to the object. • **50%** - The object will be filled in the Pattern Color of 50%-transparency. • **horizontal** - The pattern will be made of horizontal lines of the Pattern Color. • **vertical** - The pattern will be made of vertical lines of the Pattern Color. • **grid** - The pattern will be an overlapping of horizontal and vertical lines of the Pattern Color. • **diagonal** - The pattern will be made of diagonal lines of the Pattern Color. Data type: Enumeration
Text Format	
Auto Fit	Specifies whether to adjust the width of the object according to the contents. Data type: Boolean
Bold	Specifies whether to make the text bold. Data type: Boolean
Font Face	Specifies the font of the text. Choose an option from the drop-down list. Data type: Enumeration
Font Size	Specifies the font size of the text. Enter an integer value to change the size. Data type: Integer
Format	Specifies the display format of the text in the report result. It varies with data type and can be manually defined. Data type: String **Note:** For the BigDecimal type, to avoid precision loss, you should enter a prefix JRD when setting the format property value.
Horizontal Alignment	Specifies the horizontal justification of the text in the object. Choose an option from the drop-down list. Data type: Enumeration
Ignore HTML Tag	Specifies whether or not to ignore the HTML tags when exporting the report results to HTML format. • **true** - The text will appear in the HTML file the same as that in JReport Designer (HTML tag elements in the text, if any, will not be parsed). • **false** - HTML tag elements in the text will be parsed when exporting the report results to HTML format. This property is useful for exporting the report results to HTML format. You may want to write the label contents in HTML tags in your report, and for the exported HTML files, the tags will be transferred as they are translated into HTML. Data type: Boolean
Italic	Specifies whether to make the text italic. Data type: Boolean
Maximum Width	Specifies the maximum width of the text displayed. Enter a numeric value to change the width. Data type: Float
Strikethrough	Specifies whether to add a strikethrough line to the text. Data type: Boolean
Underline	Specifies whether to underline the text. Data type: Boolean
Vertical Alignment	Specifies the vertical justification of the text in the object. Choose an option from the drop-down list. Data type: Enumeration
Word Wrap	Specifies whether to wrap the text to the width. Data type: Boolean
Others	
Cache Value	Specifies whether to cache the value of the field instead of obtaining it repeatedly. Data type: Boolean
Column Name	Substitutes the current field with another field by selecting from the drop-down list. To make the property editable, uncheck Forbid changing column in the Options dialog (**File > Options > Panel > Forbid changing column**). Data type: String
Data Mapping File	Specifies the data mapping file to the object for NLS use. For example, if the data mapping file name is Category_de_DE.properties, set the value as Category (without the language and locale part). Data type: String
Detail Target Frame	Specifies the target window or frame in which to display the detailed information. This property is enabled only when Go To Detail is set to true. Choose an option from the drop-down list. • **<Server Setting>** Loads the detailed information according to setting of the Pop Up New Window for Links option in the Customize Profile panel of the Profile dialog in JReport Server. • **New Window** Loads the detailed information into a new window. The window is not named. • **Whole Window** Loads the detailed information into the full browser window. • **Same Frame** Loads the detailed information into the same frame as the object. • **Parent Frame** Loads the detailed information into the parent frame of the frame in which the object is. • **Other Frame** Loads the detailed information into some other specified frame. If selected, you can also directly input the name of the frame you have defined in the text box. If the frame name does not exist, the detailed information will be loaded into a new window. Data type: String
Display Null	Specifies the string to be displayed if the field value is null. Data type: String
Enable Hyperlink in Excel	Specifies whether to enable the link defined on the object when exporting the report to an Excel file. Data type: Boolean
Enable Hyperlink in HTML	Specifies whether to enable the link defined on the object when exporting the report to an HTML file. Data type: Boolean
Enable Hyperlink in PDF	Specifies whether to enable the link defined on the object when exporting the report to a PDF file. Data type: Boolean
Expand Data	Specifies whether to enable Page Report Studio users to expand/collapse the details of the field which is at dimension (group level) in the crosstab. This property works only in continuous page mode and when the Expand Data property of the crosstab is set to true. It is not available for aggregate field. Data type: Boolean
Expand Detail Data	Specifies whether to display the details of the field by default in Page Report Studio. This property works only in continuous page mode, and is not available for aggregate field. It is meaningful to enable the property only when the field's Expand Data property is set to true. Data type: Boolean
Export to CSV	Specifies whether to include the object when exporting the report to CSV. Data type: Boolean
Export to Excel	Specifies whether to include the object when exporting the report to Excel. Data type: Boolean
Export to Page Report Result	Specifies whether to include the object when exporting the report to Page Report Result. Data type: Boolean
Export to PDF	Specifies whether to include the object when exporting the report to PDF. Data type: Boolean
Filter Options	Specifies the filter-related options that will be displayed on the shortcut menu when right-clicking the object in Page Report Studio. If Default is selected, the filter options displayed are decided by the setting in the Profile dialog in JReport Server (Profile > Customize Profile > Page Report Studio > Properties > Default > Filter Menu). For more information, refer to Setting filter options for a field.
Go to Detail	Specifies whether to show the detailed information when you click the object in Page Report Studio. This property is available only when the object is in the group header/footer panel of a banded object. For details about its usage, refer to Obtaining detailed information from a banded object. Data type: Boolean
Link	Specifies to link the object to another report, a website, an e-mail address or a Blob data type field. Click [...] in the value cell to set the link target. For details, see Binding a link to a label. Data type: String
Order By	Specifies the field by which to sort the data. Not available for aggregate field. Data type: String
Record Location	Specifies the calculation point for the properties which use formulas. Choose an option from the drop-down list. • **Default** - The values of the properties are calculated in the default location where they are located. • **Page Header** - The values of the properties are calculated in the page header. • **Page Footer** - The values of the properties are calculated in the page footer. Data type: Enumeration
Suppress	Specifies whether to show the object in the design area and in the report results. All formulas and calculations will be skipped if the property is set to true. Data type: Boolean
Suppress When No Records	Specifies whether to display the object in the report results when no record is returned in its parent data container. Data type: Boolean
Suppress When Null	If true and its value is null, the field will not be displayed. Data type: Boolean
Transfer Style	Specifies whether the style group of the primary report will be applied to the linked report if the object is linked to a report. Data type: Boolean
Value Delimiter	Specifies the separator for the data whose type is DBArray. The default value space means that the elements will be displayed in a horizontal line and separated by a space. Data type: String
Accessibility	
Abbr	It is mapped to the HTML attribute abbr. This attribute specifies an abbreviated form of the content of the object. The abbreviated text may be rendered by user agents when appropriate in place of the object's content. Data type: String
Axis	It is mapped to the HTML attribute axis. This attribute is used to place a cell into conceptual categories that can be considered to form axes in an n-dimensional space. The value of this attribute is a comma-separated list of category names. Data type: String
External AccessKey	It is mapped to the HTML attribute accesskey. This attribute specifies an access key to the object. Data type: String
External CSS Class Selector	Specifies a class selector to be applied to the object when exported as HTML. Enter a valid class name from the CSS file. Data type: String
External Dir	It is mapped to the HTML attribute dir. This attribute specifies the base direction of directionally neutral text in the object's content and attribute values. It also specifies the directionality of tables. Possible values: • LTR - Left-to-right text or table. • RTL - Right-to-left text or table. Data type: String
External ID	It is mapped to the HTML attribute id. This attribute specifies a name to the object, which must be unique in the report. Data type: String
External Style	It is mapped to the HTML attribute style. This attribute specifies style information for the object. Data type: String
External Tabindex	It is mapped to the HTML attribute tabindex. This attribute specifies the position of the object in the tabbing order for the current report. Enter an integer equal to or greater than 0 and less than 65535. Data type: Integer
External Title	It is mapped to the HTML attribute title. This attribute offers advisory information about the object. Data type: String
Headers	It is mapped to the HTML attribute headers. This attribute specifies the list of header cells that provide header information for the current data cell. The value of this attribute is a space-separated list of cell names; those cells must be named by setting their id attribute. Data type: String
HrefLang	It is mapped to the HTML attribute hreflang. This attribute specifies the base language of the resource designated by a link and may only be used when a link is specified. Data type: String
Language	It is mapped to the HTML attribute lang. This attribute specifies the base language of the object's attribute values and text content. Data type: String
LongDesc	It is mapped to the HTML attribute longdesc. This attribute specifies a link to a long description of the object. Data type: String
Scope	It is mapped to the HTML attribute scope. This attribute specifies the set of data cells for which the current header cell provides header information. Available values are: • **Row** - The current cell provides header information for the rest of the row that contains it. • **Column** - The current cell provides header information for the rest of the column that contains it. • **none** - The scope attribute will not be generated when exporting to HTML. Data type: String

Label

The header of a total row or column, the display name of a column or row field, and the display name of an aggregate field are represented as labels in the Report Structure panel of the Report Inspector. Their class types are CTTextField, CTDBTitleField, and CTLabelField respectively.

The properties of a label in a crosstab of a page report are:

Property Name	Description
General	
Class Type	Indicates the class type of the object. This property is read only.
Instance Name	Shows the instance name of the object. This property is read only.
Text Format	
Bold	Specifies whether to make the text bold. Data type: Boolean
Font Face	Specifies the font of the text. Choose an option from the drop-down list. Data type: Enumeration
Font Size	Specifies the font size of the text. Enter an integer value to change the size. Data type: Integer
Horizontal Alignment	Specifies the horizontal justification of the text in the object. Choose an option from the drop-down list. Data type: Enumeration
Ignore HTML Tag	Specifies whether or not to ignore the HTML tags when exporting the report results to HTML format. • **true** - The text will appear in the HTML file the same as that in JReport Designer (HTML tag elements in the text, if any, will not be parsed). • **false** - HTML tag elements in the text will be parsed when exporting the report results to HTML format. This property is useful for exporting the report results to HTML format. You may want to write the label contents in HTML tags in your report, and for the exported HTML files, the tags will be transferred as they are translated into HTML. Data type: Boolean
Italic	Specifies whether to make the text italic. Data type: Boolean
Maximum Width	Specifies the maximum width of the text displayed. Enter a numeric value to change the width. Data type: Float
Strikethrough	Specifies whether to add a strikethrough line to the text. Data type: Boolean
Text	Specifies the text in the label. Enter a string to display as the label text. Data type: String
Underline	Specifies whether to underline the text. Data type: Boolean
Vertical Alignment	Specifies the vertical justification of the text in the object. Choose an option from the drop-down list. Data type: Enumeration
Word Wrap	Specifies whether to wrap the text to the width. Data type: Boolean
Color	
Background	Specifies the background color of the object. Choose a color from the drop-down list, enter a hexadecimal RGB value (for example, 0xff0000), choose/create a formula or edit an expression that returns a color, or select Custom from the drop-down list to specify the color. Data type: String
Foreground	Specifies the foreground color of the object. Choose a color from the drop-down list, enter a hexadecimal RGB value (for example, 0xff0000), choose/create a formula or edit an expression that returns a color, or select Custom from the drop-down list to specify the color. Data type: String
CSS	
Class	Specifies a CSS class to be applied to the object which is a valid class in the CSS file. Data type: String
ID	Specifies the identifier of the object, which must be unique in the report. The ID property can be a style sheet selector. Data type: String
Style	The property can be used in two ways. • Specifies a style to be applied to the object. Choose a style from the drop-down list which is available when a Style Group has been selected at the report tab level and when there are styles in the Style Group that can be applied to the object, or type in the style name. • Specifies a CSS selector to be applied to the object. Enter a valid class name from the CSS file. Data type: String
Padding	
Bottom Padding	Specifies the space in inches or centimeters between the text and the bottom border of the object. Enter a numeric value to change the padding. Data type: Float
Left Padding	Specifies the space in inches or centimeters between the text and the left border of the object. Enter a numeric value to change the padding. Data type: Float
Right Padding	Specifies the space in inches or centimeters between the text and the right border of the object. Enter a numeric value to change the padding. Data type: Float
Top Padding	Specifies the space in inches or centimeters between the text and the top border of the object. Enter a numeric value to change the padding. Data type: Float
Border	
Border Color	Specifies the color of the border of the object. Choose a color from the drop-down list, enter a hexadecimal RGB value (for example, 0xCCAA00), choose/create a formula or edit an expression that returns a color, or select Custom from the drop-down list to specify the color. Data type: String
Border Thickness	Specifies the width of the border in inches or centimeters. Enter a numeric value to change the thickness. Data type: Float
Bottom Line	Specifies the line style of the bottom border of the object. Choose a style from the drop-down list. Data type: Enumeration
Left Line	Specifies the line style of the left border of the object. Choose a style from the drop-down list. Data type: Enumeration
Right Line	Specifies the line style of the right border of the object. Choose a style from the drop-down list. Data type: Enumeration
Shadow	Specifies whether to draw two shadowed borders, beneath and to the right of the object. Data type: Boolean
Shadow Color	Specifies the color of the shadow. Choose a color from the drop-down list, enter a hexadecimal RGB value (for example, 0xff0000), choose/create a formula or edit an expression that returns a color, or select Custom from the drop-down list to specify the color and fill effect. Data type: String
Top Line	Specifies the line style of the top border of the object. Choose a style from the drop-down list. Data type: Enumeration
Pattern	
Pattern Color	Specifies the color in which to draw a pattern to fill the object. Choose a color from the drop-down list, enter a hexadecimal RGB value (for example, 0xff0000), choose/create a formula or edit an expression that returns a color, or select Custom from the drop-down list to specify the color and fill effect. Data type: String
Pattern Style	Specifies the style of the pattern. Choose an option from the drop-down list. • **none** - No pattern will be applied to the object. • **50 %** - The object will be filled in the Pattern Color of 50%-transparency. • **horizontal** - The pattern will be made of horizontal lines of the Pattern Color. • **vertical** - The pattern will be made of vertical lines of the Pattern Color. • **grid** - The pattern will be an overlapping of horizontal and vertical lines of the Pattern Color. • **diagonal** - The pattern will be made of diagonal lines of the Pattern Color. Data type: Enumeration
Others	
Bind Column	When the label's Sortable or Filterable property is set to true, you need specify a field as the value so that the records of the field can be sorted or filtered via the proper button beside the label when viewing the report in Page Report Studio. Data type: String
Detail Target Frame	Specifies the target window or frame in which to display the detailed information. This property is enabled only when Go To Detail is set to true. Choose an option from the drop-down list. • **<Server Setting>** Loads the detailed information according to setting of the Pop Up New Window for Links option in the Customize Profile panel of the Profile dialog in JReport Server. • **New Window** Loads the detailed information into a new window. The window is not named. • **Whole Window** Loads the detailed information into the full browser window. • **Same Frame** Loads the detailed information into the same frame as the object. • **Parent Frame** Loads the detailed information into the parent frame of the frame in which the object is. • **Other Frame** Loads the detailed information into some other specified frame. If selected, you can also directly input the name of the frame you have defined in the text box. If the frame name does not exist, the detailed information will be loaded into a new window. Data type: String
Enable Hyperlink in Excel	Specifies whether to enable the link defined on the object when exporting the report to an Excel file. Data type: Boolean
Enable Hyperlink in HTML	Specifies whether to enable the link defined on the object when exporting the report to an HTML file. Data type: Boolean
Enable Hyperlink in PDF	Specifies whether to enable the link defined on the object when exporting the report to a PDF file. Data type: Boolean
Export to CSV	Specifies whether to include the object when exporting the report to CSV. Data type: Boolean.
Export to Excel	Specifies whether to include the object when exporting the report to Excel. Data type: Boolean
Export to Page Report Result	Specifies whether to include the object when exporting the report to Page Report Result. Data type: Boolean
Export to PDF	Specifies whether to include the object when exporting the report to PDF. Data type: Boolean
Filter Options	Specifies the filter-related options that will be displayed on the shortcut menu when right-clicking the object in Page Report Studio. If Default is selected, the filter options displayed will be determined by the setting in the Profile dialog in JReport Server (Profile > Customize Profile > Page Report Studio > Properties > Default > Filter Menu). For more information, refer to Setting filter options for a field. Data type: Integer
Filterable	Specifies whether the records can be filtered. If true, a filter button appears beside the label when viewing the report in Page Report Studio. You can use the button to filter the records based on a specific field's value. The field is specified via the Bind Column property. Data type: Boolean
Go to Detail	Specifies whether to show the detailed information when you click the object in Page Report Studio. This property is available only when the object is in the group header/footer panel of a banded object. For details about its usage, refer to Obtaining detailed information from a banded object. Data type: Boolean
Invisible	Specifies whether to show the object in the design area and in the report results. All formulas and calculations will still be performed if the property is set to true. Data type: Boolean
Link	Specifies to link the object to another report, a website, an e-mail address or a Blob data type field. Click in the value cell to set the link target. For details, see Binding a link to a label. Data type: String
Record Location	Specifies the calculation point for the properties which use formulas. Choose an option from the drop-down list. • **Default** - The values of the properties are calculated in the default location where they are located. • **Page Header** - The values of the properties are calculated in the page header. • **Page Footer** - The values of the properties are calculated in the page footer. Data type: Enumeration
Sortable	Specifies whether the records can be sorted. If true, a sort button appears beside the label when viewing the report in Page Report Studio. You can use the button to arrange the records of a specific field in ascending or descending order. The field is specified via the Bind Column property. Data type: Boolean
Suppress	Specifies whether to show the object in the design area and in the report results. All formulas and calculations will be skipped if the property is set to true. Data type: Boolean **Note:** When both the Invisible and Suppress properties of an object are set to true, Suppress has the higher priority.
Suppress Aggregate	Specifies whether to hide the Total row or column. Not applicable for CTLabelField. Data type: Boolean
Suppress When No Records	Specifies whether to display the object in the report results when no record is returned to its parent data container. Data type: Boolean
Transfer Style	Specifies whether the style group of the primary report will be applied to the linked report if the object is linked to a report. Data type: Boolean
Accessibility	
Abbr	It is mapped to the HTML attribute abbr. This attribute specifies an abbreviated form of the content of the object. The abbreviated text may be rendered by user agents when appropriate in place of the object's content. Data type: String
Axis	It is mapped to the HTML attribute axis. This attribute is used to place a cell into conceptual categories that can be considered to form axes in an n-dimensional space. The value of this attribute is a comma-separated list of category names. Data type: String
External AccessKey	It is mapped to the HTML attribute accesskey. This attribute specifies an access key to the object. Data type: String
External CSS Class Selector	Specifies a class selector to be applied to the object when exported as HTML. Enter a valid class name from the CSS file. Data type: String
External Dir	It is mapped to the HTML attribute dir. This attribute specifies the base direction of directionally neutral text in the object's content and attribute values. It also specifies the directionality of tables. Possible values: • LTR - Left-to-right text or table. • RTL - Right-to-left text or table. Data type: String
External ID	It is mapped to the HTML attribute id. This attribute specifies a name to the object, which must be unique in the report. Data type: String
External Style	It is mapped to the HTML attribute style. This attribute specifies style information for the object. Data type: String
External Tabindex	It is mapped to the HTML attribute tabindex. This attribute specifies the position of the object in the tabbing order for the current report. Enter an integer equal to or greater than 0 and less than 65535. Data type: Integer
External Title	It is mapped to the HTML attribute title. This attribute offers advisory information about the object. Data type: String
Headers	It is mapped to the HTML attribute headers. This attribute specifies the list of header cells that provide header information for the current data cell. The value of this attribute is a space-separated list of cell names; those cells must be named by setting their *id* attribute. Data type: String
HrefLang	It is mapped to the HTML attribute hreflang. This attribute specifies the base language of the resource designated by a link and may only be used when a link is specified. Data type: String
Language	It is mapped to the HTML attribute lang. This attribute specifies the base language of the object's attribute values and text content. Data type: String
LongDesc	It is mapped to the HTML attribute longdesc. This attribute specifies a link to a long description of the object. Data type: String
Scope	It is mapped to the HTML attribute scope. This attribute specifies the set of data cells for which the current header cell provides header information. Available values are: • **Row** - The current cell provides header information for the rest of the row that contains it. • **Column** - The current cell provides header information for the rest of the column that contains it. • **none** - The scope attribute will not be generated when exporting to HTML. Data type: String

Dataset

The properties of a dataset in a page report are:

Property Name	Description
General	
Class Type	Indicates the class type of the object. This property is read only.
Data Name	Shows the name of the query or business view in use. It can be null. When it is null, the available resources for the dataset are parameters and constant level formulas in the data source.
Instance Name	Shows the instance name of the object. This property is read only.
Is Query	It is true when the data set is based on a query or false when based on a business view.
Others	
Data Driver	Specifies the cached query result file as the data resource. Usually used for support purpose. The input format for cached query result file is as follows: `jrquery:/jet.universe.resultfile.UResultFileResultSet;Fullpath_of_cached_query_result` Data type: String
Data Source Name	Specifies the name of the data source. Data type: String
Maximum Records	Specifies the maximum number of records you want to display for all of the data containers using this dataset. The default is to display all the records. Data type: Integer
Query Editor	Edits the dataset if it is based on a query. Click ... to display the Query Editor to edit the query as required.
Security	
Function	Specifies a formula to control the record level security. Data type: String
Record Security	Specifies the record level security for the dataset. Click ... to open the Record Level Security Information dialog in which to set the security settings.
Security Policy Name	Specifies the name of the data source security imported from the catalog. Choose a security policy from the value drop-down list. Data type: String
Data Buffer	
Maximum Page Number	Specifies the maximum number of pages in the data buffer. Data type: Integer
Records per Page	Specifies the number of records in each page in the data buffer. Data type: Integer

Data Driver

This property is usually for support purpose. When you have problems with your report, we need to run it and reproduce the problem so as to fix it. Sometimes, the database is too large to be sent. If this property is correctly set, then JReport engine will choose that query file to view the report instead of using JDBC driver or User Data Source to run the report.

To generate the cached query result, in the Catalog Browser, right-click the query of your report, click **Create Cached Query Result**, you can specify the folder and file name with or without extension (.txt).

Data Buffer

You can improve report performance by setting data buffer size. By default, the size of one data buffer is 2M. When data exceeds the default size, they will be written on the disk, which will slow down the data processing. However, you may have enough memory which is much larger than the default buffer size, so that you can use the memory resource efficiently to achieve better performance. JReport enables you to set the data buffer size by specifying the two properties: Records per Page and Maximum Page Number.

When a buffer is defined, JReport divides it into many pages and allocates records to be stored in the pages. Data will be transferred with the unit of page. The data buffer size is determined by values of maximum number of pages in the buffer and the number of records in every page. The default maximum page number in the data buffer is 2048, page size is 1k(1024 bytes).

The value of Maximum Page Number should be changed with different datasets. Generally speaking, it can be 1 ~ 50.

To view the data buffer information of a report and its subreports, click **View** > **Data Buffer Information** on the menu bar. For details, see Data Buffer Information dialog.

Notes:

- The properties should be set according to different databases, namely, different databases may have different configurations of data buffer.
- If the report runs in a different environment, the properties should be specified again.
- If the report contains subreports, the data buffer for subreports should also be configured to improve performance.
- As tested, if enough physical memory is available, and you know the total number of records in the dataset, then you can set Records per Page approximate to the total number of records to achieve better performance. Maximum Page Number is related to the report, and should not be very large.
- The size of data buffer should be less than the available physical memory.
- The Heap Size should be set when the data buffer is quite large.

DBField

The properties of a DBField in a page report are:

Property Name	Description
General	
Column	Shows the column name of the field in the raw database. This property is read only.
DBField Name	Shows the mapping name of the field in the catalog. This property is read only.
Data Type	Shows the data type of the field. This property is read only.
Field Type	Shows what kind of field it is. This property is read only.
Instance Name	Shows the instance name of the object. This property is read only.
Query	Shows the query/UDS name. This property is read only.
Table	Shows the table name of the field in the raw database. This property is read only.
Geometry	
Height	Specifies the height of the object, in inches or centimeters. Enter a numeric value to change the height. Data type: Float Data type: Float
Width	Specifies the width of the object, in inches or centimeters. Enter a numeric value to change the width. Data type: Float Data type: Float
X	Specifies the horizontal coordinate of the top left corner of the object, relative to its parent container. This property is ignored if the Position property is set to static. Enter a numeric value to change the position in inches or centimeters. Data type: Float Data type: Float
Y	Specifies the vertical coordinate of the top left corner of the object, relative to its parent container. This property is ignored if the Position property is set to static. Enter a numeric value to change the position in inches or centimeters. Data type: Float Data type: Float
Color	
Background	Specifies the background color of the object. Choose a color from the drop-down list, enter a hexadecimal RGB value (for example, 0xff0000), choose/create a formula or edit an expression that returns a color, or select Custom from the drop-down list to specify the color. Data type: String Data type: String
Foreground	Specifies the foreground color of the object. Choose a color from the drop-down list, enter a hexadecimal RGB value (for example, 0xff0000), choose/create a formula or edit an expression that returns a color, or select Custom from the drop-down list to specify the color. Data type: String Data type: String
CSS	
Class	Specifies a CSS class to be applied to the object which is a valid class in the CSS file. Data type: String Data type: String
ID	Specifies the identifier of the object, which must be unique in the report. The ID property can be a style sheet selector. Data type: String Data type: String
Style	The property is used in two ways: • Specifies a style to be applied to the object. Choose a style from the drop-down list or enter a valid Style Group has been selected at the report tab level and when there are styles in the Style Group that can be applied to the object, or type in the style name. • Specifies a CSS selector to be applied to the object. Enter a valid class name from the CSS file. Data type: String
Excel	
Column Index	Specifies the X coordinate of the object relative to its parent container when exported to Excel or CSV, measured in cells. The Columned property at the report tab level must be set to true for this property to take effect. Data type: Integer
Row Index	Specifies the Y coordinate of the object relative to its parent container when exported to Excel or CSV, measured in cells. The Columned property at the report tab level must be set to true for this property to take effect. Data type: Integer
Padding	
Bottom Padding	Specifies the space in inches or centimeters between the text and the bottom border of the object. Enter a numeric value to change the padding. Data type: Float Data type: Float
Left Padding	Specifies the space in inches or centimeters between the text and the left border of the object. Enter a numeric value to change the padding. Data type: Float Data type: Float
Right Padding	Specifies the space in inches or centimeters between the text and the right border of the object. Enter a numeric value to change the padding. Data type: Float Data type: Float
Top Padding	Specifies the space in inches or centimeters between the text and the top border of the object. Enter a numeric value to change the padding. Data type: Float Data type: Float
Border	
Border Color	Specifies the color of the border of the object. Choose a color from the drop-down list, enter a hexadecimal RGB value (for example, 0xCCAA00), choose/create a formula or edit an expression that returns a color, or select Custom from the drop-down list to specify the color and thickness. Data type: String Data type: String
Border Thickness	Specifies the width of the border in inches or centimeters. Enter a numeric value to change the thickness. Data type: Float Data type: Float
Bottom Line	Specifies the line style of the bottom border of the object. Choose a style from the drop-down list. Data type: Enumeration Data type: Enumeration
Left Line	Specifies the line style of the left border of the object. Choose a style from the drop-down list. Data type: Enumeration Data type: Enumeration
Right Line	Specifies the line style of the right border of the object. Choose a style from the drop-down list. Data type: Enumeration Data type: Enumeration
Top Line	Specifies the line style of the top border of the object. Choose a style from the drop-down list. Data type: Enumeration Data type: Enumeration
Pattern	
Pattern Color	Specifies the color in which to draw a pattern to fill the object. Choose a color from the drop-down list, enter a hexadecimal RGB value (for example, 0xff0000), choose/create a formula or edit an expression that returns a color, or select Custom from the drop-down list to specify the color and fill effect. Data type: String Data type: String
Pattern Style	Specifies the style of the pattern. Choose an option from the drop-down list: • **none** - No pattern will be applied to the object. • **50%** - The object will be filled in the Pattern Color of 50% transparency. • **horizontal** - The pattern will be made of horizontal lines of the Pattern Color. • **vertical** - The pattern will be made of vertical lines of the Pattern Color. • **grid** - The pattern will be an overlapping of horizontal and vertical lines of the Pattern Color. • **diagonal** - The pattern will be made of diagonal lines of the Pattern Color. Data type: Enumeration
Text Format	
Auto Fit	Specifies whether to adjust the width and height of the object according to the contents. Data type: Boolean Data type: Boolean
Bold	Specifies whether to make the text bold. Data type: Boolean Data type: Boolean
Font Face	Specifies the font of the text. Choose an option from the drop-down list. Data type: Enumeration Data type: Enumeration
Font Size	Specifies the font size of the text. Enter an integer value to change the size. Data type: Integer Data type: Integer
Format	Specifies the display format of the text in the report result. It varies with data type and can be manually defined. Data type: String Data type: String **Note:** For the BigDecimal type, to avoid precision loss, you should enter a prefix JRD in the formatting the format pattern.
Horizontal Alignment	Specifies the horizontal justification of the text in the object. Choose an option from the drop-down list. Data type: Enumeration Data type: Enumeration
Ignore HTML Tag	Specifies whether or not to ignore the HTML tags when exporting the report results to HTML format. • **true** - The text will appear in the HTML file the same as that in JReport Designer (HTML tag elements in the text, if any, will not be parsed). • **false** - HTML tag elements in the text will be parsed when exporting the report results to HTML format. This property is useful for exporting the report results to HTML format. You may want to write the label contents in HTML tags in your report, and for the exported HTML files, the HTML tags will be preserved as they are translated into HTML. Data type: Boolean Data type: Boolean
Italic	Specifies whether to make the text italic. Data type: Boolean Data type: Boolean
Maximum Width	Specifies the maximum width of the text displayed. Enter a numeric value to change the width. This property often works with the Auto Fit property: if Auto Fit = true and Maximum Width is not equal to 0, the text will extend until the width is this value. Data type: Float Data type: Float
Strikethrough	Specifies whether to add a strikethrough line to the text. Data type: Boolean Data type: Boolean
Underline	Specifies whether to underline the text. Data type: Boolean Data type: Boolean
Vertical Alignment	Specifies the vertical justification of the text in the object. Choose an option from the drop-down list. Data type: Enumeration Data type: Enumeration
Word Wrap	Specifies whether to wrap the text to the width. Data type: Boolean Data type: Boolean
Others	
Cache Value	Specifies whether to cache the value of the field instead of obtaining it repeatedly. Data type: Boolean Data type: Boolean
Column Name	Subroutines the current field with another field by selecting from the drop-down list. To make the property editable, uncheck Forbid changing column in the Options dialog **File > Options > Panel > Forbid changing column**. Data type: String Data type: String
Data Mapping File	Specifies the data mapping file to the object for NLS use. For example, if the data mapping file name is Category_de_DE.properties, set the value as Category. Change the language and locale part. Data type: String Data type: String
Detail Report	Specifies the detail report that the object is linked to. Click in the value cell to set the detail report. For details, refer to Linking to a detail report. Data type: Object Data type: Object
Detail Target Frame	Specifies the target window or frame in which to display the detailed information. This property is enabled only when Go To Detail is set to true. Choose an option from the drop-down list. Data type: Enumeration Data type: Enumeration • **Server Setting>** Loads the detailed information according to setting of the Pop Up New Window for Links option in the Customize Profile panel of the Profile dialog in JReport Server. • **New Window** Loads the detailed information into a new window. The window is not named. • **Whole Window** Loads the detailed information into the full browser window. • **Same Frame** Loads the detailed information into the same frame as the object. • **Parent Frame** Loads the detailed information into the parent frame of the frame in which the object is. • **Other Frame** Loads the detailed information into some other specified frame. If selected, you can also directly input the name of the frame you have defined in the text box. If the frame name does not exist, the detailed information will be loaded into a new window. Data type: String Data type: String
Display Null	Specifies the string to be displayed if the field value is null. Data type: String Data type: String
Enable Hyperlink in Excel	Specifies whether to enable the link defined on the object when exporting the report to an Excel file. Data type: Boolean Data type: Boolean
Enable Hyperlink in HTML	Specifies whether to enable the link defined on the object when exporting the report to an HTML file. Data type: Boolean Data type: Boolean
Enable Hyperlink in PDF	Specifies whether to enable the link defined on the object when exporting the report to a PDF file. Data type: Boolean Data type: Boolean
Export to Applet	Specifies whether to include the object when exporting the report to Applet. This property is available only when the object is displayed as a web control, barcode or rank. Data type: Boolean Data type: Boolean
Export to CSV	Specifies whether to include the object when exporting the report to CSV. Not available when the object is displayed as a rank, image or image button. Data type: Boolean Data type: Boolean
Export to Excel	Specifies whether to include the object when exporting the report to Excel. Data type: Boolean Data type: Boolean
Export to HTML	Specifies whether to include the object when exporting the report to HTML. This property is available only when the object is displayed as a web control, barcode or rank. Data type: Boolean Data type: Boolean
Export to PDF	Specifies whether to include the object when exporting the report to PDF. This property is available only when the object is displayed as a web control, barcode or rank. Data type: Boolean Data type: Boolean
Export to PostScript	Specifies whether to include the object when exporting the report to HTML. This property is available only when the object is displayed as a web control, barcode or rank. Data type: Boolean Data type: Boolean
Export to Report Result	Specifies whether to show the object in the design area and in the report results. All formulas and calculations will still be performed. Data type: Boolean Data type: Boolean
Export to RTF	Specifies whether to include the object when exporting the report to RTF. This property is available only when the object is displayed as a web control, barcode or rank. Data type: Boolean Data type: Boolean
Export to Text	Specifies whether to include the object when exporting the report to Text. This property is available only when the object is displayed as a barcode or web control except for image button. Data type: Boolean Data type: Boolean
Export to XML	Specifies whether to include the object when exporting the report to XML. This property is available only when the object is displayed as a barcode or web control except for image button. Data type: Boolean Data type: Boolean
Filter Options	Specifies the filter related options that will be displayed on the shortcut menu when right-clicking the object in Page Report Studio. If Default is selected, the filter options displayed are decided by the setting in the Profile dialog in JReport Server (Profile > Customize Profile > Page Report Studio > Properties > Default > Filter Menu). For more information, refer to Setting filter options for a field. Data type: Integer Data type: Integer
Go to Detail	Specifies whether to show the detailed information when you click the object in Page Report Viewer. This property is available only when the object is in the group header/footer panel or the detail. For details about its usage, refer to Obtaining detailed information from a banded object. Data type: Boolean Data type: Boolean
Invisible	Specifies whether to show the object in the design area and in the report results. All formulas and calculations will still be performed if the property is set to true. Data type: Boolean Data type: Boolean
Link	Specifies to link the object to another report, a website, an e-mail address or a Blob type field. Click in the value cell to set the link target. For details, see Binding a link to a label. Data type: String Data type: String
Logic Column	Specifies whether to show the object in the next visible table cell in the same row when the column which holds the object is hidden. Choose an option from the drop-down list. Data type: Enumeration **Notes:** • The next visible table cell should be completely empty, that is, the cell should not hold any content including blank space. • When the property for several objects in the same row is set to next visible column, the objects will be displayed in the next visible cell only if the object in the rightmost column will be shown in the next visible cell.
Position	Specifies the position of the object. Choose an option from the drop-down list: • **static** - The object is positioned at the location in which it is inserted. If selected, the object is positioned relative to the previous object in the same container. The position of the objects preceded by preceding objects in the same container. • **relative** - The object is positioned at an offset from the location at which it is inserted. The offset is determined by the X and Y property values. • **absolute** - The object will be located at the position specified by dragging and dropping or by setting the X and Y property values. The position of the object is not affected by preceding objects in the same container. Data type: Enumeration
Record Location	Specifies the calculation point for the properties which use formulas. Choose an option from the drop-down list: • **Page** - Values of the properties are calculated in the default location where they are inserted. • **Page Header** - The values of the properties are calculated in the page header. • **Page Footer** - The values of the properties are calculated in the page footer. Data type: Enumeration
Suppress	Specifies whether to show the object in the design area and in the report results. All formulas and calculations will be skipped if the property is set to true. Data type: Boolean Data type: Boolean **Note:** When both the Invisible and Suppress properties of an object are set to true, Suppress has the higher priority.
Suppress When No Records	Specifies whether to display the object in the report results when no record is returned from the data source. Data type: Boolean Data type: Boolean If true and its value is null, the field will not be displayed.
Suppress When Null	Specifies whether the style group of the primary report will be applied to the linked report if the object is linked to a report. Data type: Boolean Data type: Boolean
Transfer Style	Specifies the separator for the data whose type is DBArray. The default value space means that the elements will be displayed in a horizontal line and separated by a space. Data type: String Data type: String
TOC	
Anchor Display Value	Specifies a string or formula to display for the TOC entry for this object in the TOC Browser. By default it is the value of the specified object. The TOC Anchor property must be set to true for this property to take effect. Data type: String Data type: String
TOC Anchor	Specifies whether to include the object in the TOC Browser for the report. Data type: Boolean Data type: Boolean
Accessibility	
External Accessibly	It is mapped to the HTML attribute accessibly. This attribute specifies an access key to the object. Data type: String Data type: String
External CSS Class Selector	Specifies a class selector to be applied to the object when exported as HTML. Enter a valid class name from the CSS file. Data type: String Data type: String
External Dir	It is mapped to the HTML attribute dir. This attribute specifies the base direction of directionally neutral text in the object's content and attribute values. It also specifies the directionality of tables. Possible values: • LTR - Left-to-right text or table. • RTL - Right-to-left text or table. Data type: Enumeration Data type: Enumeration
External ID	It is mapped to the HTML attribute id. This attribute specifies a name to the object, which must be unique in the report. Data type: String Data type: String
External Style	It is mapped to the HTML attribute style. This attribute specifies style information for the object. Data type: String Data type: String
External Tabindex	It is mapped to the HTML attribute tabindex. This attribute specifies the position of the object in the tabbing order for the current report. Enter an integer equal to or greater than 0 and less than 65535. Data type: Integer Data type: Integer
External Title	It is mapped to the HTML attribute title. This attribute offers advisory information about the object. Data type: String Data type: String
HrefLang	It is mapped to the HTML attribute hreflang. This attribute specifies the base language of the resource designated by a link and may only be used when a link is specified. Data type: String Data type: String
Language	It is mapped to the HTML attribute lang. This attribute specifies the base language of the object's attribute values and text content. Data type: String Data type: String
LongDesc	It is mapped to the HTML attribute longdesc. This attribute specifies a link to a long description of the object. Data type: String Data type: String

Format

This property indicates what will be displayed in the report result. In other words, no matter in what kind of format the data is stored in the database, you can specify the format in which to display the result. For example, for a number-type field about product price, we have the following formats for you to choose:

#.###
#,###.##
$#,##0.00
...

You select $#,##0.00, and then a value displays as $6.05. You can enter manually as $#,##0.000, then the value is displayed as $6.050.

Drawing object

In JReport, you can insert five kinds of drawing objects into a report. They are: arcs, boxes, lines, ovals and round boxes, the properties of which are listed as follows:

- Arc
- Box
- Line
- Oval
- Round box

Arc

The properties of an arc object in a page report are:

Property Name	Description
General	
Class Type	Indicates the class type of the object. This property is read only.
Instance Name	Shows the instance name of the object. This property is read only.
Geometry	
Bottom Attach Pos X	Specifies the horizontal coordinate of the bottom right point of the focus box in a banded panel. A drawing object can cross panels and the coordinate indicates the relative position in the involved panel. Data type: Float
Bottom Attach Pos Y	Specifies the vertical coordinate of the bottom right point of the focus box in a banded panel. A drawing object can cross panels and the coordinate indicates the relative position in the involved panel. Data type: Float
Top Attach Pos X	Specifies the horizontal coordinate of the top left point of the focus box in a banded panel. A drawing object can cross panels and the coordinate indicates the relative position in the involved panel. Data type: Float
Top Attach Pos Y	Specifies the vertical coordinate of the top left point of the focus box in a banded panel. A drawing object can cross panels and the coordinate indicates the relative position in the involved panel. Data type: Float
Color	
Background	Specifies the background color of the object. Choose a color from the drop-down list, enter a hexadecimal RGB value (for example, 0xff0000), or select Custom from the drop-down list to specify the color. Data type: String
CSS	
Class	Specifies a CSS class to be applied to the object which is a valid class in the CSS file. Data type: String
Style	The property can be used in two ways. • Specifies a style to be applied to the object. Choose a style from the drop-down list which is available when a Style Group has been selected at the report tab level and when there are styles in the Style Group that can be applied to the object, or type in the style name. • Specifies a CSS selector to be applied to the object. Enter a valid class name from the CSS file. Data type: String
Others	
Export to Page Report Result	Specifies whether to include the object when exporting the report to Page Report Result. Data type: Boolean
Invisible	Specifies whether to show the object in the design area and in the report results. All formulas and calculations will still be performed. Data type: Boolean
Record Location	Specifies the calculation point for the properties which use formulas. Choose an option from the drop-down list. • **Default** - The values of the properties are calculated in the default location where they are located. • **Page Header** - The values of the properties are calculated in the page header. • **Page Footer** - The values of the properties are calculated in the page footer. Data type: Enumeration
Suppress When No Records	Specifies whether to display the object in the report results when no record is returned to its parent data container. Data type: Boolean
Arc Property	
Arc Angle	Specifies the angle of the arc, in degrees. The value ranges from 0 to 360. As it gets closer to 360, the arc begins to take on an elliptical shape. Data type: Float
Border Color	Specifies the color of the object's border. Choose a color from the drop-down list, enter a hexadecimal RGB value (0xCCAA00), or select Custom from the drop-down list to specify the color. Data type: String
Border Style	Specifies the line style of the object's border. Choose an option from the drop-down list. Data type: Enumeration
Border Thickness	Specifies the width of the object's border, in inches or centimeters. Enter a numeric value to change the thickness. Data type: Float
Start Angle	Specifies the start angle of the arc, in degrees. The value ranges from 0 to 360. At 0, the vertex of the arc starts at 12:15 clock position. As it gets closer to 360, the arc travels in a counterclockwise motion. Data type: Float
Accessibility	
External CSS Class Selector	Specifies a class selector to be applied to the object when exported as HTML. Enter a valid class name from the CSS file. Data type: String
External ID	It is mapped to the HTML attribute *id*. This attribute specifies a name to the object, which must be unique in the report. Data type: String
External Style	It is mapped to the HTML attribute *style*. This attribute specifies style information for the object. Data type: String
External Title	It is mapped to the HTML attribute *title*. This attribute offers advisory information about the object. Data type: String
Language	It is mapped to the HTML attribute *lang*. This attribute specifies the base language of the object's attribute values and text content. Data type: String

Note: When a report is exported to an HTML/RTF/Excel format file, arcs in it cannot be shown. There is no such problem for PDF, PS, and other output formats.

Box

The properties of a box object in a page report are:

Property Name	Description
General	
Class Type	Indicates the class type of the object. This property is read only.
Instance Name	Shows the instance name of the object. This property is read only.
Geometry	
Bottom Attach Pos X	Specifies the horizontal coordinate of the bottom right point of the box in a banded panel. A drawing object can cross panels and the coordinate indicates the relative position in the involved panel. Data type: Float
Bottom Attach Pos Y	Specifies the vertical coordinate of the bottom right point of the box in a banded panel. A drawing object can cross panels and the coordinate indicates the relative position in the involved panel. Data type: Float
Top Attach Pos X	Specifies the horizontal coordinate of the top left point of the box in a banded panel. A drawing object can cross panels and the coordinate indicates the relative position in the involved panel. Data type: Float
Top Attach Pos Y	Specifies the vertical coordinate of the top left point of the box in a banded panel. A drawing object can cross panels and the coordinate indicates the relative position in the involved panel. Data type: Float
Color	
Background	Specifies the background color of the object. Choose a color from the drop-down list, enter a hexadecimal RGB value (for example, 0xff0000), or select Custom from the drop-down list to specify the color. Data type: String
CSS	
Class	Specifies a CSS class to be applied to the object which is a valid class in the CSS file. Data type: String
Style	The property can be used in two ways. • Specifies a style to be applied to the object. Choose a style from the drop-down list which is available when a Style Group has been selected at the report tab level and when there are styles in the Style Group that can be applied to the object, or type in the style name. • Specifies a CSS selector to be applied to the object. Enter a valid class name from the CSS file. Data type: String
Others	
Export to Page Report Result	Specifies whether to include the object when exporting the report to Page Report Result. Data type: Boolean
Export to Excel	Specifies whether to include the object when exporting the report to Excel. Data type: Boolean
Invisible	Specifies whether to show the object in the design area and in the report results. All formulas and calculations will still be performed. Data type: Boolean
Record Location	Specifies the calculation point for the properties which use formulas. Choose an option from the drop-down list. • **Default** - The values of the properties are calculated in the default location where they are located. • **Page Header** - The values of the properties are calculated in the page header. • **Page Footer** - The values of the properties are calculated in the page footer. Data type: Enumeration
Suppress When No Records	Specifies whether to display the object in the report results when no record is returned to its parent data container. Data type: Boolean
Excel	
Bottom Attach Column	The four Excel properties, Top Attach Column, Top Attach Row, Bottom Attach Column and Bottom Attach Row, work together to control the position of a drawing object in the exported Excel file. Bottom Attach Column specifies the X coordinate of the lower right corner of the drawing object in the exported Excel file, measured in cells. The Columned property at the report tab level must be set to true for this property to take effect. Data type: Float
Bottom Attach Row	The four Excel properties, Top Attach Column, Top Attach Row, Bottom Attach Column and Bottom Attach Row, work together to control the position of a drawing object in the exported Excel file. Bottom Attach Row specifies the Y coordinate of the lower right corner of the drawing object in the exported Excel file, measured in cells. The Columned property at the report tab level must be set to true for this property to take effect. Data type: Float
Top Attach Column	The four Excel properties, Top Attach Column, Top Attach Row, Bottom Attach Column and Bottom Attach Row, work together to control the position of a drawing object in the exported Excel file. Top Attach Column specifies the X coordinate of the upper left corner of the drawing object in the exported Excel file, measured in cells. The Columned property at the report tab level must be set to true for this property to take effect. Data type: Float
Top Attach Row	The four Excel properties, Top Attach Column, Top Attach Row, Bottom Attach Column and Bottom Attach Row, work together to control the position of a drawing object in the exported Excel file. Top Attach Row specifies the Y coordinate of the upper left corner of the drawing object in the exported Excel file, measured in cells. The Columned property at the report tab level must be set to true for this property to take effect. Data type: Float
Box Property	
Border Color	Specifies the color of the border of the object. Choose a color from the drop-down list, enter a hexadecimal RGB value (0xCCAA00), or select Custom from the drop-down list to specify the color. Data type: String
Border Style	Specifies the line style of the box border. Choose an option from the drop-down list. Data type: Enumeration
Border Thickness	Specifies the width of the object's border, in inches or centimeters. Enter a numeric value to change the thickness. Data type: Float
Accessibility	
External CSS Class Selector	Specifies a class selector to be applied to the object when exported as HTML. Enter a valid class name from the CSS file. Data type: String
External ID	It is mapped to the HTML attribute *id*. This attribute specifies a name to the object, which must be unique in the report. Data type: String
External Style	It is mapped to the HTML attribute *style*. This attribute specifies style information for the object. Data type: String
External Title	It is mapped to the HTML attribute *title*. This attribute offers advisory information about the object. Data type: String
Language	It is mapped to the HTML attribute *lang*. This attribute specifies the base language of the object's attribute values and text content. Data type: String

Line

The properties of a line object in a page report are:

Property Name	Description
General	
Class Type	Indicates the class type of the object. This property is read only.
Instance Name	Shows the instance name of the object. This property is read only.
Geometry	
Bottom Attach Pos X	Specifies the horizontal coordinate of the bottom right point of the line in a banded panel. A drawing object can cross panels and the coordinate indicates the relative position in the involved panel. Data type: Float
Bottom Attach Pos Y	Specifies the vertical coordinate of the bottom right point of the line in a banded panel. A drawing object can cross panels and the coordinate indicates the relative position in the involved panel. Data type: Float
Top Attach Pos X	Specifies the horizontal coordinate of the top left point of the line in a banded panel. A drawing object can cross panels and the coordinate indicates the relative position in the involved panel. Data type: Float
Top Attach Pos Y	Specifies the vertical coordinate of the top left point of the line in a banded panel. A drawing object can cross panels and the coordinate indicates the relative position in the involved panel. Data type: Float
CSS	
Class	Specifies a CSS class to be applied to the object which is a valid class in the CSS file. Data type: String
Style	The property can be used in two ways. • Specifies a style to be applied to the object. Choose a style from the drop-down list which is available when a Style Group has been selected at the report tab level and when there are styles in the Style Group that can be applied to the object, or type in the style name. • Specifies a CSS selector to be applied to the object. Enter a valid class name from the CSS file. Data type: String
Others	
Export to Page Report Result	Specifies whether to include the object when exporting the report to Page Report Result. Data type: Boolean
Export to Excel	Specifies whether to include the object when exporting the report to Excel. Data type: Boolean
Invisible	Specifies whether to show the object in the design area and in the report results. All formulas and calculations will still be performed. Data type: Boolean
Record Location	Specifies the calculation point for the properties which use formulas. Choose an option from the drop-down list. • **Default** - The values of the properties are calculated in the default location where they are located. • **Page Header** - The values of the properties are calculated in the page header. • **Page Footer** - The values of the properties are calculated in the page footer. Data type: Enumeration
Suppress When No Records	Specifies whether to display the object in the report results when no record is returned to its parent data container. Data type: Boolean
Excel	
Bottom Attach Column	The four Excel properties, Top Attach Column, Top Attach Row, Bottom Attach Column and Bottom Attach Row, work together to control the position of a drawing object in the exported Excel file. Bottom Attach Column specifies the X coordinate of the lower right corner of the drawing object in the exported Excel file, measured in cells. The Columned property at the report tab level must be set to true for this property to take effect. Data type: Float
Bottom Attach Row	The four Excel properties, Top Attach Column, Top Attach Row, Bottom Attach Column and Bottom Attach Row, work together to control the position of a drawing object in the exported Excel file. Bottom Attach Row specifies the Y coordinate of the lower right corner of the drawing object in the exported Excel file, measured in cells. The Columned property at the report tab level must be set to true for this property to take effect. Data type: Float
Top Attach Column	The four Excel properties, Top Attach Column, Top Attach Row, Bottom Attach Column and Bottom Attach Row, work together to control the position of a drawing object in the exported Excel file. Top Attach Column specifies the X coordinate of the upper left corner of the drawing object in the exported Excel file, measured in cells. The Columned property at the report tab level must be set to true for this property to take effect. Data type: Float
Top Attach Row	The four Excel properties, Top Attach Column, Top Attach Row, Bottom Attach Column and Bottom Attach Row, work together to control the position of a drawing object in the exported Excel file. Top Attach Row specifies the Y coordinate of the upper left corner of the drawing object in the exported Excel file, measured in cells. The Columned property at the report tab level must be set to true for this property to take effect. Data type: Float
Line Property	
Line Color	Specifies the color of the line. Choose a color from the drop-down list, enter a hexadecimal RGB value (for example, 0xff0000), or select Custom from the drop-down list to specify the color. Data type: String
Line Style	Specifies the style of the line. Choose an option from the drop-down list. Data type: Enumeration
Line Thickness	Specifies the thickness of the line. Enter a numeric value to change the thickness. Data type: Float
Accessibility	
External CSS Class Selector	Specifies a class selector to be applied to the object when exported as HTML. Enter a valid class name from the CSS file. Data type: String
External ID	It is mapped to the HTML attribute *id*. This attribute specifies a name to the object, which must be unique in the report. Data type: String
External Style	It is mapped to the HTML attribute *style*. This attribute specifies style information for the object. Data type: String
External Title	It is mapped to the HTML attribute *title*. This attribute offers advisory information about the object. Data type: String
Language	It is mapped to the HTML attribute *lang*. This attribute specifies the base language of the object's attribute values and text content. Data type: String

Note: When a report is exported to HTML files, the lines in it should be vertical or horizontal lines. Otherwise, they cannot be shown properly in the HTML file. You should make sure that Top Attach Pos X = Bottom Attach Pos X or Top Attach Pos Y= Bottom Attach Pos Y. There is no such problem for PDF, PS, RTF and other output formats.

Oval

The properties of an oval object in a page report are:

Property Name	Description
General	
Class Type	Indicates the class type of the object. This property is read only.
Instance Name	Shows the instance name of the object. This property is read only.
Geometry	
Bottom Attach Pos X	Specifies the horizontal coordinate of the bottom right point of the focus box in a banded panel. A drawing object can cross panels and the coordinate indicates the relative position in the involved panel. Data type: Float
Bottom Attach Pos Y	Specifies the vertical coordinate of the bottom right point of the focus box in a banded panel. A drawing object can cross panels and the coordinate indicates the relative position in the involved panel. Data type: Float
Top Attach Pos X	Specifies the horizontal coordinate of the top left point of the focus box in a banded panel. A drawing object can cross panels and the coordinate indicates the relative position in the involved panel. Data type: Float
Top Attach Pos Y	Specifies the vertical coordinate of the top left point of the focus box in a banded panel. A drawing object can cross panels and the coordinate indicates the relative position in the involved panel. Data type: Float
Color	
Background	Specifies the background color of the object. Choose a color from the drop-down list, enter a hexadecimal RGB value (for example, 0xff0000), or select Custom from the drop-down list to specify the color. Data type: String
CSS	
Class	Specifies a CSS class to be applied to the object which is a valid class in the CSS file. Data type: String
Style	The property can be used in two ways. • Specifies a style to be applied to the object. Choose a style from the drop-down list which is available when a Style Group has been selected at the report tab level and when there are styles in the Style Group that can be applied to the object, or type in the style name. • Specifies a CSS selector to be applied to the object. Enter a valid class name from the CSS file. Data type: String
Others	
Export to Page Report Result	Specifies whether to include the object when exporting the report to Page Report Result. Data type: Boolean
Export to Excel	Specifies whether to include the object when exporting the report to Excel. Data type: Boolean
Invisible	Specifies whether to show the object in the design area and in the report results. All formulas and calculations will still be performed. Data type: Boolean
Record Location	Specifies the calculation point for the properties which use formulas. Choose an option from the drop-down list. • **Default** - The values of the properties are calculated in the default location where they are located. • **Page Header** - The values of the properties are calculated in the page header. • **Page Footer** - The values of the properties are calculated in the page footer. Data type: Enumeration
Suppress When No Records	Specifies whether to display the object in the report results when no record is returned to its parent data container. Data type: Boolean
Excel	
Bottom Attach Column	The four Excel properties, Top Attach Column, Top Attach Row, Bottom Attach Column and Bottom Attach Row, work together to control the position of a drawing object in the exported Excel file. Bottom Attach Column specifies the X coordinate of the lower right corner of the drawing object in the exported Excel file, measured in cells. The Columned property at the report tab level must be set to true for this property to take effect. Data type: Float
Bottom Attach Row	The four Excel properties, Top Attach Column, Top Attach Row, Bottom Attach Column and Bottom Attach Row, work together to control the position of a drawing object in the exported Excel file. Bottom Attach Row specifies the Y coordinate of the lower right corner of the drawing object in the exported Excel file, measured in cells. The Columned property at the report tab level must be set to true for this property to take effect. Data type: Float
Top Attach Column	The four Excel properties, Top Attach Column, Top Attach Row, Bottom Attach Column and Bottom Attach Row, work together to control the position of a drawing object in the exported Excel file. Top Attach Column specifies the X coordinate of the upper left corner of the drawing object in the exported Excel file, measured in cells. The Columned property at the report tab level must be set to true for this property to take effect. Data type: Float
Top Attach Row	The four Excel properties, Top Attach Column, Top Attach Row, Bottom Attach Column and Bottom Attach Row, work together to control the position of a drawing object in the exported Excel file. Top Attach Row specifies the Y coordinate of the upper left corner of the drawing object in the exported Excel file, measured in cells. The Columned property at the report tab level must be set to true for this property to take effect. Data type: Float
Oval Property	
Border Color	Specifies the color of the object's border. Choose a color from the drop-down list, enter a hexadecimal RGB value (0xCCAA00), or select Custom from the drop-down list to specify the color. Data type: String
Border Style	Specifies the line style of the object's border. Choose an option from the drop-down list. Data type: Enumeration
Border Thickness	Specifies the width of the object's border, in inches or centimeters. Enter a numeric value to change the thickness. Data type: Float
Accessibility	
External CSS Class Selector	Specifies a class selector to be applied to the object when exported as HTML. Enter a valid class name from the CSS file. Data type: String
External ID	It is mapped to the HTML attribute *id*. This attribute specifies a name to the object, which must be unique in the report. Data type: String
External Style	It is mapped to the HTML attribute *style*. This attribute specifies style information for the object. Data type: String
External Title	It is mapped to the HTML attribute *title*. This attribute offers advisory information about the object. Data type: String
Language	It is mapped to the HTML attribute *lang*. This attribute specifies the base language of the object's attribute values and text content. Data type: String

Note: When a report is exported to an HTML or RTF format file, ovals in it cannot be shown. There is no such problem for PDF, PS and other output formats.

Round box

The properties of a round box in a page report are:

Property Name	Description
General	
Class Type	Indicates the class type of the object. This property is read only.
Instance Name	Shows the instance name of the object. This property is read only.
Geometry	
Bottom Attach Pos X	Specifies the horizontal coordinate of the bottom right point of the focus box in a banded panel. A drawing object can cross panels and the coordinate indicates the relative position in the involved panel. Data type: Float
Bottom Attach Pos Y	Specifies the vertical coordinate of the bottom right point of the focus box in a banded panel. A drawing object can cross panels and the coordinate indicates the relative position in the involved panel. Data type: Float
Top Attach Pos X	Specifies the horizontal coordinate of the top left point of the focus box in a banded panel. A drawing object can cross panels and the coordinate indicates the relative position in the involved panel. Data type: Float
Top Attach Pos Y	Specifies the vertical coordinate of the top left point of the focus box in a banded panel. A drawing object can cross panels and the coordinate indicates the relative position in the involved panel. Data type: Float
Color	
Background	Specifies the background color of the object. Choose a color from the drop-down list, enter a hexadecimal RGB value (for example, 0xff0000), or select Custom from the drop-down list to specify the color. Data type: String
CSS	
Class	Specifies a CSS class to be applied to the object which is a valid class in the CSS file. Data type: String
Style	The property can be used in two ways. • Specifies a style to be applied to the object. Choose a style from the drop-down list which is available when a Style Group has been selected at the report tab level and when there are styles in the Style Group that can be applied to the object, or type in the style name. • Specifies a CSS selector to be applied to the object. Enter a valid class name from the CSS file. Data type: String
Others	
Export to Page Report Result	Specifies whether to include the object when exporting the report to Page Report Result. Data type: Boolean
Export to Excel	Specifies whether to include the object when exporting the report to Excel. Data type: Boolean
Invisible	Specifies whether to show the object in the design area and in the report results. All formulas and calculations will still be performed. Data type: Boolean
Record Location	Specifies the calculation point for the properties which use formulas. Choose an option from the drop-down list. • **Default** - The values of the properties are calculated in the default location where they are located. • **Page Header** - The values of the properties are calculated in the page header. • **Page Footer** - The values of the properties are calculated in the page footer. Data type: Enumeration
Suppress When No Records	Specifies whether to display the object in the report results when no record is returned to its parent data container. Data type: Boolean
Excel	
Bottom Attach Column	The four Excel properties, Top Attach Column, Top Attach Row, Bottom Attach Column and Bottom Attach Row, work together to control the position of a drawing object in the exported Excel file. Bottom Attach Column specifies the X coordinate of the lower right corner of the drawing object in the exported Excel file, measured in cells. The Columned property at the report tab level must be set to true for this property to take effect. Data type: Float
Bottom Attach Row	The four Excel properties, Top Attach Column, Top Attach Row, Bottom Attach Column and Bottom Attach Row, work together to control the position of a drawing object in the exported Excel file. Bottom Attach Row specifies the Y coordinate of the lower right corner of the drawing object in the exported Excel file, measured in cells. The Columned property at the report tab level must be set to true for this property to take effect. Data type: Float
Top Attach Column	The four Excel properties, Top Attach Column, Top Attach Row, Bottom Attach Column and Bottom Attach Row, work together to control the position of a drawing object in the exported Excel file. Top Attach Column specifies the X coordinate of the upper left corner of the drawing object in the exported Excel file, measured in cells. The Columned property at the report tab level must be set to true for this property to take effect. Data type: Float
Top Attach Row	The four Excel properties, Top Attach Column, Top Attach Row, Bottom Attach Column and Bottom Attach Row, work together to control the position of a drawing object in the exported Excel file. Top Attach Row specifies the Y coordinate of the upper left corner of the drawing object in the exported Excel file, measured in cells. The Columned property at the report tab level must be set to true for this property to take effect. Data type: Float
Round Box Property	
Border Color	Specifies the color of the object's border. Choose a color from the drop-down list, enter a hexadecimal RGB value (0xCCAA00), or select Custom from the drop-down list to specify the color. Data type: String
Border Style	Specifies the line style of the object's border. Choose an option from the drop-down list. Data type: Enumeration
Border Thickness	Specifies the width of the object's border, in inches or centimeters. Enter a numeric value to change the thickness. Data type: Float
Corner Factor	Specifies the relative radius of the object's corners. Data type: Float
Accessibility	
External CSS Class Selector	Specifies a class selector to be applied to the object when exported as HTML. Enter a valid class name from the CSS file. Data type: String
External ID	It is mapped to the HTML attribute *id*. This attribute specifies a name to the object, which must be unique in the report. Data type: String
External Style	It is mapped to the HTML attribute *style*. This attribute specifies style information for the object. Data type: String
External Title	It is mapped to the HTML attribute *title*. This attribute offers advisory information about the object. Data type: String
Language	It is mapped to the HTML attribute *lang*. This attribute specifies the base language of the object's attribute values and text content. Data type: String

Note: When a report is exported to an HTML or PS format file, round boxes in it cannot be shown. There is no such problem for PDF, RTF and other output formats.

Form

The properties of a form in a page report are:

Property Name	Description
General	
Class Type	Indicates the class type of the object. This property is read only.
Instance Name	Shows the instance name of the object. This property is read only.
CSS	
Class	Specifies a CSS class to be applied to the object which is a valid class in the CSS file. Data type: String
Style	The property can be used in two ways. • Specifies a style to be applied to the object. Choose a style from the drop-down list which is available when a Style Group has been selected at the report tab level and when there are styles in the Style Group that can be applied to the object, or type in the style name. • Specifies a CSS selector to be applied to the object. Enter a valid class name from the CSS file. Data type: String
Form	
Disabled	Specifies whether the user can interact with the form. Data type: Boolean
Form Action	Specifies the URL to which the form content is sent for processing. Data type: String
Form ID	Specifies the string identifying the form. Data type: String
Form Method	Specifies how to send the form data to the server. Choose an option from the drop-down list. • **get** - Appends the arguments to the Form Action URL and opens it as if it were an anchor. • **post** - Sends the data through an HTTP post transaction. Data type: Enumeration
Form Name	Specifies the name of the form. Data type: String
Form Target	Specifies the window or frame at which to target contents. Choose an option from the drop-down list. • **default** - Loads the linked document into the window in which the link is clicked (the active window). Same as "self". • **blank** - Loads the linked document into a new blank window. This window is not named. • **media** - Loads the linked document into the HTML content area of the Media Bar. Available in Microsoft Internet Explorer 6 or later. • **parent** - Loads the linked document into the immediate parent window of the active window. If there is no parent window, the active window is used. • **search** - Loads the linked document into the browser search page. Available in Internet Explorer 5 or later. • **self** - Loads the linked document into the window in which the link is clicked (the active window). • **top** - Loads the linked document into the topmost window. Data type: Enumeration
Accessibility	
External CSS Class Selector	Specifies a class selector to be applied to the object when exported as HTML. Enter a valid class name from the CSS file. Data type: String
External Style	It is mapped to the HTML attribute *style*. This attribute specifies style information for the object. Data type: String
External Title	It is mapped to the HTML attribute *title*. This attribute offers advisory information about the object. Data type: String
Language	It is mapped to the HTML attribute *lang*. This attribute specifies the base language of the object's attribute values and text content. Data type: String

Formula field

The properties of a formula field in a page report are:

Property Name	Description
General	
Data Type	Shows the data type of the formula. This property is read only.
Expression	Shows the expression of the formula. This property is read only.
Field Type	Shows what kind of field it is. This property is read only.
Formula Editor	Edits the formula. Click ... to display the Formula Editor to edit the formula as required.
Formula Name	Shows the name of the formula. This property is read only.
Owner Name	Shows the instance name of the object. This property is read only.
Geometry	
Height	Specifies the height of the object, in inches or centimeters. Enter a numeric value to change the height. Data type: Float
Width	Specifies the width of the object, in inches or centimeters. Enter a numeric value to change the width. Data type: Float
X	Specifies the horizontal coordinate of the top left corner of the object, relative to its parent container. This property is ignored if the Position property is set to static. Enter a numeric value to change the position in inches or centimeters. Data type: Float
Y	Specifies the vertical coordinate of the top left corner of the object, relative to its parent container. This property is ignored if the Position property is set to static. Enter a numeric value to change the position in inches or centimeters. Data type: Float
Color	
Background	Specifies the background color of the object. Choose a color from the drop-down list, enter a hexadecimal RGB value (for example, 0xff0000), choose/create a formula or edit an expression that returns a color, or select Custom from the drop-down list to specify the color. Data type: String
Foreground	Specifies the foreground color of the object. Choose a color from the drop-down list, enter a hexadecimal RGB value (for example, 0xff0000), choose/create a formula or edit an expression that returns a color, or select Custom from the drop-down list to specify the color. Data type: String
CSS	
Class	Specifies a CSS class to be applied to the object which is a valid class in the CSS file. Data type: String
ID	Specifies the identifier of the object, which must be unique in the report. The ID property can be a style sheet selector. Data type: String
Style	The property can be used in two ways. • Specifies a style to be applied to the object. Choose a style from the drop-down list which is available when a Style Group has been selected at the report tab level and when there are styles in the Style Group that can be applied to the object, or type in the style name. • Specifies a CSS selector to be applied to the object. Enter a valid class name from the CSS file. Data type: String
Excel	
Column Index	Specifies the X coordinate of the object relative to its parent container when exported to Excel or CSV, measured in cells. The Columned property at the report tab level must be set to true for this property to take effect. Data type: Integer
Row Index	Specifies the Y coordinate of the object relative to its parent container when exported to Excel or CSV, measured in cells. The Columned property at the report tab level must be set to true for this property to take effect. Data type: Integer
Padding	
Bottom Padding	Specifies the space in inches or centimeters between the text and the bottom border of the object. Enter a numeric value to change the padding. Data type: Float
Left Padding	Specifies the space in inches or centimeters between the text and the left border of the object. Enter a numeric value to change the padding. Data type: Float
Right Padding	Specifies the space in inches or centimeters between the text and the right border of the object. Enter a numeric value to change the padding. Data type: Float
Top Padding	Specifies the space in inches or centimeters between the text and the top border of the object. Enter a numeric value to change the padding. Data type: Float
Border	
Border Color	Specifies the color of the border of the object. Choose a color from the drop-down list, enter a hexadecimal RGB value (for example, 0xCCAA00), choose/create a formula or edit an expression that returns a color, or select Custom from the drop-down list to specify the color. Data type: String
Border Thickness	Specifies the width of the border in inches or centimeters. Enter a numeric value to change the thickness. Data type: Float
Bottom Line	Specifies the line style of the bottom border of the object. Choose a style from the drop-down list. Data type: Enumeration
Left Line	Specifies the line style of the left border of the object. Choose a style from the drop-down list. Data type: Enumeration
Right Line	Specifies the line style of the right border of the object. Choose a style from the drop-down list. Data type: Enumeration
Shadow	Specifies whether to draw two shadowed borders, beneath and to the right of the object. Data type: Boolean
Shadow Color	Specifies the color of the shadow. Choose a color from the drop-down list, enter a hexadecimal RGB value (for example, 0xff0000), choose/create a formula or edit an expression that returns a color, or select Custom from the drop-down list to specify the color and tint effect. Data type: String
Top Line	Specifies the line style of the top border of the object. Choose a style from the drop-down list. Data type: Enumeration
Pattern	
Pattern Color	Specifies the color in which to draw a pattern to fill the object. Choose a color from the drop-down list, enter a hexadecimal RGB value (for example, 0xff0000), choose/create a formula or edit an expression that returns a color, or select Custom from the drop-down list to specify the color and tint effect. Data type: String
Pattern Style	Specifies the style of the pattern. Choose an option from the drop-down list. • **none** - No pattern will be applied to the object. • **50%** - The object will be filled in the Pattern Color of 50%-transparency. • **horizontal** - The pattern will be made of horizontal lines of the Pattern Color. • **vertical** - The pattern will be made of vertical lines of the Pattern Color. • **grid** - The pattern will be an overlapping of horizontal and vertical lines of the Pattern Color. • **diagonal** - The pattern will be made of diagonal lines of the Pattern Color. Data type: Enumeration
Text Format	
Auto Fit	Specifies whether to adjust the width and height of the object according to the contents. Data type: Boolean
Bold	Specifies whether to make the text bold. Data type: Boolean
Font Face	Specifies the font of the text. Choose an option from the drop-down list. Data type: Enumeration
Font Size	Specifies the font size of the text. Enter an integer value to change the size. Data type: Integer
Format	Specifies the display format of the text in the report result. It varies with data type and can be manually defined. Data type: String **Note:** For the BigDecimal type, to avoid precision loss, you should enter a prefix JRD when setting the format property value.
Horizontal Alignment	Specifies the horizontal justification of the text in the object. Choose an option from the drop-down list. Data type: Enumeration
Ignore HTML Tag	Specifies whether or not to ignore the HTML tags when exporting the report results to HTML format. • **true** - The text will appear in the HTML file the same as that in iReport Designer (HTML tag elements in the text, if any, will not be parsed). • **false** - HTML tag elements in the text will be parsed when exporting the report results to HTML format. This property is useful for exporting the report results to HTML format. You may want to write the label contents in HTML tags in your report, and for the exported HTML files, the tags will be transferred as they are translated into HTML. Data type: Boolean
Italic	Specifies whether to make the text italic. Data type: Boolean
Maximum Width	Specifies the maximum width of the text displayed. Enter a numeric value to change the width. This property often works with the Auto Fit property. If Auto Fit = true and Maximum Width is not equal to 0, the text will extend until the width is this value. Data type: Float
Strikethrough	Specifies whether to add a strikethrough line to the text. Data type: Boolean
Underline	Specifies whether to underline the text. Data type: Boolean
Vertical Alignment	Specifies the vertical justification of the text in the object. Choose an option from the drop-down list. Data type: Enumeration
Word Wrap	Specifies whether to wrap the text to the width. Data type: Boolean
Others	
Cache Value	Specifies whether to cache the value of the field instead of obtaining it repeatedly. Data type: Boolean
Column Name	Substitutes the current field with another field by selecting from the drop-down list. To make the property editable, uncheck Forbid changing column in the Options dialog (**File** > **Options** > **Panel** > **Forbid changing column**). Data type: String
Data Mapping File	Specifies the data mapping file to object for NLS use. For example, if the data mapping file name is Category_de_DE.properties, set the value as Category (without the language and locale part).
Detail Report	Specifies the detail report that the object is linked to. Click ... in the value cell to set the detail report. For details, refer to Linking to a detail report.
Detail Target Frame	Specifies the target window or frame in which to display the detailed information. This property is enabled only when Go To Detail is set to true. Choose an option from the drop-down list. • **<Server Setting>** Loads the detailed information according to setting of the Pop Up New Window for Links option in the Customize Profile panel of the Profile dialog in JReport Server. • **New Window** Loads the detailed information into a new window. The window is not named. • **Whole Window** Loads the detailed information into the full browser window. • **Same Frame** Loads the detailed information into the same frame as the object. • **Parent Frame** Loads the detailed information into the parent frame of the frame in which the object is. • **Other Frame** Loads the detailed information into some other specified frame. If selected, you can enter the name of the frame into which the detailed information is to be loaded. If the frame name does not exist, the detailed information will be loaded into a new window. Data type: String
Display Null	Specifies the string to be displayed if the field value is null. Data type: String
Enable Hyperlink in Excel	Specifies whether to enable the link defined on the object when exporting the report to an Excel file. Data type: Boolean
Enable Hyperlink in HTML	Specifies whether to enable the link defined on the object when exporting the report to an HTML file. Data type: Boolean
Enable Hyperlink in PDF	Specifies whether to enable the link defined on the object when exporting the report to a PDF file. Data type: Boolean
Export to CSV	Specifies whether to include the object when exporting the report to CSV. Data type: Boolean
Export to Page Report Result	Specifies whether to include the object when exporting the report to Page Report Result. Data type: Boolean
Export to Excel	Specifies whether to include the object when exporting the report to Excel. Data type: Boolean
Export to PDF	Specifies whether to include the object when exporting the report to PDF. This property is available only when the object is displayed as a web control. Data type: Boolean
Filter Options	Specifies the filter-related options that will be displayed on the shortcut menu when right-clicking the object in Page Report Studio. If Default is selected, the filter options displayed are decided by the setting in the Profile dialog in JReport Server (Profile > Customize Profile > Page Report Studio > Properties > Default > Filter Menu). For more information, refer to Setting filter options for a field.
Go to Detail	Specifies whether to show the detailed information when you click the object in Page Report Studio. This property is available only when the object is in the group header/ footer panel of a banded object. For details about its usage, refer to Obtaining detailed information from a banded object. Data type: Boolean
Invisible	Specifies whether to show the object in the design area and in the report results. All formulas and calculations will still be performed if the property is set to true. Data type: Boolean
Link	Specifies to link the object to another report, a website, an e-mail address or a Blob data type field. Click ... in the value cell to set the link target. For details, see Binding a link to a label. Data type: String
Logic Column	Specifies whether to show the object in the next visible table cell in the same row when the column which holds the object is hidden. Choose an option from the drop-down list. Data type: Enumeration **Notes:** • The next visible table cell should be completely empty, that is, the cell should not hold any content including blank space. • When the property for several objects in the same row is set to next visible column, and the columns holding these objects are all hidden, then only the object in the rightmost column will be shown in the next visible cell.
Position	Specifies the position of the object. Choose an option from the drop-down list. • **static** - The object is positioned at the location in which it is inserted. If selected, the X, Y and other position-related properties will be hidden or disabled. The position of the object is affected by preceding objects in the same container. • **relative** - The object is positioned as an offset to the position at which it is inserted. The offset is determined by the X and Y property values. • **absolute** - The object will be located at the position specified by dragging and dropping or setting the X and Y property values. The position of the object is not affected by preceding objects in the same container. Data type: Enumeration
Record Location	Specifies the calculation point for the properties which use formulas. Choose an option from the drop-down list. • **Default** - The values of the properties are calculated in the default location where they are located. • **Page Header** - The values of the properties are calculated in the page header. • **Page Footer** - The values of the properties are calculated in the page footer. Data type: Enumeration
Suppress	Specifies whether to show the object in the design area and in the report results. All formulas and calculations will be skipped if the property is set to true. Data type: Boolean **Note:** When both the Invisible and Suppress properties of an object are set to true, Suppress has the higher priority.
Suppress When No Records	Specifies whether to display the object in the report results when no record is returned in its parent data container. Data type: Boolean
Suppress When Null	If true and its value is null, the field will not be displayed. Data type: Boolean
Transfer Style	Specifies whether the style group of the primary report will be applied to the linked report if the object is linked to a report. Data type: Boolean
Value Delimiter	Specifies the separator for the data whose type is DBArray. The default value space means that the elements will be displayed in a horizontal line and separated by a space. Data type: String
TOC	
Anchor Display Value	Specifies a string or formula to display for the TOC entry for this object in the TOC Browser. By default it is the value of the specified object. The TOC Anchor property must be set to true for this property to take effect. Data type: String
TOC Anchor	Specifies whether to include the object in the TOC Browser for the report. Data type: Boolean
Accessibility	
External AccessKey	It is mapped to the HTML attribute accesskey. This attribute specifies an access key to the object. Data type: String
External CSS Class Selector	Specifies a class selector to be applied to the object when exported as HTML. Enter a valid class name from the CSS file. Data type: String
External Dir	It is mapped to the HTML attribute dir. This attribute specifies the base direction of directionally neutral text in the object's content and attribute values. It also specifies the directionality of tables. Possible values: • LTR - Left-to-right text or table. • RTL - Right-to-left text or table. Data type: String
External ID	It is mapped to the HTML attribute id. This attribute specifies a name to the object, which must be unique in the report. Data type: String
External Style	It is mapped to the HTML attribute style. This attribute specifies style information for the object. Data type: String
External Tabindex	It is mapped to the HTML attribute tabindex. This attribute specifies the position of the object in the tabbing order for the current report. Enter an integer equal to or greater than 0 and less than 65535. Data type: Integer
External Title	It is mapped to the HTML attribute title. This attribute offers advisory information about the object. Data type: String
HrefLang	It is mapped to the HTML attribute hreflang. This attribute specifies the base language of the resource designated by a link and may only be used when a link is specified. Data type: String
Language	It is mapped to the HTML attribute lang. This attribute specifies the base language of the object's attribute values and text content. Data type: String
LongDesc	It is mapped to the HTML attribute longdesc. This attribute specifies a link to a long description of the object. Data type: String

Geographic map

The properties of a geographic map in a page report are:

Property Name	Description
General	
Class Type	Indicates the class type of the object. This property is read only.
Data Inherit	Indicates whether the dataset for this object is inherited from another object. This property is read only.
Dataset	Indicates the dataset used by the object. This property is read only.
Instance Name	Shows the instance name of the object. This property is read only.
Geometry	
Height	Specifies the height of the object, in inches or centimeters. Enter a numeric value to change the height. Data type: Float
Width	Specifies the width of the object, in inches or centimeters. Enter a numeric value to change the width. Data type: Float
X	Specifies the horizontal coordinate of the top left corner of the object, relative to its parent container. This property is ignored if the Position property is set to static. Enter a numeric value to change the position in inches or centimeters. Data type: Float
Y	Specifies the vertical coordinate of the top left corner of the object, relative to its parent container. This property is ignored if the Position property is set to static. Enter a numeric value to change the position in inches or centimeters. Data type: Float
Others	
Default for Filter	Specifies whether to show the data component as the default component in the Apply to drop-down list of the Filter dialog in Page Report Studio. Data type: Boolean **Note:** In a page report tab, only one data component's Default for Filter property can be set to true.
Export to Applet	Specifies whether to include the object when exporting the report to Applet. Data type: Boolean
Export to Excel	Specifies whether to include the object when exporting the report to Excel. Data type: Boolean
Export to HTML	Specifies whether to include the object when exporting the report to HTML. Data type: Boolean
Export to PDF	Specifies whether to include the object when exporting the report to PDF. Data type: Boolean
Export to PostScript	Specifies whether to include the object when exporting the report to PostScript. Data type: Boolean
Export to Report Result	Specifies whether to include the object when exporting the report to Report Result. Data type: Boolean
Export to RTF	Specifies whether to include the object when exporting the report to RTF. Data type: Boolean
Invisible	Specifies whether to show the object in the design area and in the report results. All formulas and calculations will still be performed if the property is set to true. Data type: Boolean
Invisible for Filter Dialogs	Specifies whether to show the data component in the Apply to drop-down list of the Filter dialog in Page Report Studio. This property cannot be edited when Default for Filter is set to true. Data type: Boolean
Position	Specifies the position of the object. Choose an option from the drop-down list. • **static** - The object is positioned at the location in which it is inserted. If selected, the X, Y and other position-related properties will be hidden or disabled. The position of the object is affected by preceding objects in the same container. • **absolute** - The object will be located at the position specified by dragging and dropping or by setting its X and Y property values. The position of the object is not affected by preceding objects in the same container. Data type: Enumeration
Suppress	Specifies whether to show the object in the design area and in the report results. All formulas and calculations will be skipped if the property is set to true. Data type: Boolean **Note:** When both the Invisible and Suppress properties of an object are set to true, Suppress has the higher priority.
Suppress When No Records	Specifies whether to display the object in the report results when no record is returned to its parent data container. Data type: Boolean
Excel	
Column Index	Specifies the X coordinate of the object relative to its parent container when exported to Excel or CSV, measured in cells. The Columned property at the report tab level must be set to true for this property to take effect. Data type: Integer
Row Index	Specifies the Y coordinate of the object relative to its parent container when exported to Excel or CSV, measured in cells. The Columned property at the report tab level must be set to true for this property to take effect. Data type: Integer
TOC	
Anchor Display Value	Specifies a string or formula to display for the TOC entry for this object in the TOC Browser. By default it is the value of the specified object. The TOC Anchor property must be set to true for this property to take effect. Data type: String
TOC Anchor	Specifies whether to include the object in the TOC Browser for the report. Data type: Boolean
Accessibility	
External CSS Class Selector	Specifies a class selector to be applied to the object when exported as HTML. Enter a valid class name from the CSS file. Data type: String
External ID	It is mapped to the HTML attribute *id*. This attribute specifies a name to the object, which must be unique in the report. Data type: String
External Style	It is mapped to the HTML attribute *style*. This attribute specifies style information for the object. Data type: String
Language	It is mapped to the HTML attribute *lang*. This attribute specifies the base language of the object's attribute values and text content. Data type: String

A geographic map in a page report has a child object, the properties of which will be shown in the following topic:

- Geographic map marker

Geographic map marker

The properties of a geographic map marker in a page report are:

Property Name	Description
General	
Class Type	Indicates the class type of the object. This property is read only.
Group By	Shows the group field that the object is based on. If null, the object is grouped based on the whole dataset. This property is read only.
Instance Name	Shows the instance name of the object. This property is read only.
Others	
Invisible	Specifies whether to show the object in the design area and in the report results. All formulas and calculations will still be performed if the property is set to true. Data type: Boolean

Image

The properties of an image in a page report are:

Property Name	Description
General	
Class Type	Indicates the class type of the object. This property is read only.
Instance Name	Shows the instance name of the object. This property is read only.
Geometry	
Height	Specifies the height of the object, in inches or centimeters. Enter a numeric value to change the height. Data type: Float
Width	Specifies the width of the object, in inches or centimeters. Enter a numeric value to change the width. Data type: Float
X	Specifies the horizontal coordinate of the top left corner of the object, relative to its parent container. This property is ignored if the Position property is set to static. Enter a numeric value to change the position in inches or centimeters. Data type: Float
Y	Specifies the vertical coordinate of the top left corner of the object, relative to its parent container. This property is ignored if the Position property is set to static. Enter a numeric value to change the position in inches or centimeters. Data type: Float
CSS	
Class	Specifies a CSS class to be applied to the object which is a valid class in the CSS file. Data type: String
ID	Specifies the identifier of the object, which must be unique in the report. The ID property can be a style sheet selector. Data type: String
Style	The property can be used in two ways. • Specifies a style to be applied to the object. Choose a style from the drop-down list which is available when a Style Group has been selected at the report tab level and which includes styles in the Style Group that can be applied to the object, or type in the style name. • Specifies a CSS selector to be applied to the object. Enter a valid class name from the CSS file. Data type: String
Others	
Detail Report	Specifies the detail report that the object is linked to. Click [...] in the value cell to set the detail report. For details, refer to Linking to a detail report. Data type: String
Detail Target Frame	Specifies the target window or frame in which to display the detailed information. This property is enabled only when Go To Detail is set to true. Choose an option from the drop-down list. • **<Server Setting>** Loads the detailed information according to setting of the Pop Up New Window for Links option in the Customize Profile panel of the Profile dialog in jReport Server. • **New Window** Loads the detailed information into a new window. The window is not named. • **Whole Window** Loads the detailed information into the full browser window. • **Same Frame** Loads the detailed information into the same frame as the object. • **Parent Frame** Loads the detailed information into the parent frame of the frame in which the object is. • **Other Frame** Loads the detailed information into some other specified frame. If selected, you can also directly input the name of the frame you have defined in the text box. If the frame name does not exist, the detailed information will be loaded into a new window. Data type: String
Export to Applet	Specifies whether to include the object when exporting the report to Applet. Data type: Boolean
Export to Excel	Specifies whether to include the object when exporting the report to Excel. Data type: Boolean
Export to HTML	Specifies whether to include the object when exporting the report to HTML. Data type: Boolean
Export to PDF	Specifies whether to include the object when exporting the report to PDF. Data type: Boolean
Export to PostScript	Specifies whether to include the object when exporting the report to PostScript. Data type: Boolean
Export to Report Result	Specifies whether to include the object when exporting the report to Report Result. Data type: Boolean
Export to RTF	Specifies whether to include the object when exporting the report to RTF. Data type: Boolean
Go to Detail	Specifies whether to show the detailed information when you click the object in Page Report Studio. This property is available only when the object is in the group header/footer panel of a banded object. For details about its usage, refer to Obtaining detailed information from a banded object. Data type: Boolean
Horizontal Alignment	Specifies the horizontal alignment of the image in the container. Choose an option from the drop-down list. Data type: Enumeration
Invisible	Specifies whether to show the object in the design area and in the report results. All formulas and calculations will still be performed if the property is set to true. Data type: Boolean
Link	Specifies to link the object to another report, a website, an e-mail address or a Blob data type field. Click [...] in the value cell to set the link target. For details, see Binding a link to a label. Data type: String
Logic Column	Specifies whether to show the object in the next visible table cell in the same row when the column which holds the object is hidden. Choose an option from the drop-down list. Data type: Enumeration **Notes:** • The next visible table cell should be completely empty, that is, the cell should not hold any content including blank space. • When the property for several objects in the same row is set to next visible column, and the column holding these objects are all hidden, then only the object in the rightmost column will be shown in the next visible cell.
Position	Specifies the position of the object. Choose an option from the drop-down list. • **static** - The object is positioned at the location in which it is inserted. If selected, the X, Y and other position-related properties will be hidden or disabled. The position of the object is affected by preceding objects in the same container. • **relative** - The object is positioned at an offset to the position at which it is inserted. The offset is determined by the X and Y property values. • **absolute** - The object will be located at the position specified by dragging and dropping or by setting its X and Y property values. The position of the object is not affected by preceding objects in the same container. Data type: Enumeration
Record Location	Specifies the calculation point for the properties which use formulas. Choose an option from the drop-down list. • **Default** - The values of the properties are calculated in the default location where they are located. • **Page Header** - The values of the properties are calculated in the page header. • **Page Footer** - The values of the properties are calculated in the page footer. Data type: Enumeration
Suppress	Specifies whether to show the object in the design area and in the report results. All formulas and calculations will be skipped if the property is set to true. Data type: Boolean **Note:** When both the Invisible and Suppress properties of an object are set to true, Suppress has the higher priority.
Suppress When No Records	Specifies whether to display the object in the report results when no record is returned to its parent data container. Data type: Boolean
Transfer Style	Specifies whether the style group of the primary report will be applied to the linked report if the object is linked to a report. Data type: Boolean
Vertical Alignment	Specifies the vertical alignment of the image in the container. Choose an option from the drop-down list. Data type: Enumeration
Excel	
Column Index	Specifies the X coordinate of the object relative to its parent container when exported to Excel or CSV, measured in cells. The Columned property at the report tab level must be set to true for this property to take effect. Data type: Integer
Column Number	Specifies the number of columns which will be the object's width in the exported Excel file. The Columned property at the report tab level must be set to true for this property to take effect. Data type: Integer
Row Index	Specifies the Y coordinate of the object relative to its parent container when exported to Excel or CSV, measured in cells. The Columned property at the report tab level must be set to true for this property to take effect. Data type: Integer
Row Number	Specifies the number of rows which will be the object's height in the exported Excel file. The Columned property at the report tab level must be set to true for this property to take effect. Data type: Integer
Image Property	
Alternate Text	Specifies the text that will be displayed instead if the image cannot be displayed. Data type: String
Disabled	Specifies whether to disable the actions on the image. Data type: Boolean
Image Name	Specifies the file name of the image. The image can be one on your local file system or retrieved by using a URL. Data type: String
Maximum Scaling Ratio	Specifies the maximum ratio up to which the image can be scaled. Data type: Float
Name	Specifies the name of the image. Data type: String
Rotation	Specifies the angle at which to rotate the image, in degrees. The following is the meaning of different values: • 0 - No rotation. • positive value - Rotates the image clockwise. • negative value - Rotates the image anticlockwise. Data type: Float
Scaling Mode	Specifies the scaling mode of the image, which controls the image behavior when you try to adjust the image field size. Choose an option from the drop-down list. • **actual size** - The image will remain its original size. If the image field size is smaller than the image size, part of the image will be cut off. • **fit image** - The image size will be automatically adjusted to fill up the image field, with its original perspective remained but under the limitation of Maximum Scaling Ratio. • **fit width** - The image will be automatically adjusted to fit the width of the image field but under the limitation of Maximum Scaling Ratio. • **fit height** - The image will be automatically adjusted to fit the height of the image field but under the limitation of Maximum Scaling Ratio. • **customize** - The image size will be automatically adjusted to fit the image field, regardless of the Maximum Scaling Ratio's value. Data type: Enumeration
Tab Index	Specifies the index that defines the tab order for the image. Data type: Integer
Title	Specifies the tip information about the image, which will be displayed when you hover the mouse pointer over the image in HTML result or Page Report Studio. Data type: String
Value	Specifies the initial value of the image. Data type: String
Event	
On Blur	Specifies the action when the object loses input focus. Data type: String
On Click	Specifies the action when the user clicks the left mouse button on the object. Data type: String
On Double Click	Specifies the action when the user double-clicks the object. Data type: String
On Focus	Specifies the action when the object receives focus. Data type: String
On Key Down	Specifies the action when the user presses a key. Data type: String
On Key Press	Specifies the action when the user presses an alphanumeric key. Data type: String
On Key Up	Specifies the action when the user releases a key. Data type: String
On Mouse Down	Specifies the action when the user clicks the object with either mouse button. Data type: String
On Mouse Move	Specifies the action when the user moves the mouse over the object. Data type: String
On Mouse Out	Specifies the action when the user moves the mouse pointer outside the boundaries of the object. Data type: String
On Mouse Over	Specifies the action when the user moves the mouse pointer into the object. Data type: String
On Mouse Up	Specifies the action when the user releases a mouse button while the mouse is over the object. Data type: String
TOC	
Anchor Display Value	Specifies a string or formula to display for the TOC entry for this object in the TOC Browser. By default it is the value of the specified object. The TOC Anchor property must be set to true for this property to take effect. Data type: String
TOC Anchor	Specifies whether to include the object in the TOC Browser for the report. Data type: Boolean
Accessibility	
External AccessKey	It is mapped to the HTML attribute accesskey. This attribute specifies an access key to the object. Data type: String
External CSS Class Selector	Specifies a class selector to be applied to the object when exported as HTML. Enter a valid class name from the CSS file. Data type: String
External ID	It is mapped to the HTML attribute id. This attribute specifies a name to the object, which must be unique in the report. Data type: String
External Style	It is mapped to the HTML attribute style. This attribute specifies style information for the object. Data type: String
HrefLang	It is mapped to the HTML attribute hreflang. This attribute specifies the base language of the resource designated by a link and may only be used when a link is specified. Data type: String
Language	It is mapped to the HTML attribute lang. This attribute specifies the base language of the object's attribute values and text content. Data type: String
LongDesc	It is mapped to the HTML attribute longdesc. This attribute specifies a link to a long description of the object. Data type: String

Event property values

The provided values are based on the image client API in Version 7. The following table lists their mapping values in current version. For information about the descriptions and parameters of the mapping values, see Appendix 3: Parameters of user defined web actions.

Value	Mapping Name
exit(popwin)	user_exit()
firstPage()	user_firstPage()
lastPage()	user_lastPage()
nextPage()	user_nextPage()
oneSimpleFilter(columns, operators, values, logics)	user_oneSimpleFilter(columnMappingNames, operators, values, logics, instanceName)
oneSteSearch(value, isContent, coIName, isUp, matchcase, isWordOnly)	user_oneSteSearch(value, isContent, coIName, isUp, matchcase, isWordOnly)
oneSteSort(columns, sorts)	user_oneSteSort(columnMappingNames, orders, instanceName)
prevPage()	user_prevPage()
redo()	user_redo()
refresh()	user_refresh()
reset()	user_reset(instanceName)
savRpt()	user_savRpt()
searchNext(isFindTxt)	This API is not supported now.
saveRst()	
showExportExcelDlg()	
showExportHTMLDlg()	
showExportPDFDlg()	
showExportPSDlg()	user_showSaveResultDialog()
showExportRtfDlg()	
showExportTextDlg()	
showExportXmlDlg()	
showFilterPanel()	user_showFilterDialog()
showHelp()	user_showHelp(null)
showNavbar()	This API is not supported now.
showPageSetupDlg()	user_showPageSetupDialog()
showPrintDialog()	user_showPrintDialog()
showSearchDlg()	user_showSearchDialog()
showSortPanel()	user_showSortDialog()
showTOC()	user_showTOC()
showUserPanel()	user_showUserPanel()
undo()	user_undo()
zoom(value)	user_zoomTo(value)

Label

The properties of a label in a page report are:

Property Name	Description
General	
Auto Fit	Indicates the class type of the object. This property is read only.
Instance Name	Shows the instance name of the object. This property is read only.
Text Format	
Auto Fit	Specifies whether to adjust the width and height of the object according to the contents.
Data Type: Boolean	
Bold	Specifies whether to make the text bold.
Data Type: Boolean	
Font Face	Specifies the font of the text. Choose an option from the drop-down list.
Data Type: Enumeration	
Font Size	Specifies the font size of the text. Enter an integer value to change the size.
Data Type: Integer	
Horizontal Alignment	Specifies the horizontal alignment of the text in the object. Choose an option from the drop-down list.
Data Type: Enumeration	
Ignore HTML Tag	Specifies whether or not to ignore the HTML tags when exporting the report results to HTML format. • **true**: The text will appear in the HTML file the same as that in Report Designer (HTML tag elements are treated as the text, if any, will not be parsed). • **false**: HTML tag elements in the text will be parsed when exporting the report results to HTML format.
Data Type: Boolean	
Italic	This property is only for exporting the report results to HTML format. You may want to write the label contents in HTML tags in your report, and for the exported HTML files, the tags will be transferred as they are translated into HTML.
Data Type: Boolean	
Italic	Specifies whether to make the text italic.
Data Type: Boolean	
Maximum Width	Specifies the maximum width of the text displayed. Enter a numeric value to change the width.
Data Type: Integer	
This property often works with the Auto Fit property. If Auto Fit = true and Maximum Width is not equal to 0, the text will scroll until the width is this value.	
Strikethrough	Specifies whether to add a strikethrough line to the text.
Data Type: Boolean	
Text	Specifies the text in the label. Enter a string to display as the label text.
Data Type: String	
Underline	Specifies whether to underline the text.
Data Type: Boolean	
Vertical Alignment	Specifies the vertical justification of the text in the object. Choose an option from the drop-down list.
Data Type: Enumeration	
Word Wrap	Specifies whether to wrap the text to the width.
Data Type: Boolean	
Geometry	
Height	Specifies the height of the object, in inches or centimeters. Enter a numeric value to change the height.
Data Type: Float	
Width	Specifies the width of the object, in inches or centimeters. Enter a numeric value to change the width.
Data Type: Float	
X	Specifies the horizontal coordinate of the top left corner of the object, relative to its parent. This property is ignored if the Position property is set to static. Enter a numeric value to change the position in inches or centimeters.
Data Type: Float	
Y	Specifies the vertical coordinate of the top left corner of the object, relative to its parent. This property is ignored if the Position property is set to static. Enter a numeric value to change the position in inches or centimeters.
Data Type: Float	
Color	
Background	Specifies the background color of the object. Choose a color from the drop-down list, enter a hexadecimal RGB value (for example, 0xC0C0C0), chooseincreate a formula or add an expression that returns a color, or select Custom from the drop-down list to specify the color.
Data Type: String	
Foreground	Specifies the foreground color of the object. Choose a color from the drop-down list, enter a hexadecimal RGB value (for example, 0xC0C0C0), chooseincreate a formula or add an expression that returns a color, or select Custom from the drop-down list to specify the color.
Data Type: String	
CSS	
Class	Specifies a CSS class to be applied to the object which is a valid class in the CSS file.
Data Type: String	
ID	Specifies the ID of the object, which must be unique in the report. The ID property can be a style sheet selector.
Data Type: String	
Style	The property can be used in two ways. • Specifies a style to be applied to the object. Choose a style from the drop-down list or enter a valid style name. If the style name is not valid, the object will use the default style property. Refer to the Style Sheet Reference Guide for valid styles in the style name. • Specifies a CSS selector to be applied to the object. Enter a valid class name from the CSS file.
Data Type: String	
Excel	
Category Index	Specifies the X coordinate of the object relative to its parent container when exported to Excel (OSC. Assumed with Top. Columns) property at the report tab level must be set to true for this property to take effect.
Data Type: Integer	
Row Index	Specifies the Y coordinate of the object relative to its parent container when exported to Excel (OSC. Assumed with Top. Columns) property at the report tab level must be set to true for this property to take effect.
Data Type: Integer	
Padding	
Bottom Padding	Specifies the space in inches or centimeters between the text and the bottom border of the object. Enter a numeric value to change the bottom padding.
Data Type: Float	
Left Padding	Specifies the space in inches or centimeters between the text and the left border of the object. Enter a numeric value to change the left padding.
Data Type: Float	
Right Padding	Specifies the space in inches or centimeters between the text and the right border of the object. Enter a numeric value to change the right padding.
Data Type: Float	
Top Padding	Specifies the space in inches or centimeters between the text and the top border of the object. Enter a numeric value to change the top padding.
Data Type: Float	
Border	
Border Color	Specifies the color of the border of the object. Choose a color from the drop-down list, enter a hexadecimal RGB value (for example, 0xC0C0C0), chooseincreate a formula or add an expression that returns a color, or select Custom from the drop-down list to specify the color.
Data Type: String	
Border Thickness	Specifies the width of the border in inches or centimeters. Enter a numeric value to change the thickness.
Data Type: Float	
Bottom Line	Specifies the line style of the bottom border of the object. Choose a style from the drop-down list.
Data Type: Enumeration	
Left Line	Specifies the line style of the left border of the object. Choose a style from the drop-down list.
Data Type: Enumeration	
Right Line	Specifies the line style of the right border of the object. Choose a style from the drop-down list.
Data Type: Enumeration	
Shadow	Specifies whether to draw two shadowed borders, beneath and to the right of the object.
Data Type: Boolean	
Shadow Color	Specifies the color of the shadow. Choose a color from the drop-down list, enter a hexadecimal RGB value (for example, 0xC0C0C0), chooseincreate a formula or add an expression that returns a color, or select Custom from the drop-down list to specify the color.
Data Type: String	
Top Line	Specifies the style of the top border of the object. Choose a style from the drop-down list.
Data Type: Enumeration	
Pattern	
Pattern Color	Specifies the color in which to draw a pattern in fill the object. Choose a color from the drop-down list, enter a hexadecimal RGB value (for example, 0xC0C0C0), chooseincreate a formula or add an expression that returns a color, or select Custom from the drop-down list to specify the color and RGB value.
Data Type: String	
Pattern Style	Specifies the style of the pattern. Choose an option from the drop-down list. • **none** - No pattern will be applied to the object. • **50%** - The object will be filled in the Pattern Color of 50% transparency. • **horizontal** - The pattern will be made of horizontal lines of the Pattern Color. • **vertical** - The pattern will be made of vertical lines of the Pattern Color. • **grid** - The pattern will be an overlapping of horizontal and vertical lines of the Pattern Color. • **diagonal** - The pattern will be made of diagonal lines of the Pattern Color.
Data Type: Enumeration	
Others	
Bind Column	When the label's Sortable or Filterable property is set to true, you need to specify a field as the value to bind the records of the field can be sorted or filtered on the project fetch binding table. Enter the field name in Page Report Studio.
Data Type: String	
Detail Target Frame	Specifies the target frame where the object is linked to. Click in the value cell to set the detail target frame details. refer to Linking to a detail report.
Data Type: String	
This property is only for a target window or frame in which to display the detailed information. This property is enabling when Go To Page property is set to Detail Report. Enter a valid target frame name. • **Current Setting** - Indicates the detailed information according to setting of the Pop Up New Window for Links in the Customize Portal panel of the Profile dialog in JReport Server. • **New Window** - Loads the detailed information into a new window. The window is not named. • **Whole Window** - Loads the detailed information into the full browser window. • **Same Frame** - Loads the detailed information into the same frame as the object. • **Parent Frame** - Loads the detailed information into the parent frame of the frame in which the object is displayed. • **Other Frame** - Loads the detailed information into some other specified frame. If selected, you can also directly input the name of the frame you have defined in the text box. If the frame is not defined in advance, the detailed information will be loaded into a new window.	
Data Type: String	
Enable Hyperlink in Excel	Specifies whether to enable the link defined on the object when exporting the report to an Excel file.
Data Type: Boolean	
Enable Hyperlink in HTML	Specifies whether to enable the link defined on the object when exporting the report to HTML file.
Data Type: Boolean	
Enable Hyperlink in PDF	Specifies whether to enable the link defined on the object when exporting the report to a PDF file.
Data Type: Boolean	
Export to Applet	Specifies whether to include the object when exporting the report to Applet.
Data Type: Boolean	
Export to CSV	Specifies whether to include the object when exporting the report to CSV. If it is set to true, only the string value will be exported when the object is displayed as a hands-on text field, and only the text will be exported when the object is displayed as a checkbox, radio button or button.
Data Type: Boolean	
Export to Excel	Specifies whether to include the object when exporting the report to Excel. It is set to true, only the string value will be exported when the object is displayed as a hands-on text field, and only the text will be exported when the object is displayed as a checkbox, radio button or button.
Data Type: Boolean	
Export to HTML	Specifies whether to include the object when exporting the report to HTML.
Data Type: Boolean	
Export to PDF	Specifies whether to include the object when exporting the report to PDF. If it is set to true, only the string value will be exported when the object is displayed as a text field.
Data Type: Boolean	
Export to PostScript	Specifies whether to include the object when exporting the report to PostScript. If it is set to true, only the string value will be exported when the object is displayed as a text field.
Data Type: Boolean	
Export to Report Ready	Specifies whether to include the object when exporting the report to Report Ready.
Data Type: Boolean	
Export to RTF	Specifies whether to include the object when exporting the report to RTF. If it is set to true, only the string value will be exported when the object is displayed as a text field, and only the text will be exported when the object is displayed as a checkbox, radio button or button.
Data Type: Boolean	
Export to Text	Specifies whether to include the object when exporting the report to Text. If it is set to true, only the string value will be exported when the object is displayed as a hands-on text field, and only the text will be exported when the object is displayed as a checkbox, radio button or button.
Data Type: Boolean	
Export to XML	Specifies whether to include the object when exporting the report to XML. If it is set to true, only the string value will be exported when the object is displayed as a hands-on text field, and only the text will be exported when the object is displayed as a checkbox, radio button or button.
Data Type: Boolean	
Filter Options	Specifies the filter-related options that will be displayed on the shortcut menu when right-clicking the object in a Report Viewer. If Default is selected, the filter options displayed will be determined by the setting in the Filter Options tab of the Report Properties in Customize Portal in Page Report Studio > Properties > Select > Select Menu. For more information, refer to Setting filter options for a field.
Data Type: Enumeration	
Filterable	Specifies whether the records can be filtered. If true, a filter button appears beside the bind column in the report viewer. Clicking the button displays a dialog box that lists the records based on a specific field's value. The field is specified via the Bind Column property.
Data Type: Boolean	
Go to Detail	Specifies whether to show the detailed information when you click the object in Page Report Viewer. This property is enabling when Go To Page property is set to Detail Report. Select an option from the drop-down list.
Data Type: Enumeration	
Invisible	Specifies whether to show the object in the design area and in the report results. All formulas and calculations will be prepared if the property is set to false.
Data Type: Boolean	
Link	Specifies to link the object to another report, a website, an e-mail address or a Bookmark type field. Click in the value cell to set the link target. For details, see Binding a link to a label.
Data Type: String	
Logic Column	Specifies whether to show the object in the text inside table cell in the same row when the Page Report Viewer's Logic Column Option is hidden. Available to labels in table only. Choose an option from the drop-down list.
Data Type: Enumeration	
Notes:	
• The next visible table cell should be completely empty, that is, the cell should not hold any content including blank space. • When the property for several objects in the same row is set to next visible column, the next visible cell of the last object in the same row will be the next visible cell. The optional column will be shown in the next visible cell.	
Position	Specifies the position of the object. Choose an option from the drop-down list. • **static** - The object is positioned in the location in which it is created. If selected, the X, Y and other position-related properties will be hidden or disabled. The dimensions of the object are still controlled by the Height and Width properties. • **relative** - The object is positioned at an offset to the position in which it is created. The offset is determined by the X and Y property values. • **absolute** - The object will be removed from the previous sequence of dragging and dropping or by setting its X and Y property values. The position of the object is not related to the previous objects in the same container.
Data Type: Enumeration	
Record Location	Specifies the calculation point for the properties which use formulas. Choose an option from the drop-down list. • **Default** - The values of the properties are calculated in the default location where they are located. • **Page Header** - The values of the properties are calculated in the page header. • **Page Footer** - The values of the properties are calculated in the page footer.
Data Type: Enumeration	
Sortable	Specifies whether the records can be sorted. If true, a sort button appears beside the label when viewing the report in Page Report Studio. You can set the button to arrange the records in ascending or descending order. The field is specified via the Bind Column property.
Data Type: Boolean	
Suppress	Specifies whether to show the object in the design area and in the report results. All formulas and calculations will be prepared if the property is set to true. **Note:** When both the Invisible and Suppress properties of an object are set to true, the object will be hidden.
Data Type: Boolean	
Suppress When No Records	Specifies whether to suppress the object in the report results when no record is returned to its parent data container.
Data Type: Boolean	
Transfer Style	Specifies whether the style group of the primary report will be applied to the linked report if the object is linked to a report.
Data Type: Boolean	
TOC	
Anchor Display Value	Specifies a string or formula to display the TOC entry for this object in the TOC (Table of Contents) in the report viewer. Enter a string or click in the value cell. The TOC Anchor property must be set to true for this property to take effect.
Data Type: String	
TOC Anchor	Specifies whether to include the object in the TOC Browser for the report.
Data Type: Boolean	
Accessibility	
External Accessibility	It is mapped to the HTML attribute accessibility. This attribute specifies an access key to the object.
Data Type: String	
External CSS Class Name	Specifies a class selector to be applied to the object when exported as HTML. Enter a valid class name from the CSS file.
Data Type: String	
External Dir	It is mapped to the HTML attribute dir. This attribute specifies the base direction of the object's attribute values and text content. Possible values are: • LTR - Left-to-right text or table. • RTL - Right-to-left text or table.
Data Type: String	
External ID	It is mapped to the HTML attribute id. This attribute specifies a name to the object, which must be unique in the report.
Data Type: String	
External Style	It is mapped to the HTML attribute style. This attribute specifies style information for the object.
Data Type: String	
External Tabindex	It is mapped to the HTML attribute tabindex. This attribute specifies the position of the object in the tabbing order for the current report. Enter an integer equal to or greater than 0, and less than 65535.
Data Type: Integer	
External Title	It is mapped to the HTML attribute title. This attribute offers advisory information about the object.
Data Type: String	
HreflLang	It is mapped to the HTML attribute hreflang. This attribute specifies the base language of the resource designated by a link and may only be used when a link is specified.
Data Type: Language	
Language	It is mapped to the HTML attribute lang. This attribute specifies the base language of the object's attribute values and text content.
Data Type: Language	
LongDesc	It is mapped to the HTML attribute longdesc. This attribute specifies a link to a long description of the object.
Data Type: String	

Auto Fit , Maximum Width, Word Wrap

Here is a label:
Label product sales by region

We want to show it as:
Label product sales by region

Auto Fit = true

- Maximum Width: 1
- Word Wrap: true

Auto Fit=true enables the contents to grow horizontally but stop when the width becomes 1. Word Wrap=true enables the remaining contents to wrap downward if no space is left horizontally.

Style, ID, Class

The following is an example of using the Style, ID, Class property values as style sheet selectors in a CSS style file:

NewCSSFile.css:

```
@charset "UTF-8";
textfield {Background: #FT0000}
//textfield{}
textfield[Style="textfield"]{Background: #0000FF}
textfield#Text{}
textfield[Text="Text"]{Background: #FFFF00}
//class{}/
textfield.CIBackground: #00FFFF}
```

Record Location

This property is to specify the calculation point for the properties of the object. Here is an example:

A report specifies to display Continued on Next Page for every page that has another following it. If this label is inserted in the Page Footer section, then every page containing the last page will contain the message. There is a way to write formulas that can resolve the problem:

Continue1:
global boolean jrnc;
jrFalse;

Continue2:
global boolean jrnc;
jrTrue;

Continue3:
pageNumber();
return 1;

Insert Continue1 into the ReportHeader. Continue2 into the ReportFooter. To track the calculation, you can insert Continue3 into the PageFooter. If you do not want to see these formulas in view mode, you can suppress them.

Insert label Continued on Next Page in the PageHeader, and use Continue3 to control its property Invisible.

Set the property Record Location of the above label to be Page Footer, so that Continue3 returns value which is calculated in the last instead of in the PageHeader.

PageNumber is included in the formula to force it to execute as a page level formula in the order that formula is encountered rather than at the end of a group or report. For more information, see Formula levels.

Sorting

When you view your report at server runtime, you can sort records in the data buffer on certain columns. Sorting is supported in the Table and Crosstab objects. To sort records, you need to enable the Sortable property of the label object.

1. In the Report Inspector, set the Sortable property of this label to true.
2. Bind this label to a field on which you want to sort by setting this field as the value of the Bind Column property.
3. Save and publish the report to JReport Server.
4. Run the report, and you will see a sort button beside the label, which can be clicked for sorting.

Filterable

When you view your report at server runtime, you can set a series of conditions to filter the data buffer and reproduce the report. You are allowed to use a label to control the filter condition for the data, which can be enabled by setting the Filterable property of the label object.

1. In the Report Inspector, set the Filterable property of this label to true.
2. Bind this label to a field on which you want to filter by setting this field as the value of the Bind Column property.
3. Save the report and publish it to JReport Server.
4. Run the report. You can click the filter button beside the label to filter the data, in the same way as the shortcut menu.

Multimedia object (OLE object)

The properties of a multimedia object which is also called OLE object in a page report are:

Property Name	Description
General	
Class Type	Indicates the class type of the object. This property is read only.
Instance Name	Shows the instance name of the object. This property is read only.
Geometry	
Height	Specifies the height of the object, in inches or centimeters. Enter a numeric value to change the height. Data type: Float
Width	Specifies the width of the object, in inches or centimeters. Enter a numeric value to change the width. Data type: Float
X	Specifies the horizontal coordinate of the top left corner of the object, relative to its parent container. This property is ignored if the Position property is set to static. Enter a numeric value to change the position in inches or centimeters. Data type: Float
Y	Specifies the vertical coordinate of the top left corner of the object, relative to its parent container. This property is ignored if the Position property is set to static. Enter a numeric value to change the position in inches or centimeters. Data type: Float
Color	
Background	Specifies the background color of the object. Choose a color from the drop-down list, enter a hexadecimal RGB value (for example, 0xff0000), choose/create a formula or edit an expression that returns a color, or select Custom from the drop-down list to specify the color. Data type: String
CSS	
Class	Specifies a CSS class to be applied to the object which is a valid class in the CSS file. Data type: String
Style	The property can be used in two ways. • Specifies a style to be applied to the object. Choose a style from the drop-down list which is available when a Style Group has been selected at the report tab level and when there are styles in the Style Group that can be applied to the object, or type in the style name. • Specifies a CSS selector to be applied to the object. Enter a valid class name from the CSS file. Data type: String
Accessibility	
External CSS Class Selector	Specifies a class selector to be applied to the object when exported as HTML. Enter a valid class name from the CSS file. Data type: String
Others	
Export to Applet	Specifies whether to include the object when exporting the report to Applet. Data type: Boolean
Export to CSV	Specifies whether to include the object when exporting the report to CSV. Data type: Boolean
Export to Excel	Specifies whether to include the object when exporting the report to Excel. Data type: Boolean
Export to HTML	Specifies whether to include the object when exporting the report to HTML. Data type: Boolean
Export to PDF	Specifies whether to include the object when exporting the report to PDF. Data type: Boolean
Export to PostScript	Specifies whether to include the object when exporting the report to PostScript. Data type: Boolean
Export to Report Result	Specifies whether to include the object when exporting the report to Report Result. Data type: Boolean
Export to RTF	Specifies whether to include the object when exporting the report to RTF. Data type: Boolean
Export to Text	Specifies whether to include the object when exporting the report to Text. Data type: Boolean
Export to XML	Specifies whether to include the object when exporting the report to XML. Data type: Boolean
Invisible	Specifies whether to show the object in the design area and in the report results. All formulas and calculations will still be performed if the property is set to true. Data type: Boolean
Position	Specifies the position of the object. Choose an option from the drop-down list. • **static** - The object is positioned at the location in which it is inserted. If selected, the X, Y and other position-related properties will be hidden or disabled. The position of the object is affected by preceding objects in the same container. • **relative** - The object is positioned at an offset to the position at which it is inserted. The offset is determined by the X and Y property values. • **absolute** - The object will be located at the position specified by dragging and dropping or by setting its X and Y property values. The position of the object is not affected by preceding objects in the same container. Data type: Enumeration
Record Location	Specifies the calculation point for the properties which use formulas. Choose an option from the drop-down list. • **Default** - The values of the properties are calculated in the default location where they are located. • **Page Header** - The values of the properties are calculated in the page header. • **Page Footer** - The values of the properties are calculated in the page footer. Data type: Enumeration
Suppress	Specifies whether to show the object in the design area and in the report results. All formulas and calculations will be skipped if the property is set to true. Data type: Boolean **Note:** When both the Invisible and Suppress properties of an object are set to true, Suppress has the higher priority.
Suppress When No Records	Specifies whether to display the object in the report results when no record is returned to its parent data container. Data type: Boolean
Excel	
Column Index	Specifies the X coordinate of the object relative to its parent container when exported to Excel or CSV, measured in cells. The Columned property at the report tab level must be set to true for this property to take effect. Data type: Integer
Row Index	Specifies the Y coordinate of the object relative to its parent container when exported to Excel or CSV, measured in cells. The Columned property at the report tab level must be set to true for this property to take effect. Data type: Integer
Border	
Border Color	Specifies the color of the border of the object. Choose a color from the drop-down list, enter a hexadecimal RGB value (for example, 0xCCAA00), choose/create a formula or edit an expression that returns a color, or select Custom from the drop-down list to specify the color. Data type: String
Border Thickness	Specifies the width of the border in inches or centimeters. Enter a numeric value to change the thickness. Data type: Float
Bottom Line	Specifies the line style of the bottom border of the object. Choose a style from the drop-down list. Data type: Enumeration
Left Line	Specifies the line style of the left border of the object. Choose a style from the drop-down list. Data type: Enumeration
Right Line	Specifies the line style of the right border of the object. Choose a style from the drop-down list. Data type: Enumeration
Top Line	Specifies the line style of the top border of the object. Choose a style from the drop-down list. Data type: Enumeration
Object Settings	
Access Key	Specifies the accelerator key for the object. Data type: String
Alternate Text	Specifies the text will be displayed instead if the object cannot be displayed. Data type: String
Archive	Specifies the character string that can be used to implement your own archive functionality for the object. Data type: String
Class ID	Specifies the class identifier of the object. Data type: String
Code Base	Specifies the base path used to resolve relative URIs specified by the Class ID and Archive properties. When null, its default value is the base URI of the current report. Data type: String
Declare	Specifies whether to make the current object definition a declaration only. The object must be instantiated by a subsequent object definition referring to this declaration. Data type: Boolean
Horizontal Space	Specifies the horizontal margin for the object. Data type: Float
Name	Specifies the name of the object. Data type: String
Stand By	Specifies the message to be displayed while the object is being loaded. Data type: String
Tab Index	Specifies the index that defines the tab order for the object. Data type: Integer
Title	Specifies the tip information of the object, which will be displayed when you hover the mouse pointer over the object after the report is exported to HTML or Page Report Result. Data type: String
Use Map	Specifies the URL, often with a bookmark extension (#name), to use as a client-side image map. Data type: String
Value Type	Specifies the MIME (Multipurpose Internet Mail Extensions) type of the object. Data type: String
Vertical Space	Specifies the vertical margin for the object. Data type: Float

Page panel

The properties of a page panel in a page report are:

Property Name	Description
General	
Class Type	Indicates the class type of the object. This property is read only.
Instance Name	Shows the instance name of the object. This property is read only.
Paper	
Height	Specifies the height of the page in inches or centimeters. The Width and Height values change when Orientation changes. The Page Type setting determines the Width and Height values but you can change the latter as desired. Enter a numeric value to change the height. Data type: Float
Orientation	Specifies how to position the report page. The Orientation setting affects the Width and Height values. Choose an option from the drop-down list. • **portrait** - The page is positioned vertically. • **landscape** - The page is positioned horizontally. Data type: Enumeration
Page Type	Specifies the report page dimensions. Choose a size from the drop-down list. Data type: Enumeration
Width	Specifies the width of the page in inches or centimeters. The Width and Height values change when Orientation changes. The Page Type setting determines the Width and Height values but you can change the latter as desired. Enter a numeric value to change the width. Data type: Float
Margin	
Bottom Margin	Specifies the distance in inches or centimeters between the bottom page edge and the data area. Enter a numeric value to change the margin. Data type: Float
Left Margin	Specifies the distance in inches or centimeters between the left page edge and the data area. Enter a numeric value to change the margin. Data type: Float
Right Margin	Specifies the distance in inches or centimeters between the right page edge and the data area. Enter a numeric value to change the margin. Data type: Float
Top Margin	Specifies the distance in inches or centimeters between the top page edge and the data area. Enter a numeric value to change the thickness. Data type: Float

The following topics list properties of the children of a page panel:

- Page header panel
- Page footer panel

Page header panel

The properties of a page header panel in a page report are:

Property Name	Description
General	
Class Type	Indicates the class type of the object. This property is read only.
Data Inherit	Indicates whether the dataset for this object is inherited from another object. This property is read only.
Dataset	Indicates the dataset used by the object. This property is read only.
Instance Name	Shows the instance name of the object. This property is read only.
Geometry	
Height	Specifies the height of the object, in inches or centimeters. Enter a numeric value to change the height. Data type: Float
Width	Displays the width of the object, in inches or centimeters. This property is read only.
Color	
Background	Specifies the background color and fill effect of the object. Choose a color from the drop-down list, enter a hexadecimal RGB value (for example, 0xff0000), choose/create a formula or edit an expression that returns a color, or select Custom from the drop-down list to specify the color and fill effect. Data type: String
CSS	
Class	Specifies a CSS class to be applied to the object which is a valid class in the CSS file. Data type: String
Style	The property can be used in two ways. • Specifies a style to be applied to the object. Choose a style from the drop-down list which is available when a Style Group has been selected at the report tab level and when there are styles in the Style Group that can be applied to the object, or type in the style name. • Specifies a CSS selector to be applied to the object. Enter a valid class name from the CSS file. Data type: String
Others	
Export to CSV	Specifies whether to include the object when exporting the report to CSV. Data type: Boolean
Export to Page Report Result	Specifies whether to include the object when exporting the report to Page Report Result. Data type: Boolean
Export to Excel	Specifies whether to include the object when exporting the report to Excel. Data type: Boolean
Invisible	Specifies whether to show the object in the design area and in the report results. All formulas and calculations will still be performed if the property is set to true. Data type: Boolean
Record Location	Specifies the calculation point for the properties which use formulas. Choose an option from the drop-down list. • **Default** - The values of the properties are calculated in the default location where they are located. • **Page Header** - The values of the properties are calculated in the page header. • **Page Footer** - The values of the properties are calculated in the page footer. Data type: Enumeration
Show Bottom Line	Specifies whether to include a horizontal line at the bottom of the panel. Applies to panels in table components only. The default is false, which means that no horizontal line appears at the bottom of the panel. Data type: Boolean
Suppress	Specifies whether to show the object in the design area and in the report results. All formulas and calculations will be skipped if the property is set to true. Data type: Boolean **Note:** When both the Invisible and Suppress properties of an object are set to true, Suppress has the higher priority.
Suppress Blank Panel	Specifies whether a panel with no data is included in the report results. The default is false which means an empty panel is shown. Data type: Boolean
Suppress When No Records	Specifies whether to display the object in the report results when no record is returned to its parent data container. Data type: Boolean
Underlay	Specifies whether the panel will be layered beneath the next panel in the banded object. A panel layered beneath a subsequent panel can be used to contain a background picture or watermark. Banded page header, page footer, or group footer panels cannot be layered. The default is false which means that the panel is not layered beneath the next panel. Data type: Boolean
Border	
Border Color	Specifies the color of the border of the object. Choose a color from the drop-down list, enter a hexadecimal RGB value (for example, 0xCCAA00), choose/create a formula or edit an expression that returns a color, or select Custom from the drop-down list to specify the color. Data type: String
Border Thickness	Specifies the width of the border in inches or centimeters. Enter a numeric value to change the thickness. Data type: Float
Bottom Line	Specifies the line style of the bottom border of the object. Choose a style from the drop-down list. Data type: Enumeration
Left Line	Specifies the line style of the left border of the object. Choose a style from the drop-down list. Data type: Enumeration
Right Line	Specifies the line style of the right border of the object. Choose a style from the drop-down list. Data type: Enumeration
Top Line	Specifies the line style of the top border of the object. Choose a style from the drop-down list. Data type: Enumeration
Accessibility	
External CSS Class Selector	Specifies a class selector to be applied to the object when exported as HTML. Enter a valid class name from the CSS file. Data type: String
External ID	It is mapped to the HTML attribute *id*. This attribute specifies a name to the object, which must be unique in the report. Data type: String
External Style	It is mapped to the HTML attribute *style*. This attribute specifies style information for the object. Data type: String
Language	It is mapped to the HTML attribute *lang*. This attribute specifies the base language of the object's attribute values and text content. Data type: String

Page footer panel

The properties of a page footer panel in a page report are:

Property Name	Description
General	
Class Type	Indicates the class type of the object. This property is read only.
Data Inherit	Indicates whether the dataset for this object is inherited from another object. This property is read only.
Dataset	Indicates the dataset used by the object. This property is read only.
Instance Name	Shows the instance name of the object. This property is read only.
Geometry	
Height	Specifies the height of the object, in inches or centimeters. Enter a numeric value to change the height. Data type: Float
Width	Displays the width of the object, in inches or centimeters. This property is read only.
Color	
Background	Specifies the background color and fill effect of the object. Choose a color from the drop-down list, enter a hexadecimal RGB value (for example, 0xff0000), choose/create a formula or edit an expression that returns a color, or select Custom from the drop-down list to specify the color and fill effect. Data type: String
CSS	
Class	Specifies a CSS class to be applied to the object which is a valid class in the CSS file. Data type: String
Style	The property can be used in two ways. • Specifies a style to be applied to the object. Choose a style from the drop-down list which is available when a Style Group has been selected at the report tab level and when there are styles in the Style Group that can be applied to the object, or type in the style name. • Specifies a CSS selector to be applied to the object. Enter a valid class name from the CSS file. Data type: String
Others	
Export to CSV	Specifies whether to include the object when exporting the report to CSV. Data type: Boolean
Export to Page Report Result	Specifies whether to include the object when exporting the report to Page Report Result. Data type: Boolean
Export to Excel	Specifies whether to include the object when exporting the report to Excel. Data type: Boolean
Invisible	Specifies whether to show the object in the design area and in the report results. All formulas and calculations will still be performed if the property is set to true. Data type: Boolean
Record Location	Specifies the calculation point for the properties which use formulas. Choose an option from the drop-down list. • **Default** - The values of the properties are calculated in the default location where they are located. • **Page Header** - The values of the properties are calculated in the page header. • **Page Footer** - The values of the properties are calculated in the page footer. Data type: Enumeration
Show Bottom Line	Specifies whether to include a horizontal line at the bottom of the panel. Applies to panels in table components only. The default is false, which means that no horizontal line appears at the bottom of the panel. Data type: Boolean
Suppress	Specifies whether to show the object in the design area and in the report results. All formulas and calculations will be skipped if the property is set to true. Data type: Boolean **Note:** When both the Invisible and Suppress properties of an object are set to true, Suppress has the higher priority.
Suppress Blank Panel	Specifies whether a panel with no data is included in the report results. The default is false which means an empty panel is shown. Data type: Boolean
Suppress on Last Page	If true, the footer panel on the last page will not be displayed. Data type: Boolean
Suppress When No Records	Specifies whether to display the object in the report results when no record is returned to its parent data container. Data type: Boolean
Underlay	Specifies whether the panel will be layered beneath the next panel in the banded object. A panel layered beneath a subsequent panel can be used to contain a background picture or watermark. Banded page header, page footer, or group footer panels cannot be layered. The default is false which means that the panel is not layered beneath the next panel. Data type: Boolean
Border	
Border Color	Specifies the color of the border of the object. Choose a color from the drop-down list, enter a hexadecimal RGB value (for example, 0xCCAA00), choose/create a formula or edit an expression that returns a color, or select Custom from the drop-down list to specify the color. Data type: String
Border Thickness	Specifies the width of the border in inches or centimeters. Enter a numeric value to change the thickness. Data type: Float
Bottom Line	Specifies the line style of the bottom border of the object. Choose a style from the drop-down list. Data type: Enumeration
Left Line	Specifies the line style of the left border of the object. Choose a style from the drop-down list. Data type: Enumeration
Right Line	Specifies the line style of the right border of the object. Choose a style from the drop-down list. Data type: Enumeration
Top Line	Specifies the line style of the top border of the object. Choose a style from the drop-down list. Data type: Enumeration
Accessibility	
External CSS Class Selector	Specifies a class selector to be applied to the object when exported as HTML. Enter a valid class name from the CSS file. Data type: String
External ID	It is mapped to the HTML attribute *id*. This attribute specifies a name to the object, which must be unique in the report. Data type: String
External Style	It is mapped to the HTML attribute *style*. This attribute specifies style information for the object. Data type: String
Language	It is mapped to the HTML attribute *lang*. This attribute specifies the base language of the object's attribute values and text content. Data type: String

Page report

A page report may contain one or more report tabs, and each report tab contain a report body. The Report Inspector enables you to focus on one of the three levels, namely the page report, report tab, and report body. By default, the root node in the Report Inspector represents the current report tab.

You can click the Up button on the toolbar to show the page report as the root node (if the current root node represents the report body, you should click that button twice). After showing the page report node in the Report Inspector, you can set its properties:

Property Name	Description
General	
Class Type	Indicates the class type of the object. This property is read only.
Instance Name	Shows the instance name of the object. This property is read only.
Others	
Import Parameter Values	Specifies the name of a class file from which to import default values for parameters used in the report, for example, TestParamList. Then when viewing the report, the displayed Enter Parameter Values dialog will only list the imported default parameters. To make the property work, you need specify the Parameter List Auto property to false. For more information, see Importing parameter values. Data type: String
Parameter List Auto	Specifies whether to get the default values for parameters used in the report from values defined in the catalog. • **true** - Gets parameter default values from values defined in catalog. • **false** - Gets parameter default values from the class file specified by the Import Parameter Values property. Data type: Boolean
Push Down Group Query	Specifies whether to push down group level summary computations in the page report to the DBMS at runtime. Choose an option from the drop-down list. • default - The settings of the Push Down Group Query property for the queries on which the data components in the page report are created will be applied for each data component correspondingly. • true - JReport will push down the group level summary computations to the DBMS if the DBMS can do the computations, otherwise, JReport will do the computations itself. • false - JReport will do the group level summary computations itself. Data type: Boolean

Parameter field

The properties of a parameter field in a page report are:

Property Name	Description
General	
Data Type	Shows the data type of the parameter. This property is read only.
Default Value	Shows the default value of the parameter. This property is read only.
Field Type	Shows what kind of field it is. This property is read only.
Instance Name	Shows the instance name of the object. This property is read only.
Parameter Name	Shows the name of the parameter. This property is read only.
Geometry	
Height	Specifies the height of the object, in inches or centimeters. Enter a numeric value to change the height. Data type: Float
Width	Specifies the width of the object, in inches or centimeters. Enter a numeric value to change the width. Data type: Float
X	Specifies the horizontal coordinate of the top left corner of the object, relative to its parent container. This property is ignored if the Position property is set to static. Enter a numeric value to change the position in inches or centimeters. Data type: Float
Y	Specifies the vertical coordinate of the top left corner of the object, relative to its parent container. This property is ignored if the Position property is set to static. Enter a numeric value to change the position in inches or centimeters. Data type: Float
Color	
Background	Specifies the background color of the object. Choose a color from the drop-down list, enter a hexadecimal RGB value (for example, 0xff0000), choose/create a formula or edit an expression that returns a color, or select Custom from the drop-down list to specify the color. Data type: String
Foreground	Specifies the foreground color of the object. Choose a color from the drop-down list, enter a hexadecimal RGB value (for example, 0xff0000), choose/create a formula or edit an expression that returns a color, or select Custom from the drop-down list to specify the color. Data type: String
CSS	
ID	Specifies the identifier of the object, which must be unique in the report. The ID property can be a style sheet selector. A String type value.
Class	Specifies a CSS class to be applied to the object which is a valid class in the CSS file. Data type: String
Style	The property can be used in two ways. • Specifies a style to be applied to the object. Choose a style from the drop-down list which is available when a Style Group has been selected at the report tab level and when there are styles in the Style Group that can be applied to the object, or type in the style name. • Specifies a CSS selector to be applied to the object. Enter a valid class name from the CSS file. Data type: String
Excel	
Column Index	Specifies the X coordinate of the object relative to its parent container when exported to Excel or CSV, measured in cells. The Columned property at the report tab level must be set to true for this property to take effect. Data type: Integer
Row Index	Specifies the Y coordinate of the object relative to its parent container when exported to Excel or CSV, measured in cells. The Columned property at the report tab level must be set to true for this property to take effect. Data type: Integer
Padding	
Bottom Padding	Specifies the space in inches or centimeters between the text and the bottom border of the object. Enter a numeric value to change the padding. Data type: Float
Left Padding	Specifies the space in inches or centimeters between the text and the left border of the object. Enter a numeric value to change the padding. Data type: Float
Right Padding	Specifies the space in inches or centimeters between the text and the right border of the object. Enter a numeric value to change the padding. Data type: Float
Top Padding	Specifies the space in inches or centimeters between the text and the top border of the object. Enter a numeric value to change the padding. Data type: Float
Border	
Border Color	Specifies the color of the border of the object. Choose a color from the drop-down list, enter a hexadecimal RGB value (for example, 0xCCAA00), choose/create a formula or edit an expression that returns a color, or select Custom from the drop-down list to specify the color. Data type: String
Border Thickness	Specifies the width of the border in inches or centimeters. Enter a numeric value to change the thickness. Data type: Float
Bottom Line	Specifies the line style of the bottom border of the object. Choose a style from the drop-down list. Data type: Enumeration
Left Line	Specifies the line style of the left border of the object. Choose a style from the drop-down list. Data type: Enumeration
Right Line	Specifies the line style of the right border of the object. Choose a style from the drop-down list. Data type: Enumeration
Shadow	Specifies whether to draw two shadowed borders, beneath and to the right of the object. Data type: Boolean
Shadow Color	Specifies the color of the shadow. Choose a color from the drop-down list, enter a hexadecimal RGB value (for example, 0xff0000), choose/create a formula or edit an expression that returns a color, or select Custom from the drop-down list to specify the color and fill effect. Data type: String
Top Line	Specifies the line style of the top border of the object. Choose a style from the drop-down list. Data type: Enumeration
Pattern	
Pattern Color	Specifies the color in which to draw a pattern to fill the object. Choose a color from the drop-down list, enter a hexadecimal RGB value (for example, 0xff0000), choose/create a formula or edit an expression that returns a color, or select Custom from the drop-down list to specify the color and fill effect. Data type: String
Pattern Style	Specifies the style of the pattern. Choose an option from the drop-down list. • **none** - No pattern will be applied to the object. • **50 %** - The object will be filled in the Pattern Color of 50%-transparency. • **horizontal** - The pattern will be made of horizontal lines of the Pattern Color. • **vertical** - The pattern will be made of vertical lines of the Pattern Color. • **grid** - The pattern will be an overlapping of horizontal and vertical lines of the Pattern Color. • **diagonal** - The pattern will be made of diagonal lines of the Pattern Color. Data type: Enumeration
Text Format	
Auto Fit	Specifies whether to adjust the width and height of the object according to the contents. Data type: Boolean
Bold	Specifies whether to make the text bold. Data type: Boolean
Font Face	Specifies the font of the text. Choose an option from the drop-down list. Data type: Enumeration
Font Size	Specifies the font size of the text. Enter an integer value to change the size. Data type: Integer
Format	Specifies the display format of the text in the report result. It varies with data type and can be manually defined. Data type: String **Note:** For the BigDecimal type, to avoid precision loss, you should enter a prefix JRD when setting the format property value.
Horizontal Alignment	Specifies the horizontal justification of the text in the object. Choose an option from the drop-down list. Data type: Enumeration
Ignore HTML Tag	Specifies whether or not to ignore the HTML tags when exporting the report results to HTML format. • **true** - The text will appear in the HTML file the same as that in JReport Designer (HTML tag elements in the text, if any, will not be parsed). • **false** - HTML tag elements in the text will be parsed when exporting the report results to HTML format. This property is useful for exporting the report results to HTML format. You may want to write the label contents in HTML tags in your report, and for the exported HTML files, the tags will be transferred as they are translated into HTML. Data type: Boolean
Italic	Specifies whether to make the text italic. Data type: Boolean
Maximum Width	Specifies the maximum width of the text displayed. Enter a numeric value to change the width. This property often works with the Auto Fit property. If Auto Fit = true and Maximum Width is not equal to 0, the text will extend until the width is this value. Data type: Float
Strikethrough	Specifies whether to add a strikethrough line to the text. Data type: Boolean
Underline	Specifies whether to underline the text. Data type: Boolean
Vertical Alignment	Specifies the vertical justification of the text in the object. Choose an option from the drop-down list. Data type: Enumeration
Word Wrap	Specifies whether to wrap the text to the width. Data type: Boolean
Others	
Data Mapping File	Specifies the data mapping file to the object for NLS use. For example, if the data mapping file name is Category_de_DE.properties, set the value as Category (without the language and locale part). Data type: String
Detail Report	Specifies the detail report that the object is linked to. Click ... in the value cell to set the detail report. For details, refer to Linking to a detail report. Data type: Boolean
Detail Target Frame	Specifies the target window or frame in which to display the detailed information. This property is enabled only when Go To Detail is set to true. Choose an option from the drop-down list. • **<Server Setting>** Loads the detailed information according to setting of the Pop Up New Window for Links option in the Customize Profile panel of the Profile dialog in JReport Server. • **New Window** Loads the detailed information into a new window. The window is not named. • **Whole Window** Loads the detailed information into the full browser window. • **Same Frame** Loads the detailed information into the same frame as the object. • **Parent Frame** Loads the detailed information into the parent frame of the frame in which the object is. • **Other Frame** Loads the detailed information into some other specified frame. If selected, you can also directly input the name of the frame you have defined in the text box. If the frame name does not exist, the detailed information will be loaded into a new window. Data type: String
Display Null	Specifies the string to be displayed if the field value is null.> Data type: String
Enable Hyperlink in Excel	Specifies whether to enable the link defined on the object when exporting the report to an Excel file. Data type: Boolean
Enable Hyperlink in HTML	Specifies whether to enable the link defined on the object when exporting the report to an HTML file. Data type: Boolean
Enable Hyperlink in PDF	Specifies whether to enable the link defined on the object when exporting the report to a PDF file. Data type: Boolean
Export to CSV	Specifies whether to include the object when exporting the report to CSV. Data type: Boolean
Export to Page Report Result	Specifies whether to include the object when exporting the report to Page Report Result. Data type: Boolean
Export to Excel	Specifies whether to include the object when exporting the report to Excel. Data type: Boolean
Export to PDF	Specifies whether to include the object when exporting the report to PDF. This property is available only when the object is displayed as a web control. Data type: Boolean
Go to Detail	Specifies whether to show the detailed information when you click the object in Page Report Studio. This property is available only when the object is in the group header/footer panel of a banded object. For details about its usage, refer to Obtaining detailed information from a banded object. Data type: Boolean
Invisible	Specifies whether to show the object in the design area and in the report results. All formulas and calculations will still be performed if the property is set to true. Data type: Boolean
Link	Specifies to link the object to another report, a website, an e-mail address or a Blob data type field. Click ... in the value cell to set the link target. For details, see Binding a link to a label. Data type: String
Logic Column	Specifies whether to show the object in the next visible table cell in the same row when the column which holds the object is hidden. Choose an option from the drop-down list. Data type: Enumeration **Notes:** • The next visible table cell should be completely empty, that is, the cell should not hold any content including blank space. • When the property for several objects in the same row is set to next visible column, and the columns holding these objects are all hidden, then only the object in the rightmost column will be shown in the next visible cell.
Position	Specifies the position of the object. Choose an option from the drop-down list. • **static** - The object is positioned at the location in which it is inserted. If selected, the X, Y and other position-related properties will be hidden or disabled. The position of the object is affected by preceding objects in the same container. • **relative** - The object is positioned as an offset to the position at which it is inserted. The offset is determined by the X and Y property values. • **absolute** - The object will be located at the position specified by dragging and dropping or by setting its X and Y property values. The position of the object is not affected by preceding objects in the same container. Data type: Enumeration
Parameter	Substitutes the current parameter with another one by selecting from the drop-down list. To make the property editable, uncheck Forbid changing parameter in the Options dialog (**File > Options > Panel > Forbid changing parameter**). Data type: Boolean
Record Location	Specifies the calculation point for the properties which use formulas. Choose an option from the drop-down list. • **Default** - The values of the properties are calculated in the default location where they are located. • **Page Header** - The values of the properties are calculated in the page header. • **Page Footer** - The values of the properties are calculated in the page footer. Data type: Enumeration
Suppress	Specifies whether to show the object in the design area and in the report results. All formulas and calculations will be skipped if the property is set to true. Data type: Boolean **Note:** When both the Invisible and Suppress properties of an object are set to true, Suppress has the higher priority.
Suppress When No Records	Specifies whether to display the object in the report results when no record is returned to its parent data container. Data type: Boolean
Suppress When Null	If true and its value is null, the field will not be displayed. Data type: Boolean
Transfer Style	Specifies whether the style group of the primary report will be applied to the linked report if the object is linked to a report. Data type: Boolean
Value Delimiter	Specifies the separator for the data whose type is DBArray. The default value space means that the elements will be displayed in a horizontal line and separated by a space. Data type: String
TOC	
Anchor Display Value	Specifies a string or formula to display for the TOC entry for this object in the TOC Browser. By default it is the value of the specified object. The TOC Anchor property must be set to true for this property to take effect. Data type: String
TOC Anchor	Specifies whether to include the object in the TOC Browser for the report. Data type: Boolean
Accessibility	
External AccessKey	It is mapped to the HTML attribute accesskey. This attribute specifies an access key to the object. Data type: String
External CSS Class Selector	Specifies a class selector to be applied to the object when exported as HTML. Enter a valid class name from the CSS file. Data type: String
External Dir	It is mapped to the HTML attribute dir. This attribute specifies the base direction of directionally neutral text in the object's content and attribute values. It also specifies the directionality of tables. Possible values: • LTR - Left-to-right text or table. • RTL - Right-to-left text or table. Data type: String
External ID	It is mapped to the HTML attribute id. This attribute specifies a name to the object, which must be unique in the report. Data type: String
External Style	It is mapped to the HTML attribute style. This attribute specifies style information for the object. Data type: String
External TabIndex	It is mapped to the HTML attribute tabindex. This attribute specifies the position of the object in the tabbing order for the current report. Enter an integer equal to or greater than 0 and less than 65535. Data type: Integer
HrefLang	It is mapped to the HTML attribute hreflang. This attribute specifies the base language of the resource designated by a link and may only be used when a link is specified. Data type: String
Language	It is mapped to the HTML attribute lang. This attribute specifies the base language of the object's attribute values and text content. Data type: String

Report body

The properties of the report body of a page report are:

Property Name	Description
General	
Class Type	Indicates the class type of the object. This property is read only.
Data Inherit	Indicates whether the dataset for this object is inherited from another object. This property is read only.
Dataset	Indicates the dataset used by the object. This property is read only.
Instance Name	Shows the instance name of the object. This property is read only.
Color	
Background	Specifies the background color and fill effect of the object. Choose a color from the drop-down list, enter a hexadecimal RGB value (for example, 0xff0000), choose/create a formula or edit an expression that returns a color, or select Custom from the drop-down list to specify the color and fill effect. Data type: String
CSS	
Class	Specifies a CSS class to be applied to the object which is a valid class in the CSS file. Data type: String
Style	The property can be used in two ways. • Specifies a style to be applied to the object. Choose a style from the drop-down list which is available when a Style Group has been selected at the report tab level and when there are styles in the Style Group that can be applied to the object, or type in the style name. • Specifies a CSS selector to be applied to the object. Enter a valid class name from the CSS file. Data type: String
Others	
Export to CSV	Specifies whether to include the object when exporting the report to CSV. Data type: Boolean
Export to Page Report Result	Specifies whether to include the object when exporting the report to Page Report Result. Data type: Boolean
Export to Excel	Specifies whether to include the object when exporting the report to Excel. Data type: Boolean
Record Location	Specifies the calculation point for the properties which use formulas. Choose an option from the drop-down list. • **Default** - The values of the properties are calculated in the default location where they are located. • **Page Header** - The values of the properties are calculated in the page header. • **Page Footer** - The values of the properties are calculated in the page footer. Data type: Enumeration
Suppress When Empty	Specifies whether to display the object in the report results when no record is returned to it. Data type: Boolean
Excel	
Column Index	Specifies the X coordinate of the object relative to its parent container when exported to Excel or CSV, measured in cells. The Columned property at the report tab level must be set to true for this property to take effect. Data type: Integer
Row Index	Specifies the Y coordinate of the object relative to its parent container when exported to Excel or CSV, measured in cells. The Columned property at the report tab level must be set to true for this property to take effect. Data type: Integer
Border	
Border Color	Specifies the color of the border of the object. Choose a color from the drop-down list, enter a hexadecimal RGB value (for example, 0xCCAA00), choose/create a formula or edit an expression that returns a color, or select Custom from the drop-down list to specify the color. Data type: String
Border Thickness	Specifies the width of the border in inches or centimeters. Enter a numeric value to change the thickness. Data type: Float
Bottom Line	Specifies the line style of the bottom border of the object. Choose a style from the drop-down list. Data type: Enumeration
Left Line	Specifies the line style of the left border of the object. Choose a style from the drop-down list. Data type: Enumeration
Right Line	Specifies the line style of the right border of the object. Choose a style from the drop-down list. Data type: Enumeration
Top Line	Specifies the line style of the top border of the object. Choose a style from the drop-down list. Data type: Enumeration
TOC	
TOC Anchor	Specifies whether to include the object in the TOC Browser for the report. Data type: Boolean
Accessibility	
External CSS Class Selector	Specifies a class selector to be applied to the object when exported as HTML. Enter a valid class name from the CSS file. Data type: String
External ID	It is mapped to the HTML attribute *id*. This attribute specifies a name to the object, which must be unique in the report. Data type: String
External Style	It is mapped to the HTML attribute *style*. This attribute specifies style information for the object. Data type: String
Language	It is mapped to the HTML attribute *lang*. This attribute specifies the base language of the object's attribute values and text content. Data type: String

Shape map

The properties of a shape map in a page report are:

Property Name	Description
General	
Class Type	Indicates the class type of the object. This property is read only.
Data Inherit	Indicates whether the dataset for this object is inherited from another object. This property is read only.
Dataset	Indicates the dataset used by the object. This property is read only.
Instance Name	Shows the instance name of the object. This property is read only.
Geometry	
Height	Specifies the height of the object, in inches or centimeters. Enter a numeric value to change the height. Data type: Float
Width	Specifies the width of the object, in inches or centimeters. Enter a numeric value to change the width. Data type: Float
X	Specifies the horizontal coordinate of the top left corner of the object, relative to its parent container. This property is ignored if the Position property is set to static. Enter a numeric value to change the position in inches or centimeters. Data type: Float
Y	Specifies the vertical coordinate of the top left corner of the object, relative to its parent container. This property is ignored if the Position property is set to static. Enter a numeric value to change the position in inches or centimeters. Data type: Float
Color	
Background	Specifies the background color of the object. Choose a color from the drop-down list, enter a hexadecimal RGB value (for example, 0xff0000), choose/create a formula or edit an expression that returns a color, or select Custom from the drop-down list to specify the color. Data type: String
Others	
Default for Filter	Specifies whether to show the data component as the default component in the Apply to drop-down list of the Filter dialog in Page Report Studio. Data type: Boolean **Note:** In a page report tab, only one data component's Default for Filter property can be set to true.
Detail Target Frame	Specifies the target window or frame in which to display the detailed information. This property is enabled only when Go To Detail is set to true. Choose an option from the drop-down list. • **<Server Setting>** Loads the detailed information according to setting of the Pop Up New Window for Links option in the Page Report Studio tab of the Profile dialog in JReport Server. • **New Window** Loads the detailed information into a new window. The window is not named. • **Whole Window** Loads the detailed information into the full browser window. • **Same Frame** Loads the detailed information into the same frame as the object. • **Parent Frame** Loads the detailed information into the parent frame of the frame in which the object is. • **Other Frame** Loads the detailed information into some other specified frame. If selected, you can also directly input the name of the frame you have defined in the text box. If the frame name does not exist, the detailed information will be loaded into a new window. Data type: String
Export to Excel	Specifies whether to include the object when exporting the report to Excel. Data type: Boolean
Export to Page Report Result	Specifies whether to include the object when exporting the report to Page Report Result. Data type: Boolean
Go to Detail	Specifies whether to show the detailed information when you click the object in Page Report Studio. This property is available only when the object is in the group header/ footer panel of a banded object. For details about its usage, refer to Obtaining detailed information from a banded object. Data type: Boolean
Invisible	Specifies whether to show the object in the design area and in the report results. All formulas and calculations will still be performed if the property is set to true. Data type: Boolean
Invisible for Filter Dialogs	Specifies whether to show the data component in the Apply to drop-down list of the Filter dialog in Page Report Studio. This property cannot be edited when Default for Filter is set to true. Data type: Boolean
Position	Specifies the position of the object. Choose an option from the drop-down list. • **static** - The object is positioned at the location in which it is inserted. If selected, the X, Y and other position-related properties will be hidden or disabled. The position of the object is affected by preceding objects in the same container. • **relative** - The object is positioned at an offset to the position at which it is inserted. The offset is determined by the X and Y property values. • **absolute** - The object will be located at the position specified by dragging and dropping or by setting its X and Y property values. The position of the object is not affected by preceding objects in the same container. Data type: Enumeration
Suppress	Specifies whether to show the object in the design area and in the report results. All formulas and calculations will be skipped if the property is set to true. Data type: Boolean **Note:** When both the Invisible and Suppress properties of an object are set to true, Suppress has the higher priority.
Suppress When No Records	Specifies whether to display the object in the report results when no record is returned to its parent data container. Data type: Boolean
Excel	
Column Index	Specifies the X coordinate of the object relative to its parent container when exported to Excel or CSV, measured in cells. The Columned property at the report tab level must be set to true for this property to take effect. Data type: Integer
Row Index	Specifies the Y coordinate of the object relative to its parent container when exported to Excel or CSV, measured in cells. The Columned property at the report tab level must be set to true for this property to take effect. Data type: Integer
Border	
Border Color	Specifies the color of the border of the object. Choose a color from the drop-down list, enter a hexadecimal RGB value (for example, 0xCCAA00), choose/create a formula or edit an expression that returns a color, or select Custom from the drop-down list to specify the color. Data type: String
Border Thickness	Specifies the width of the border in inches or centimeters. Enter a numeric value to change the thickness. Data type: Float
Bottom Line	Specifies the line style of the bottom border of the object. Choose a style from the drop-down list. Data type: Enumeration
Left Line	Specifies the line style of the left border of the object. Choose a style from the drop-down list. Data type: Enumeration
Right Line	Specifies the line style of the right border of the object. Choose a style from the drop-down list. Data type: Enumeration
Top Line	Specifies the line style of the top border of the object. Choose a style from the drop-down list. Data type: Enumeration
Map	
Alternate Content Type	Specifies the tip type of the areas that have data. The tip is displayed when you hover the mouse pointer over the area in HTML result or in Page Report Studio. Choose an option from the drop-down list. • **name** - If selected, names of the areas will be used as the area tips. • **value** - If selected, the summary values on the areas will be used as the area tips. If there is more than one value on an area, the summary values will be separated by comma. • **customized** - If selected, you can specify the tip for each area via the Alternate Text property of the area. Data type: Enumeration
Column Name	Displays the mapping name of the field used as the image source. This property is read only.
Image Source	Specifies the source of the map image. Data type: String
Name	Specifies the name of the map. Data type: String
TOC	
Anchor Display Value	Specifies a string or formula to display for the TOC entry for this object in the TOC Browser. By default it is the value of the specified object. The TOC Anchor property must be set to true for this property to take effect. Data type: String
TOC Anchor	Specifies whether to include the object in the TOC Browser for the report. Data type: Boolean
Accessibility	
External CSS Class Selector	Specifies a class selector to be applied to the object when exported as HTML. Enter a valid class name from the CSS file. Data type: String
External ID	It is mapped to the HTML attribute *id*. This attribute specifies a name to the object, which must be unique in the report. Data type: String
External Style	It is mapped to the HTML attribute *style*. This attribute specifies style information for the object. Data type: String
Language	It is mapped to the HTML attribute *lang*. This attribute specifies the base language of the object's attribute values and text content. Data type: String

A shape map has some child objects, the properties of which will be shown in the following topics:

- Area
- Label
- Summary field
- Line

Area

The properties of a shape map area in a page report are listed as follows. You can set these properties globally for all the map areas, or respectively for each specific map area.

Property Name	Description
General	
Class Type	Indicates the class type of the object. This property is read only.
Data Inherit	Indicates whether the dataset for this object is inherited from another object. This property is read only.
Dataset	Indicates the dataset used by the object. This property is read only.
Instance Name	Shows the instance name of the object. This property is read only.
Use Global Setting	Specifies whether to apply the settings that you have set globally for all the map areas to this object. • **true** - If selected, the object will take the global settings, and all the properties for the object will become read only. • **false**- If selected, the properties for the object can be edited if required. Data type: Boolean
Area	
Alternate Text	Specifies the tip of the area, which is displayed when you hover the mouse pointer over the area in HTML result or in Page Report Studio. This property takes effect only when Alternate Content Type on the map object is set to customized and the area has data. Data type: String
Detail Report	Specifies the detail report that the object is linked to. Click ... in the value cell to set the detail report. For details, refer to Linking to a detail report. Data type: String
Fill Color	Specifies the color with which to fill the area which has records. Choose a color from the drop-down list, enter a hexadecimal RGB value (for example, 0xff0000), choose/create a formula or edit an expression that returns a color, or select Custom from the drop-down list to specify the color and fill effect. Data type: String
Link	Specifies to link the object to another report, a website, an e-mail address or a Blob data type field. Click ... in the value cell to set the link target. For details, see Binding a link to a label. Data type: String
Name	Specifies the name of the area. Data type: String
Shape	
Coordinates	Indicates the coordinates of the area for laying out the map. The coordinates are displayed as: X1,Y1,X2,Y2,X3,Y3,... (X/Y: The X/Y coordinate of one of the polygon's corner.) This property is read only.
Type	Indicates the shape of the area. This property is read only.
Boundary	
Boundary Color	Specifies the color of the area border. Choose a color from the drop-down list, enter a hexadecimal RGB value (for example, 0xff0000), choose/create a formula or edit an expression that returns a color, or select Custom from the drop-down list to specify the color and fill effect. Data type: String
Boundary Style	Specifies the line style of the area border. Choose an option from the drop-down list. Data type: Enumeration
Boundary Width	Specifies the width of the area border. Enter a numeric value to change the width. Data type: Float
Accessibility	
External AccessKey	It is mapped to the HTML attribute *accesskey*. This attribute specifies an access key to the object. Data type: String
External CSS Class Selector	Specifies a class selector to be applied to the object when exported as HTML. Enter a valid class name from the CSS file. Data type: String
External TabIndex	It is mapped to the HTML attribute *tabindex*. This attribute specifies the position of the object in the tabbing order for the current report. Enter an integer equal to or greater than 0 and less than 65535. Data type: Integer
External Title	It is mapped to the HTML attribute *title*. This attribute offers advisory information about the object. Data type: String
HrefLang	It is mapped to the HTML attribute *hreflang*. This attribute specifies the base language of the resource designated by a link and may only be used when a link is specified. Data type: String

Label

The properties of a label in a shape map of a page report are listed as follows. You can set these properties globally for all the labels in a shape map, or respectively for the label in each specific shape map area, or for the one independently inserted in the shape map object.

Property Name	Description
General	
Class Type	Indicates the class type of the object. This property is read only.
Instance Name	Shows the instance name of the object. This property is read only.
Use Global Setting	Specifies whether to apply the settings that you have set globally for all labels in the map to this object. • **true** - If selected, the object will take the global settings, and all the properties for the object will become read only. • **false**- If selected, the properties for the object can be edited if required. Data type: Boolean
Text Format	
Auto Fit	Specifies whether to adjust the width and height of the object according to the contents. Data type: Boolean
Bold	Specifies whether to make the text bold. Data type: Boolean
Font Face	Specifies the font of the text. Choose an option from the drop-down list. Data type: Enumeration
Font Size	Specifies the font size of the text. Enter an integer value to change the size. Data type: Integer
Horizontal Alignment	Specifies the horizontal justification of the text in the object. Choose an option from the drop-down list. Data type: Enumeration
Italic	Specifies whether to make the text italic. Data type: Boolean
Maximum Width	Specifies the maximum width of the text displayed. Enter a numeric value to change the width. This property often works with the Auto Fit property. If Auto Fit = true and Maximum Width is not equal to 0, the text will extend until the width is this value. Data type: Float
Strikethrough	Specifies whether to add a strikethrough line to the text. Data type: Boolean
Text	Specifies the text in the label. Enter a string to display as the label text. Data type: String
Underline	Specifies whether to underline the text. Data type: Boolean
Vertical Alignment	Specifies the vertical justification of the text in the object. Choose an option from the drop-down list. Data type: Enumeration
Word Wrap	Specifies whether to wrap the text to the width. Data type: Boolean
Geometry	
Height	Specifies the height of the object, in inches or centimeters. Enter a numeric value to change the height. Data type: Float
Width	Specifies the width of the object, in inches or centimeters. Enter a numeric value to change the width. Data type: Float
X	Specifies the horizontal coordinate of the top left corner of the object, relative to its parent container. Enter a numeric value to change the position in inches or centimeters. Data type: Float
Y	Specifies the vertical coordinate of the top left corner of the object, relative to its parent container. Enter a numeric value to change the position in inches or centimeters. Data type: Float
Color	
Background	Specifies the background color of the object. Choose a color from the drop-down list, enter a hexadecimal RGB value (for example, 0xff0000), choose/create a formula or edit an expression that returns a color, or select Custom from the drop-down list to specify the color. Data type: String
Foreground	Specifies the foreground color of the object. Choose a color from the drop-down list, enter a hexadecimal RGB value (for example, 0xff0000), choose/create a formula or edit an expression that returns a color, or select Custom from the drop-down list to specify the color. Data type: String
Padding	
Bottom Padding	Specifies the space in inches or centimeters between the text and the bottom border of the object. Enter a numeric value to change the padding. Data type: Float
Left Padding	Specifies the space in inches or centimeters between the text and the left border of the object. Enter a numeric value to change the padding. Data type: Float
Right Padding	Specifies the space in inches or centimeters between the text and the right border of the object. Enter a numeric value to change the padding. Data type: Float
Top Padding	Specifies the space in inches or centimeters between the text and the top border of the object. Enter a numeric value to change the padding. Data type: Float
Border	
Border Color	Specifies the color of the border of the object. Choose a color from the drop-down list, enter a hexadecimal RGB value (for example, 0xCCAA00), choose/create a formula or edit an expression that returns a color, or select Custom from the drop-down list to specify the color. Data type: String
Border Thickness	Specifies the width of the border in inches or centimeters. Enter a numeric value to change the thickness. Data type: Float
Bottom Line	Specifies the line style of the bottom border of the object. Choose a style from the drop-down list. Data type: Enumeration
Left Line	Specifies the line style of the left border of the object. Choose a style from the drop-down list. Data type: Enumeration
Right Line	Specifies the line style of the right border of the object. Choose a style from the drop-down list. Data type: Enumeration
Shadow	Specifies whether to draw two shadowed borders, beneath and to the right of the object. Data type: Boolean
Shadow Color	Specifies the color of the shadow. Choose a color from the drop-down list, enter a hexadecimal RGB value (for example, 0xff0000), choose/create a formula or edit an expression that returns a color, or select Custom from the drop-down list to specify the color and fill effect. Data type: String
Top Line	Specifies the line style of the top border of the object. Choose a style from the drop-down list. Data type: Enumeration
Pattern	
Pattern Color	Specifies the color in which to draw a pattern to fill the object. Choose a color from the drop-down list, enter a hexadecimal RGB value (for example, 0xff0000), choose/create a formula or edit an expression that returns a color, or select Custom from the drop-down list to specify the color and fill effect. Data type: String
Pattern Style	Specifies the style of the pattern. Choose an option from the drop-down list. • **none** - No pattern will be applied to the object. • **50**% - The object will be filled in the Pattern Color of 50%-transparency. • **horizontal** - The pattern will be made of horizontal lines of the Pattern Color. • **vertical** - The pattern will be made of vertical lines of the Pattern Color. • **grid** - The pattern will be an overlapping of horizontal and vertical lines of the Pattern Color. • **diagonal** - The pattern will be made of diagonal lines of the Pattern Color. Data type: Enumeration
Others	
Auto Connector	Specifies whether a line between the area and the object will be automatically drawn, when the object is dragged and dropped out of the area. Data type: Boolean
Invisible	Specifies whether to show the object in the design area and in the report results. All formulas and calculations will still be performed if the property is set to true. Data type: Boolean
Accessibility	
External AccessKey	It is mapped to the HTML attribute *accesskey*. This attribute specifies an access key to the object. Data type: String
External CSS Class Selector	Specifies a class selector to be applied to the object when exported as HTML. Enter a valid class name from the CSS file. Data type: String
External Dir	It is mapped to the HTML attribute *dir*. This attribute specifies the base direction of directionally neutral text in the object's content and attribute values. It also specifies the directionality of tables. Possible values: • LTR - Left-to-right text or table. • RTL - Right-to-left text or table. Data type: String
External ID	It is mapped to the HTML attribute *id*. This attribute specifies a name to the object, which must be unique in the report. Data type: String
External Style	It is mapped to the HTML attribute *style*. This attribute specifies style information for the object. Data type: String
External TabIndex	It is mapped to the HTML attribute *tabindex*. This attribute specifies the position of the object in the tabbing order for the current report. Enter an integer equal to or greater than 0 and less than 65535. Data type: Integer
External Title	It is mapped to the HTML attribute *title*. This attribute offers advisory information about the object. Data type: String
HrefLang	It is mapped to the HTML attribute *hreflang*. This attribute specifies the base language of the resource designated by a link and may only be used when a link is specified. Data type: String
Language	It is mapped to the HTML attribute *lang*. This attribute specifies the base language of the object's attribute values and text content. Data type: String
LongDesc	It is mapped to the HTML attribute *longdesc*. This attribute specifies a link to a long description of the object. Data type: String

Summary field

The properties of a summary field in a shape map of a page report are listed as follows. You can set these properties globally for all the summary fields in a shape map, or respectively for the summary field in each specific shape map area.

Property Name	Description
General	
Aggregate Function	Shows the function of the summary. This property is read only.
Field Type	Shows what kind of field it is. This property is read only.
Group By	Shows the group field that the summary is based on. If null, the summary is grouped based on the whole dataset. This property is read only.
Instance Name	Shows the instance name of the object. This property is read only.
Summary Name	Shows the name of the summary. This property is read only.
Summary On	Shows the name of the field on which to perform the summary function. This property is read only.
Use Global Setting	Specifies whether to apply the settings that you have set globally for all summary fields in the map to this object. • **true** - If selected, the object will take the global settings, and all the properties for the object will become read only. • **false**- If selected, the properties for the object can be edited if required. Data type: Boolean
Geometry	
Height	Specifies the height of the object, in inches or centimeters. Enter a numeric value to change the height. Data type: Float
Width	Specifies the width of the object, in inches or centimeters. Enter a numeric value to change the width. Data type: Float
X	Specifies the horizontal coordinate of the top left corner of the object, relative to its parent container. Enter a numeric value to change the position in inches or centimeters. Data type: Float
Y	Specifies the vertical coordinate of the top left corner of the object, relative to its parent container. Enter a numeric value to change the position in inches or centimeters. Data type: Float
Color	
Background	Specifies the background color of the object. Choose a color from the drop-down list, enter a hexadecimal RGB value (for example, 0xff0000), choose/create a formula or edit an expression that returns a color, or select Custom from the drop-down list to specify the color. Data type: String
Foreground	Specifies the foreground color of the object. Choose a color from the drop-down list, enter a hexadecimal RGB value (for example, 0xff0000), choose/create a formula or edit an expression that returns a color, or select Custom from the drop-down list to specify the color. Data type: String
Padding	
Bottom Padding	Specifies the space in inches or centimeters between the text and the bottom border of the object. Enter a numeric value to change the padding. Data type: Float
Left Padding	Specifies the space in inches or centimeters between the text and the left border of the object. Enter a numeric value to change the padding. Data type: Float
Right Padding	Specifies the space in inches or centimeters between the text and the right border of the object. Enter a numeric value to change the padding. Data type: Float
Top Padding	Specifies the space in inches or centimeters between the text and the top border of the object. Enter a numeric value to change the padding. Data type: Float
Border	
Border Color	Specifies the color of the border of the object. Choose a color from the drop-down list, enter a hexadecimal RGB value (for example, 0xCCAA00), choose/create a formula or edit an expression that returns a color, or select Custom from the drop-down list to specify the color. Data type: String
Border Thickness	Specifies the width of the border in inches or centimeters. Enter a numeric value to change the thickness. Data type: Float
Bottom Line	Specifies the line style of the bottom border of the object. Choose a style from the drop-down list. Data type: Enumeration
Left Line	Specifies the line style of the left border of the object. Choose a style from the drop-down list. Data type: Enumeration
Right Line	Specifies the line style of the right border of the object. Choose a style from the drop-down list. Data type: Enumeration
Shadow	Specifies whether to draw two shadowed borders, beneath and to the right of the object. Data type: Boolean
Shadow Color	Specifies the color of the shadow. Choose a color from the drop-down list, enter a hexadecimal RGB value (for example, 0xff0000), choose/create a formula or edit an expression that returns a color, or select Custom from the drop-down list to specify the color and fill effect. Data type: String
Top Line	Specifies the line style of the top border of the object. Choose a style from the drop-down list. Data type: Enumeration
Pattern	
Pattern Color	Specifies the color in which to draw a pattern to fill the object. Choose a color from the drop-down list, enter a hexadecimal RGB value (for example, 0xff0000), choose/create a formula or edit an expression that returns a color, or select Custom from the drop-down list to specify the color and fill effect. Data type: String
Pattern Style	Specifies the style of the pattern. Choose an option from the drop-down list. • **none** - No pattern will be applied to the object. • **50**% - The object will be filled in the Pattern Color of 50%-transparency. • **horizontal** - The pattern will be made of horizontal lines of the Pattern Color. • **vertical** - The pattern will be made of vertical lines of the Pattern Color. • **grid** - The pattern will be an overlapping of horizontal and vertical lines of the Pattern Color. • **diagonal** - The pattern will be made of diagonal lines of the Pattern Color. Data type: Enumeration
Text Format	
Auto Fit	Specifies whether to adjust the width and height of the object according to the contents. Data type: Boolean
Bold	Specifies whether to make the text bold. Data type: Boolean
Font Face	Specifies the font of the text. Choose an option from the drop-down list. Data type: Enumeration
Font Size	Specifies the font size of the text. Enter an integer value to change the size. Data type: Integer
Format	Specifies the display format of the text in the report result. It varies with data type and can be manually defined. Data type: String **Note:** For the BigDecimal type, to avoid precision loss, you should enter a prefix JRD when setting the format property value.
Horizontal Alignment	Specifies the horizontal justification of the text in the object. Choose an option from the drop-down list. Data type: Enumeration
Italic	Specifies whether to make the text italic. Data type: Boolean
Maximum Width	Specifies the maximum width of the text displayed. Enter a numeric value to change the width. This property often works with the Auto Fit property. If Auto Fit = true and Maximum Width is not equal to 0, the text will extend until the width is this value. Data type: Float
Strikethrough	Specifies whether to add a strikethrough line to the text. Data type: Boolean
Underline	Specifies whether to underline the text. Data type: Boolean
Vertical Alignment	Specifies the vertical justification of the text in the object. Choose an option from the drop-down list. Data type: Enumeration
Word Wrap	Specifies whether to wrap the text to the width. Data type: Boolean
Others	
Auto Connector	Specifies whether a line between the area and the object will be automatically drawn, when the object is dragged and dropped out of the area. Data type: Boolean
Cache Value	Specifies whether to cache the value of the field instead of obtaining it repeatedly. Data type: Boolean
Column Name	Substitutes the current field with another field by selecting from the drop-down list. To make the property editable, uncheck Forbid changing column in the Options dialog (**File** > **Options** > **Panel** > **Forbid changing column**). Data type: String
Data Mapping File	Specifies the data mapping file to the object for NLS use. For example, if the data mapping file name is Category_de_DE.properties, set the value as Category (without the language and locale part). Data type: String
Display Null	Specifies the string to be displayed if the field value is null. Data type: String
Invisible	Specifies whether to show the object in the design area and in the report results. All formulas and calculations will still be performed if the property is set to true. Data type: Boolean
Accessibility	
External AccessKey	It is mapped to the HTML attribute *accesskey*. This attribute specifies an access key to the object. Data type: String
External CSS Class Selector	Specifies a class selector to be applied to the object when exported as HTML. Enter a valid class name from the CSS file. Data type: String
External Dir	It is mapped to the HTML attribute *dir*. This attribute specifies the base direction of directionally neutral text in the object's content and attribute values. It also specifies the directionality of tables. Possible values: • LTR - Left-to-right text or table. • RTL - Right-to-left text or table. Data type: String
External ID	It is mapped to the HTML attribute *id*. This attribute specifies a name to the object, which must be unique in the report. Data type: String
External Style	It is mapped to the HTML attribute *style*. This attribute specifies style information for the object. Data type: String
External TabIndex	It is mapped to the HTML attribute *tabindex*. This attribute specifies the position of the object in the tabbing order for the current report. Enter an integer equal to or greater than 0 and less than 65535. Data type: Integer
External Title	It is mapped to the HTML attribute *title*. This attribute offers advisory information about the object. Data type: String
HrefLang	It is mapped to the HTML attribute *hreflang*. This attribute specifies the base language of the resource designated by a link and may only be used when a link is specified. Data type: String
Language	It is mapped to the HTML attribute *lang*. This attribute specifies the base language of the object's attribute values and text content. Data type: String
LongDesc	It is mapped to the HTML attribute *longdesc*. This attribute specifies a link to a long description of the object. Data type: String

Line

The properties of a line object in a shape map of a page report are listed as follows:

Property Name	Description
General	
Class Type	Indicates the class type of the object. This property is read only.
Instance Name	Shows the instance name of the object. This property is read only.
Others	
Invisible	Specifies whether to show the object in the design area and in the report results. All formulas and calculations will still be performed. Data type: Boolean
Line Property	
Line Color	Specifies the color of the line. Choose a color from the drop-down list, enter a hexadecimal RGB value (for example, 0xff0000), choose/create a formula or edit an expression that returns a color, or select Custom from the drop-down list to specify the color and fill effect. Data type: String
Line Style	Specifies the style of the line. Choose an option from the drop-down list. Data type: Enumeration
Line Thickness	Specifies the thickness of the line. Enter a numeric value to change the thickness. Data type: Float
Accessibility	
External CSS Class Selector	Specifies a class selector to be applied to the object when exported as HTML. Enter a valid class name from the CSS file. Data type: String
External ID	It is mapped to the HTML attribute *id*. This attribute specifies a name to the object, which must be unique in the report. Data type: String
External Style	It is mapped to the HTML attribute *style*. This attribute specifies style information for the object. Data type: String
Language	It is mapped to the HTML attribute *lang*. This attribute specifies the base language of the object's attribute values and text content. Data type: String

Special field

The properties of a special field in a page report are:

Property Name	Description
General	
Class Type	Indicates the class type of the object. This property is read only.
Instance Name	Shows the instance name of the object. This property is read only.
Geometry	
Height	Specifies the height of the object, in inches or centimeters. Enter a numeric value to change the height. Data type: Float
Width	Specifies the width of the object, in inches or centimeters. Enter a numeric value to change the width. Data type: Float
X	Specifies the horizontal coordinate of the top left corner of the object, relative to its parent container. This property is ignored if the Position property is set to static. Enter a numeric value to change the position in inches or centimeters. Data type: Float
Y	Specifies the vertical coordinate of the top left corner of the object, relative to its parent container. This property is ignored if the Position property is set to static. Enter a numeric value to change the position in inches or centimeters. Data type: Float
Color	
Background	Specifies the background color of the object. Choose a color from the drop-down list, enter a hexadecimal RGB value (for example, 0xff0000), choose/create a formula or edit an expression that returns a color, or select Custom from the drop-down list to specify the color. Data type: String
Foreground	Specifies the foreground color of the object. Choose a color from the drop-down list, enter a hexadecimal RGB value (for example, 0xff0000), choose/create a formula or edit an expression that returns a color, or select Custom from the drop-down list to specify the color. Data type: String
CSS	
Class	Specifies a CSS class to be applied to the object which is a valid class in the CSS file. Data type: String
ID	Specifies the identifier of the object, which must be unique in the report. The ID property can be a style sheet selector. Data type: String
Style	The property can be used in two ways. • Specifies a style to be applied to the object. Choose a style from the drop-down list which is available when a Style Group has been selected at the report tab level and when there are styles in the Style Group that can be applied to the object, or type in the style name. • Specifies a CSS selector to be applied to the object. Enter a valid class name from the CSS file. Data type: String
Excel	
Column Index	Specifies the X coordinate of the object relative to its parent container when exported to Excel or CSV, measured in cells. The Columned property at the report lab level must be set to true for this property to take effect. Data type: Integer
Row Index	Specifies the Y coordinate of the object relative to its parent container when exported to Excel or CSV, measured in cells. The Columned property at the report lab level must be set to true for this property to take effect. Data type: Integer
Padding	
Bottom Padding	Specifies the space in inches or centimeters between the text and the bottom border of the object. Enter a numeric value to change the padding. Data type: Float
Left Padding	Specifies the space in inches or centimeters between the text and the left border of the object. Enter a numeric value to change the padding. Data type: Float
Right Padding	Specifies the space in inches or centimeters between the text and the right border of the object. Enter a numeric value to change the padding. Data type: Float
Top Padding	Specifies the space in inches or centimeters between the text and the top border of the object. Enter a numeric value to change the padding. Data type: Float
Border	
Border Color	Specifies the color of the border of the object. Choose a color from the drop-down list, enter a hexadecimal RGB value (for example, 0xCCAADD), choose/create a formula or edit an expression that returns a color, or select Custom from the drop-down list to specify the color. Data type: String
Border Thickness	Specifies the width of the border in inches or centimeters. Enter a numeric value to change the thickness. Data type: Float
Bottom Line	Specifies the line style of the bottom border of the object. Choose a style from the drop-down list. Data type: Enumeration
Left Line	Specifies the line style of the left border of the object. Choose a style from the drop-down list. Data type: Enumeration
Right Line	Specifies the line style of the right border of the object. Choose a style from the drop-down list. Data type: Enumeration
Shadow	Specifies whether to draw two shadowed borders, beneath and to the right of the object. Data type: Boolean
Shadow Color	Specifies the color of the shadow. Choose a color from the drop-down list, enter a hexadecimal RGB value (for example, 0xff0000), choose/create a formula or edit an expression that returns a color, or select Custom from the drop-down list to specify the color and fill effect. Data type: String
Top Line	Specifies the line style of the top border of the object. Choose a style from the drop-down list. Data type: Enumeration
Pattern	
Pattern Color	Specifies the color in which to draw a pattern to fill the object. Choose a color from the drop-down list, enter a hexadecimal RGB value (for example, 0xff0000), choose/create a formula or edit an expression that returns a color, or select Custom from the drop-down list to specify the color and fill effect. Data type: String
Pattern Style	Specifies the style of the pattern. Choose an option from the drop-down list. • **none** - No pattern will be applied to the object. • **50%** - The object will be filled in the Pattern Color of 50%-transparency. • **horizontal** - The pattern will be made of horizontal lines of the Pattern Color. • **vertical** - The pattern will be made of vertical lines of the Pattern Color. • **grid** - The pattern will be an overlapping of horizontal and vertical lines of the Pattern Color. • **diagonal** - The pattern will be made of diagonal lines of the Pattern Color. Data type: Enumeration
Text Format	
Auto Fit	Specifies whether to adjust the width and height of the object according to the contents. Data type: Boolean
Bold	Specifies whether to make the text bold. Data type: Boolean
Font Face	Specifies the font of the text. Choose an option from the drop-down list. Data type: Enumeration
Font Size	Specifies the font size of the text. Enter an integer value to change the size. Data type: Integer
Format	Specifies the display format of the text in the report result. It varies with data type and can be manually defined. Data type: String **Note:** For the BigDecimal type, to avoid precision loss, you should enter a prefix JRD when setting the format property value.
Horizontal Alignment	Specifies the horizontal justification of the text in the object. Choose an option from the drop-down list. Data type: Enumeration
Ignore HTML Tag	Specifies whether to ignore the HTML tags when exporting the report results to HTML format. • **true** - The text will appear in the HTML file the same as that in JReport Designer (HTML tag elements in the text, if any, will not be parsed). • **false** - HTML tag elements in the text will be parsed when exporting the report results to HTML format. This property is useful for exporting the report results to HTML format. You may want to write the label contents in HTML tags in your report, and for the exported HTML files, the tags will be transferred as they are translated into HTML. Data type: Boolean
Italic	Specifies whether to make the text italic. Data type: Boolean
Maximum Width	Specifies the maximum width of the text displayed. Enter a numeric value to change the width. This property often works with the Auto Fit property. If Auto Fit = true and Maximum Width is not equal to 0, the text will extend until the width is this value. Data type: Float
Strikethrough	Specifies whether to add a strikethrough line to the text. Data type: Boolean
Underline	Specifies whether to underline the text. Data type: Boolean
Vertical Alignment	Specifies the vertical justification of the text in the object. Choose an option from the drop-down list. Data type: Enumeration
Word Wrap	Specifies whether to wrap the text to the width. Data type: Boolean
Others	
Data Evaluation Setting	Specifies the group information for the object. Available only for the group by fields in table group columns. Choose an option from the drop-down list. • **current column** The object will take value of the group by field in the current column. • **current row** The object will take value of the group by field in the current row. Data type: Enumeration
Data Mapping File	Specifies the data mapping file to the object for NLS use. For example, if the data mapping file name is Category_de_DE.properties, set the value as Category (without the language and locale part). Data type: String
Detail Report	Specifies the detail report that the object is linked to. Click [...] in the value cell to set the detail report. For details, refer to Linking to a detail report.
Detail Target Frame	Specifies the target window or frame in which to display the detailed information. This property is enabled only when Go To Detail is set to true. Choose an option from the drop-down list. • **<Server Setting>** Loads the detailed information according to setting of the Pop Up New Window for Links option in the Customize Profile panel of the Profile dialog in JReport Server. • **New Window** Loads the detailed information into a new window. The window is not named. • **Whole Window** Loads the detailed information into the full browser window. • **Same Frame** Loads the detailed information into the same frame as the object. • **Parent Frame** Loads the detailed information into the parent frame of the frame in which the object is. • **Other Frame** Loads the detailed information into some other specified frame. If selected, you can also directly input the name of the frame you have defined in the text box. If the frame name does not exist, the detailed information will be loaded into a new window. Data type: String
Display Null	Specifies the string to be displayed if the field value is null. Data type: String
Enable Hyperlink in Excel	Specifies whether to enable the link defined on the object when exporting the report to an Excel file. Data type: Boolean
Enable Hyperlink in HTML	Specifies whether to enable the link defined on the object when exporting the report to an HTML file. Data type: Boolean
Enable Hyperlink in PDF	Specifies whether to enable the link defined on the object when exporting the report to a PDF file. Data type: Boolean
Export to CSV	Specifies whether to include the object when exporting the report to CSV. Data type: Boolean
Export to Page Report Result	Specifies whether to include the object when exporting the report to Page Report Result. Data type: Boolean
Export to Excel	Specifies whether to include the object when exporting the report to Excel. Data type: Boolean
Export to PDF	Specifies whether to include the object when exporting the report to PDF. This property is available only when the object is displayed as a web control. Data type: Boolean
Field Type	Specifies another special field format to substitute the current special field. Choose an option from the drop-down list. Data type: Enumeration
Go to Detail	Specifies whether to show the detailed information when you click the object in Page Report Studio. This property is available only when the object is in the group header/ footer panel of a banded object. For details about its usage, refer to Obtaining detailed information from a banded object. Data type: Boolean
Invisible	Specifies whether to show the object in the design area and in the report results. All formulas and calculations will still be performed if the property is set to true. Data type: Boolean
Link	Specifies to link the object to another report, a website, an e-mail address or a Blob data type field. Click [...] in the value cell to set the link target. For details, see Binding a link to a label. Data type: String
Logic Column	Specifies whether to show the object in the next visible table cell in the same row when the column which holds the object is hidden. Choose an option from the drop-down list. Data type: Enumeration **Notes:** • The next visible table cell should be completely empty, that is, the cell should not hold any content including blank space. • When the property for several objects in the same row is set to next visible column, and the columns holding these objects are all hidden, then only the object in the rightmost column will be shown in the next visible cell.
Position	Specifies the position of the object. Choose an option from the drop-down list. • **static** - The object is positioned at the location in which it is inserted. If selected, the X, Y and other position-related properties will be hidden or disabled. The position of the object is affected by preceding objects in the same container. • **relative** - The object is positioned at an offset to the position at which it is inserted. The offset is determined by the X and Y property values. • **absolute** - The object will be located at the position specified by dragging and dropping or by setting its X and Y property values. The position of the object is not affected by preceding objects in the same container. Data type: Enumeration
Record Location	Specifies the calculation point for the properties which use formulas. Choose an option from the drop-down list. • **Default** - The values of the properties are calculated in the default location where they are located. • **Page Header** - The values of the properties are calculated in the page header. • **Page Footer** - The values of the properties are calculated in the page footer. Data type: Enumeration
Repeat	Specifies whether to repeat the group name in the report results. This property is available only for the group by fields in table, and it takes effect only when the group by fields are in the table detail row. Data type: Boolean
Suppress	Specifies whether to show the object in the design area and in the report results. All formulas and calculations will be skipped if the property is set to true. Data type: Boolean **Note:** When both the Invisible and Suppress properties of an object are set to true, Suppress has the higher priority.
Suppress When No Records	Specifies whether to display the object in the report results when no record is returned to its parent data container. Data type: Boolean
Transfer Style	Specifies whether the style group of the primary report will be applied to the linked report if the object is linked to a report. Data type: Boolean
TOC	
Anchor Display Value	Specifies a text or formula to display for the TOC entry for this object in the TOC Browser. By default it is the value of the specified object. The TOC Anchor property must be set to true for this property to take effect. Data type: String
TOC Anchor	Specifies whether to include the object in the TOC Browser for the report. Data type: Boolean
Accessibility	
External Accessekey	It is mapped to the HTML attribute accesskey. This attribute specifies an access key to the object. Data type: String
External CSS Class Selector	Specifies a class selector to be applied to the object when exported as HTML. Enter a valid class name from the CSS file. Data type: String
External Dir	It is mapped to the HTML attribute dir. This attribute specifies the base direction of directionally neutral text in the object's content and attribute values. It also specifies the directionality of tables. Possible values: • LTR - Left-to-right text or table. • RTL - Right-to-left text or table. Data type: String
External ID	It is mapped to the HTML attribute id. This attribute specifies a name to the object, which must be unique in the report. Data type: String
External Style	It is mapped to the HTML attribute style. This attribute specifies style information for the object. Data type: String
External Tabindex	It is mapped to the HTML attribute tabindex. This attribute specifies the position of the object in the tabbing order for the current report. Enter an integer equal to or greater than 0 and less than 65535. Data type: Integer
External Title	It is mapped to the HTML attribute title. This attribute offers advisory information about the object. Data type: String
HrefLang	It is mapped to the HTML attribute hreflang. This attribute specifies the base language of the resource designated by a link and may only be used when a link is specified. Data type: String
Language	It is mapped to the HTML attribute lang. This attribute specifies the base language of the object's attribute values and text content. Data type: String
LongDesc	It is mapped to the HTML attribute longdesc. This attribute specifies a link to a long description of the object. Data type: String

Subreport

The properties of a subreport in a page report are:

Property Name	Description
General	
Class Type	Indicates the class type of the object. This property is read only.
Data Inherit	Indicates whether the dataset for this object is inherited from another object. This property is read only.
Dataset	Indicates the dataset used by the object. This property is read only.
Instance Name	Shows the instance name of the object. This property is read only.
Geometry	
Height	Specifies the height of the object, in inches or centimeters. Enter a numeric value to change the height. Data type: Float
Width	Specifies the width of the object, in inches or centimeters. Enter a numeric value to change the width. Data type: Float
X	Specifies the horizontal coordinate of the top left corner of the object, relative to its parent container. This property is ignored if the Position property is set to static. Enter a numeric value to change the position in inches or centimeters. Data type: Float
Y	Specifies the vertical coordinate of the top left corner of the object, relative to its parent container. This property is ignored if the Position property is set to static. Enter a numeric value to change the position in inches or centimeters. Data type: Float
Color	
Background	Specifies the background color of the object. Choose a color from the drop-down list, enter a hexadecimal RGB value (for example, 0xff0000), choose/create a formula or edit an expression that returns a color, or select Custom from the drop-down list to specify the color. Data type: String
CSS	
Class	Specifies a CSS class to be applied to the object which is a valid class in the CSS file. Data type: String
Style	The property can be used in two ways. • Specifies a style to be applied to the object. Choose a style from the drop-down list which is available when a Style Group has been selected at the report tab level and when there are styles in the Style Group that can be applied to the object, or type in the style name. • Specifies a CSS selector to be applied to the object. Enter a valid class name from the CSS file. Data type: String
Others	
Cache	Specifies whether to cache the dataset for this object in the data buffer so other objects which use the same dataset can share the data rather than perform their own SQL query. Data type: Boolean
Embedded	If false, the page header and footer of the subreport will not be shown. Data type: Boolean
Export to Applet	Specifies whether to include the object when exporting the report to Applet. Data type: Boolean
Export to CSV	Specifies whether to include the object when exporting the report to CSV. Data type: Boolean
Export to Excel	Specifies whether to include the object when exporting the report to Excel. Data type: Boolean
Export to HTML	Specifies whether to include the object when exporting the report to HTML. Data type: Boolean
Export to PDF	Specifies whether to include the object when exporting the report to PDF. Data type: Boolean
Export to PostScript	Specifies whether to include the object when exporting the report to PostScript. Data type: Boolean
Export to Report Result	Specifies whether to include the object when exporting the report to Report Result. Data type: Boolean
Export to RTF	Specifies whether to include the object when exporting the report to RTF. Data type: Boolean
Export to Text	Specifies whether to include the object when exporting the report to Text. Data type: Boolean
Export to XML	Specifies whether to include the object when exporting the report to XML. Data type: Boolean
Invisible	Specifies whether to show the object in the design area and in the report results. All formulas and calculations will still be performed if the property is set to true. Data type: Boolean
Maximum Page Number	Specifies the maximum number of pages in the data buffer. Data type: Integer
Position	Specifies the position of the object. Choose an option from the drop-down list. • **static** - The object is positioned at the location in which it is inserted. If selected, the X, Y and other position-related properties will be hidden or disabled. The position of the object is affected by preceding objects in the same container. • **absolute** - The object will be located at the position specified by dragging and dropping or by setting its X and Y property values. The position of the object is not affected by preceding objects in the same container. Data type: Enumeration
Record Location	Specifies the calculation point for the properties which use formulas. Choose an option from the drop-down list. • **Default** - The values of the properties are calculated in the default location where they are located. • **Page Header** - The values of the properties are calculated in the page header. • **Page Footer** - The values of the properties are calculated in the page footer. Data type: Enumeration
Records per Page	Specifies the number of records in each page in the data buffer. Data type: Integer
Subreport Data Source Name	Displays the name of the data source used by the subreport. If nothing is displayed, the default data source will be used. This property is read only.
Subreport Query Name	Displays the name of the query used by the subreport. This property is read only.
Subreport Security Name	Displays the security name (record level security) used by the subreport. This property is read only.
Suppress	Specifies whether to show the object in the design area and in the report results. All formulas and calculations will be skipped if the property is set to true. Data type: Boolean **Note:** When both the Invisible and Suppress properties of an object are set to true, Suppress has the higher priority.
Suppress When Empty	Specifies whether to display the object in the report results when no record is returned to it. Data type: Boolean
Suppress When No Records	Specifies whether to display the object in the report results when no record is returned to its parent data container. Data type: Boolean
Excel	
Column Index	Specifies the X coordinate of the object relative to its parent container when exported to Excel or CSV, measured in cells. The Columned property at the report tab level must be set to true for this property to take effect. Data type: Integer
On New Sheet	Specifies whether to put the subreport in a new sheet when exporting the report to Excel. Data type: Boolean
Row Index	Specifies the Y coordinate of the object relative to its parent container when exported to Excel or CSV, measured in cells. The Columned property at the report tab level must be set to true for this property to take effect. Data type: Integer
Sheet Name	Specifies the name of the new sheet in which the subreport will be put. Make sure the name specified here isn't used by any other sheets. It is meaningful to specify the property value when the On New Sheet property is set to true. Data type: String **Note:** The characters "[", "]", "/", "?", "\", "*", "]", "[" cannot be displayed in the sheet name in Excel and the last character in the sheet name cannot be single quotation mark ('). If any one of them is used when you specify the property, it will be replaced by "_" in Excel.
Sheet Name Postfix	Specifies how to add postfix number to the sheets in which the subreport will be put when exporting the report to Excel. Choose an option from the drop-down list. • **Add postfix number to every page** - Specifies to add the postfix number to every sheet for the subreport (Subreport1, Subreport2, Subreport3, Subreport4, ...). • **Add postfix number only to additional pages** - Specifies to add the postfix number to the subreport sheets starting from the second one (Subreport, Subreport1, Subreport2, Subreport3, ...). It is meaningful to specify the property value when the On New Sheet property is set to true. Data type: Enumeration
Border	
Border Color	Specifies the color of the border of the object. Choose a color from the drop-down list, enter a hexadecimal RGB value (for example, 0xCCAA00), choose/create a formula or edit an expression that returns a color, or select Custom from the drop-down list to specify the color. Data type: String
Border Thickness	Specifies the width of the border in inches or centimeters. Enter a numeric value to change the thickness. Data type: Float
Bottom Line	Specifies the line style of the bottom border of the object. Choose a style from the drop-down list. Data type: Enumeration
Left Line	Specifies the line style of the left border of the object. Choose a style from the drop-down list. Data type: Enumeration
Right Line	Specifies the line style of the right border of the object. Choose a style from the drop-down list. Data type: Enumeration
Top Line	Specifies the line style of the top border of the object. Choose a style from the drop-down list. Data type: Enumeration
TOC	
Anchor Display Value	Specifies a string or formula to display for the TOC entry for this object in the TOC Browser. By default it is the value of the specified object. The TOC Anchor property must be set to true for this property to take effect. Data type: String
TOC Anchor	Specifies whether to include the object in the TOC Browser for the report. Data type: Boolean
Accessibility	
External CSS Class Selector	Specifies a class selector to be applied to the object when exported as HTML. Enter a valid class name from the CSS file. Data type: String
External ID	It is mapped to the HTML attribute *id*. This attribute specifies a name to the object, which must be unique in the report. Data type: String
External Style	It is mapped to the HTML attribute *style*. This attribute specifies style information for the object. Data type: String
Language	It is mapped to the HTML attribute *lang*. This attribute specifies the base language of the object's attribute values and text content. Data type: String

Cache

In a report, when the subreport needs to be run more than once, set the property Cache to true. The data of the subreport will be cached in the data buffer. When the subreport is run the next time, the sub-engine will retrieve the data from the data buffer instead of DBMS, and if sublink exists, it will also be calculated in the data buffer, thus improving the engine performance.

When a data buffer is defined in the sub-engine, the data buffer is divided into many pages and records are allocated in the pages. You can use the properties Records per Page and Maximum Page Number to set the number of the records in one page and the maximum number of pages in the data buffer. If enough physical memory is available and you know the total number of records in the subreport, to achieve better performance, you can set Records per Page approximate to the total number of records.

Note: The Cache property will not work in the following cases:

- When the parameter is displayed in the Parameters in subreport tab which is contained in the query of the subreport.
- When HDS is used by the subreport.
- When dynamic query is used by the subreport.

Summary field

The properties of a summary field in a page report are:

Property Name	Description
General	
Aggregate Function	Shows the function of the summary. This property is read only.
Field Type	Shows what kind of field it is. This property is read only.
Formula	Shows the group field that the summary is based on. If null, the summary is grouped based on the whole dataset. This property is read only.
Instance Name	Shows the instance name of the object. This property is read only.
Name	Shows the name of the summary. This property is read only.
Summary On	Shows the name of the field on which to perform the summary function. This property is read only.
Geometry	
Height	Specifies the height of the object, in inches or centimeters. Enter a numeric value to change the height. Data type: Float
Width	Specifies the width of the object, in inches or centimeters. Enter a numeric value to change the width. Data type: Float
X	Specifies the horizontal coordinate of the top left corner of the object, relative to its parent container. This property is ignored if the Position property is set to static. Enter a numeric value to change the position in inches or centimeters. Data type: Float
Y	Specifies the vertical coordinate of the top left corner of the object, relative to its parent container. This property is ignored if the Position property is set to static. Enter a numeric value to change the position in inches or centimeters. Data type: Float
Color	
Background	Specifies the background color of the object. Choose a color from the drop-down list, enter a hexadecimal RGB value (for example, 0xff0000), choose/create a formula or edit an expression that returns a color, or select Custom from the drop-down list to specify the color. Data type: String
Foreground	Specifies the foreground color of the object. Choose a color from the drop-down list, enter a hexadecimal RGB value (for example, 0xff0000), choose/create a formula or edit an expression that returns a color, or select Custom from the drop-down list to specify the color. Data type: String
CSS	
Class	Specifies a CSS class to be applied to the object which is a valid class in the CSS file. Data type: String
ID	Specifies the identifier of the object, which must be unique in the report. The ID property can be a style sheet selector. Data type: String
Style	The property can be used in two ways. • Specifies a style to be applied to the object. Choose a style from the drop-down list which is available in the Report a Style Group has been selected at the report tab level and when there are styles in the Style Group that can be applied to the object, or type in the style name. • Specifies a CSS selector to be applied to the object. Enter a valid class name from the CSS file. Data type: String
Excel	
Column Index	Specifies the X coordinate of the object relative to its parent container when exported to Excel or CSV, measured in cells. The Columned property at the report lab level must be set to true for this property to take effect. Data type: Integer
Row Index	Specifies the Y coordinate of the object relative to its parent container when exported to Excel or CSV, measured in cells. The Columned property at the report lab level must be set to true for this property to take effect. Data type: Integer
Padding	
Bottom Padding	Specifies the space in inches or centimeters between the text and the bottom border of the object. Enter a numeric value to change the padding. Data type: Float
Left Padding	Specifies the space in inches or centimeters between the text and the left border of the object. Enter a numeric value to change the padding. Data type: Float
Right Padding	Specifies the space in inches or centimeters between the text and the right border of the object. Enter a numeric value to change the padding. Data type: Float
Top Padding	Specifies the space in inches or centimeters between the text and the top border of the object. Enter a numeric value to change the padding. Data type: Float
Border	
Border Color	Specifies the color of the border of the object. Choose a color from the drop-down list, enter a hexadecimal RGB value (for example, 0xCCAA00), choose/create a formula or edit an expression that returns a color, or select Custom from the drop-down list to specify the color. Data type: String
Border Thickness	Specifies the width of the border in inches or centimeters. Enter a numeric value to change the thickness. Data type: Float
Bottom Line	Specifies the line style of the bottom border of the object. Choose a style from the drop-down list. Data type: Enumeration
Left Line	Specifies the line style of the left border of the object. Choose a style from the drop-down list. Data type: Enumeration
Right Line	Specifies the line style of the right border of the object. Choose a style from the drop-down list. Data type: Enumeration
Shadow	Specifies whether to draw two shadowed borders, beneath and to the right of the object. Data type: Boolean
Shadow Color	Specifies the color of the shadow. Choose a color from the drop-down list, enter a hexadecimal RGB value (for example, 0xff0000), choose/create a formula or edit an expression that returns a color, or select Custom from the drop-down list to specify the color and fill effect. Data type: String
Top Line	Specifies the line style of the top border of the object. Choose a style from the drop-down list. Data type: Enumeration
Pattern	
Pattern Color	Specifies the color in which to draw a pattern to fill the object. Choose a color from the drop-down list, enter a hexadecimal RGB value (for example, 0xff0000), choose/create a formula or edit an expression that returns a color, or select Custom from the drop-down list to specify the color and fill effect. Data type: String
Pattern Style	Specifies the style of the pattern. Choose an option from the drop-down list. • **none** - No pattern will be applied to the object. • **50%** - The object will be filled in the Pattern Color of 50%-transparency. • **horizontal** - The pattern will be made of horizontal lines of the Pattern Color. • **vertical** - The pattern will be made of vertical lines of the Pattern Color. • **grid** - The pattern will be an overlapping of horizontal and vertical lines of the Pattern Color. • **diagonal** - The pattern will be made of diagonal lines of the Pattern Color. Data type: Enumeration
Text Format	
Auto Fit	Specifies whether to adjust the width and height of the object according to the contents. Data type: Boolean
Bold	Specifies whether to make the text bold. Data type: Boolean
Font Face	Specifies the font of the text. Choose an option from the drop-down list. Data type: Enumeration
Font Size	Specifies the font size of the text. Enter an integer value to change the size. Data type: Integer
Format	Specifies the display format of the text in the report result. It varies with data type and can be manually defined. Data type: String **Note:** For the BigDecimal type, to avoid precision loss, you should enter a prefix JRD when setting the format property value.
Horizontal Alignment	Specifies the horizontal justification of the text in the object. Choose an option from the drop-down list. Data type: Enumeration
Ignore HTML Tag	Specifies whether or not to ignore the HTML tags when exporting the report results to HTML format. • **true** - The text will appear in the HTML file the same as that in JReport Designer (HTML tag elements in the text, if any, will not be parsed). • **false** - HTML tag elements in the text will be parsed when exporting the report results to HTML format. This property is useful for exporting the report results to HTML format. You may want to write the label contents in HTML tags in your report, and for the exported HTML files, the tags will be transferred as they are translated into HTML. Data type: Boolean
Italic	Specifies whether to make the text italic. Data type: Boolean
Maximum Width	Specifies the maximum width of the text displayed. Enter a numeric value to change the width. This property often works with the Auto Fit property. If Auto Fit = true and Maximum Width is not equal to 0, the text will extend until the width is this value. Data type: Float
Strikethrough	Specifies whether to add a strikethrough line to the text. Data type: Boolean
Underline	Specifies whether to underline the text. Data type: Boolean
Vertical Alignment	Specifies the vertical justification of the text in the object. Choose an option from the drop-down list. Data type: Enumeration
Word Wrap	Specifies whether to wrap the text to the width. Data type: Boolean
Others	
Cache Value	Specifies whether to cache the value of the field instead of obtaining it repeatedly. Data type: Boolean
Column Name	Substitutes the current field with another field by selecting from the drop-down list. To make the property editable, uncheck Forbid changing column in the Options dialog (**File > Options > Panel > Forbid changing column**). Data type: String
Data Mapping File	Specifies the data mapping file to the object for NLS use. For example, if the data mapping file name is Category_de_DE.properties, set the value as Category (without the language and locale part). Data type: String
Detail Report	Specifies the detail report that the object is linked to. Click ... in the value cell to set the detail report. For details, refer to Linking to a detail report.
Detail Target Frame	Specifies the target window or frame in which to display the detailed information. This property is enabled only when Go To Detail is set to true. Choose an option from the drop-down list: • **<Server Setting>** Loads the detailed information according to setting of the Pop Up New Window for Links option in the Customize Profile panel of the Profile dialog in JReport Server. • **New Window** Loads the detailed information into a new window. The window is not named. • **Whole Window** Loads the detailed information into the full browser window. • **Same Frame** Loads the detailed information into the same frame as the object. • **Parent Frame** Loads the detailed information into the parent frame of the frame in which the object is. • **Other Frame** Loads the detailed information into some other specified frame. If selected, you can also directly input the name of the frame you have defined in the text box. If the frame name does not exist, the detailed information will be loaded into a new window. Data type: String
Display Null	Specifies the string to be displayed if the field value is null. Data type: String
Enable Hyperlink in Excel	Specifies whether to enable the link defined on the object when exporting the report to an Excel file. Data type: Boolean
Enable Hyperlink in HTML	Specifies whether to enable the link defined on the object when exporting the report to an HTML file. Data type: Boolean
Enable Hyperlink in PDF	Specifies whether to enable the link defined on the object when exporting the report to a PDF file. Data type: Boolean
Export to CSV	Specifies whether to include the object when exporting the report to CSV. Data type: Boolean
Export to Page Report Result	Specifies whether to include the object when exporting the report to Page Report Result. Data type: Boolean
Export to Excel	Specifies whether to include the object when exporting the report to Excel. Data type: Boolean
Export to PDF	Specifies whether to include the object when exporting the report to PDF. This property is available only when the object is displayed as a web control. Data type: Boolean
Filter Options	Specifies the filter-related options that will be displayed on the shortcut menu when right-clicking the object in Page Report Studio. If Default is selected, the filter options displayed are decided by the setting in the Profile dialog in JReport Server (Profile > Customize Profile > Page Report Studio > Properties > Default > Filter Menu). For more information, refer to Setting filter options for a field.
Go to Detail	Specifies whether to show the detailed information when you click the object in Page Report Studio. This property is available only when the object is in the group header/ footer panel of a banded object. For details about its usage, refer to Obtaining detailed information from a banded object. Data type: Boolean
Invisible	Specifies whether to show the object in the design area and in the report results. All formulas and calculations will still be performed if the property is set to true. Data type: Boolean
Link	Specifies to link the object to another report, a website, an e-mail address or a Blob data type field. Click ... in the value cell to set the link target. For details, see Binding a link to a label. Data type: String
Logic Column	Specifies whether to show the object in the next visible table cell in the same row when the column which holds the object is hidden. Choose an option from the drop-down list. Data type: Enumeration **Notes:** • The next visible table cell should be completely empty, that is, the cell should not hold any content including blank space. • When the property for several objects in the same row is set to next visible column, and the columns holding these objects are all hidden, then only the object in the rightmost column will be shown in the next visible cell.
Position	Specifies the position of the object. Choose an option from the drop-down list. • **static** - The object is positioned at the location in which it is inserted. If selected, the X, Y and other related properties will be hidden or disabled. The position of the object is affected by preceding objects in the same container. • **relative** - The object is positioned as an offset to the position at which it is inserted. The offset is determined by the X and Y property values. • **absolute** - The object is positioned at the position specified by dragging and dropping or by setting its X and Y property values. The position of the object is not affected by preceding objects in the same container. Data type: Enumeration
Record Location	Specifies the calculation point for the properties which use formulas. Choose an option from the drop-down list. • **Default** - The values of the properties are calculated in the default location where they are located. • **Page Header** - The values of the properties are calculated in the page header. • **Page Footer** - The values of the properties are calculated in the page footer. Data type: Enumeration
Suppress	Specifies whether to show the object in the design area and in the report results. All formulas and calculations will be skipped if the property is set to true. Data type: Boolean **Note:** When both the Invisible and Suppress properties of an object are set to true, Suppress has the higher priority.
Suppress When No Records	Specifies whether to display the object in the report results when no record is returned to its parent data container. Data type: Boolean
Suppress When Null	If true and its value is null, the field will not be displayed. Data type: Boolean
Transfer Style	Specifies whether the style group of the primary report will be applied to the linked report if the object is linked to a report. Data type: Boolean
Value Delimiter	Specifies the separator for the data whose type is DBArray. The default value space means that the elements will be displayed in a horizontal line and separated by a space. Data type: String
TOC	
Anchor Display Value	Specifies a string or formula to display for the TOC entry for this object in the TOC Browser. By default it is the value of the specified object. The TOC Anchor property must be set to true for this property to take effect. Data type: String
TOC Anchor	Specifies whether to include the object in the TOC Browser for the report. Data type: Boolean
Accessibility	
External AccessKey	It is mapped to the HTML attribute accesskey. This attribute specifies an access key to the object. Data type: String
External CSS Class Selector	Specifies a class selector to be applied to the object when exported as HTML. Enter a valid class name from the CSS file. Data type: String
External Dir	It is mapped to the HTML attribute dir. This attribute specifies the base direction of directionally neutral text in the object's content and attribute values. It also specifies the directionality of tables. Possible values: • LTR - Left-to-right text or table. • RTL - Right-to-left text or table. Data type: String
External ID	It is mapped to the HTML attribute id. This attribute specifies a name to the object, which must be unique in the report. Data type: String
External Style	It is mapped to the HTML attribute style. This attribute specifies style information for the object. Data type: String
External Tabindex	It is mapped to the HTML attribute tabindex. This attribute specifies the position of the object in the tabbing order for the current report. Enter an integer equal to or greater than 0 and less than 65535. Data type: Integer
External Title	It is mapped to the HTML attribute title. This attribute offers advisory information about the object. Data type: Integer
HrefLang	It is mapped to the HTML attribute hreflang. This attribute specifies the base language of the resource designated by a link and may only be used when a link is specified. Data type: String
Language	It is mapped to the HTML attribute lang. This attribute specifies the base language of the object's attribute values and text content. Data type: String
LongDesc	It is mapped to the HTML attribute longdesc. This attribute specifies a link to a long description of the object. Data type: String

Table

The properties of a table in a page report are:

Property Name	Description
General	
Class Type	Indicates the class type of the object. This property is read only.
Data Inherit	Indicates whether the dataset for this object is inherited from another object. This property is read only.
Dataset	Indicates the dataset used by the object. This property is read only.
Instance Name	Shows the instance name of the object. This property is read only.
Geometry	
Height	Specifies the height of the object, in inches or centimeters. Enter a numeric value to change the height. Data type: Float
Width	Specifies the width of the object, in inches or centimeters. Enter a numeric value to change the width. Data type: Float
X	Specifies the horizontal coordinate of the top left corner of the object, relative to its parent container. This property is ignored if the Position property is set to static. Enter a numeric value to change the position in inches or centimeters. Data type: Float
Y	Specifies the vertical coordinate of the top left corner of the object, relative to its parent container. This property is ignored if the Position property is set to static. Enter a numeric value to change the position in inches or centimeters. Data type: Float
Color	
Background	Specifies the background color of the object. Choose a color from the drop-down list, enter a hexadecimal RGB value (for example, 0xff0000), choose/create a formula or edit an expression that returns a color, or select Custom from the drop-down list to specify the color. Data type: String
CSS	
Class	Specifies a CSS class to be applied to the object which is a valid class in the CSS file. Data type: String
Style	The property can be used in two ways. • Specifies a style to be applied to the object. Choose a style from the drop-down list which is available when a Style Group has been selected at the report tab level and when there are styles in the Style Group that can be applied to the object, or type in the style name. • Specifies a CSS selector to be applied to the object. Enter a valid class name from the CSS file. Data type: String
Others	
Auto Expands	Specifies whether the objects in cells of the table will expand to their empty neighbouring cells horizontally when the actual size of the objects is bigger than the cells due to no enough column or page width for auto fit. This property requires that the property Auto Fit for the objects in the table cells is set to true. Data type: Boolean **Note:** The property does not apply to horizontal tables.
Cache	Specifies whether to cache the dataset for this object in the data buffer so other objects which use the same dataset can share the data rather than perform their own SQL query. Data type: Boolean
Default for Filter	Specifies whether to show the data component as the default component in the Apply to drop-down list of the Filter dialog in Page Report Studio. Data type: Boolean **Note:** In a page report tab, only one data component's Default for Filter property can be set to true.
Export to Applet	Specifies whether to include the object when exporting the report to Applet. Data type: Boolean
Export to CSV	Specifies whether to include the object when exporting the report to CSV. Data type: Boolean
Export to Excel	Specifies whether to include the object when exporting the report to Excel. Data type: Boolean
Export to HTML	Specifies whether to include the object when exporting the report to HTML. Data type: Boolean
Export to PDF	Specifies whether to include the object when exporting the report to PDF. Data type: Boolean
Export to PostScript	Specifies whether to include the object when exporting the report to PostScript. Data type: Boolean
Export to Report Result	Specifies whether to include the object when exporting the report to Report Result. Data type: Boolean
Export to RTF	Specifies whether to include the object when exporting the report to RTF. Data type: Boolean
Export to Text	Specifies whether to include the object when exporting the report to Text. Data type: Boolean
Export to XML	Specifies whether to include the object when exporting the report to XML. Data type: Boolean
Horizontal	Specifies whether to show the table in horizontal style. Data type: Boolean
Invisible	Specifies whether to show the object in the design area and in the report results. All formulas and calculations will still be performed if the property is set to true. Data type: Boolean
Invisible for Filter Dialogs	Specifies whether to show the data component in the Apply to drop-down list of the Filter dialog in Page Report Studio. This property cannot be edited when Default for Filter is set to true. Data type: Boolean
Position	Specifies the position of the object. Choose an option from the drop-down list. • **static** - The object is positioned at the location in which it is inserted. If selected, the X, Y and other position-related properties will be hidden or disabled. The position of the object is affected by preceding objects in the same container. • **absolute** - The object will be located at the position specified by dragging and dropping or by setting its X and Y property values. The position of the object is not affected by preceding objects in the same container. Data type: Enumeration
Record Location	Specifies the calculation point for the properties which use formulas. Choose an option from the drop-down list. • **Default** - The values of the properties are calculated in the default location where they are located. • **Page Header** - The values of the properties are calculated in the page header. • **Page Footer** - The values of the properties are calculated in the page footer. Data type: Enumeration
Suppress	Specifies whether to show the object in the design area and in the report results. All formulas and calculations will be skipped if the property is set to true. Data type: Boolean **Note:** When both the Invisible and Suppress properties of an object are set to true, Suppress has the higher priority.
Suppress When Empty	Specifies whether to display the object in the report results when no record is returned to it. Data type: Boolean
Suppress When No Records	Specifies whether to display the object in the report results when no record is returned to its parent data container. Data type: Boolean
Border	
Border Color	Specifies the color of the border of the object. Choose a color from the drop-down list, enter a hexadecimal RGB value (for example, 0xCCAA00), choose/create a formula or edit an expression that returns a color, or select Custom from the drop-down list to specify the color. Data type: String
Border Thickness	Specifies the width of the border in inches or centimeters. Enter a numeric value to change the thickness. Data type: Float
Bottom Line	Specifies the line style of the bottom border of the object. Choose a style from the drop-down list. Data type: Enumeration
Left Line	Specifies the line style of the left border of the object. Choose a style from the drop-down list. Data type: Enumeration
Right Line	Specifies the line style of the right border of the object. Choose a style from the drop-down list. Data type: Enumeration
Top Line	Specifies the line style of the top border of the object. Choose a style from the drop-down list. Data type: Enumeration
TOC	
Anchor Display Value	Specifies a string or formula to display for the TOC entry for this object in the TOC Browser. By default it is the value of the specified object. The TOC Anchor property must be set to true for this property to take effect. Data type: String
TOC Anchor	Specifies whether to include the object in the TOC Browser for the report. Data type: Boolean
Accessibility	
External CSS Class Selector	Specifies a class selector to be applied to the object when exported as HTML. Enter a valid class name from the CSS file. Data type: String
External ID	It is mapped to the HTML attribute *id*. This attribute specifies a name to the object, which must be unique in the report. Data type: String
External Style	It is mapped to the HTML attribute *style*. This attribute specifies style information for the object. Data type: String
Summary	It is mapped to the HTML attribute *summary*. This attribute provides a summary of the object's purpose and structure. Data type: String

The following topics list properties of the children of a table:

- Table header row
- Table group
- Table footer row
- Table column
- Table cell

Table header row

The properties of a table header row in a page report are:

Property Name	Description
General	
Class Type	Indicates the class type of the object. This property is read only.
Data Inherit	Indicates whether the dataset for this object is inherited from another object. This property is read only.
Dataset	Indicates the dataset used by the object. This property is read only.
Instance Name	Shows the instance name of the object. This property is read only.
Geometry	
Height	Specifies the height of the object, in inches or centimeters. Enter a numeric value to change the height. Data type: Float
Width	Specifies the width of the object, in inches or centimeters. Enter a numeric value to change the width. Data type: Float
Color	
Background	Specifies the background color and fill effect of the object. Choose a color from the drop-down list, enter a hexadecimal RGB value (for example, 0xff0000), choose/create a formula or edit an expression that returns a color, or select Custom from the drop-down list to specify the color and fill effect. Data type: String
CSS	
Class	Specifies a CSS class to be applied to the object which is a valid class in the CSS file. Data type: String
Style	The property can be used in two ways. • Specifies a style to be applied to the object. Choose a style from the drop-down list which is available when a Style Group has been selected at the report tab level and when there are styles in the Style Group that can be applied to the object, or type in the style name. • Specifies a CSS selector to be applied to the object. Enter a valid class name from the CSS file. Data type: String
Others	
Export to CSV	Specifies whether to include the object when exporting the report to CSV. Data type: Boolean
Export to Page Report Result	Specifies whether to include the object when exporting the report to Page Report Result. Data type: Boolean
Export to Excel	Specifies whether to include the object when exporting the report to Excel. Data type: Boolean
Invisible	Specifies whether to show the object in the design area and in the report results. All formulas and calculations will still be performed if the property is set to true. Data type: Boolean
Label	Specifies the popup text displayed when the mouse pointer hovers over the left edge of the row. Data type: String
On New Page	Specifies whether the row starts on a new page. This property is applied only when the report has page layout enabled. The default is false which means the row starts on a new page only if the previous page is filled. Data type: Boolean
Record Location	Specifies the calculation point for the properties which use formulas. Choose an option from the drop-down list. • **Default** - The values of the properties are calculated in the default location where they are located. • **Page Header** - The values of the properties are calculated in the page header. • **Page Footer** - The values of the properties are calculated in the page footer. Data type: Enumeration
Repeat Header for Excel	Specifies whether to make the object appear in every page of the report results when exporting the report to Excel. Data type: Boolean
Repeat Header for TXT	Specifies whether to make the object appear in every page of the report results when exporting the report to a .txt file. Data type: Boolean
Repeat Header for XML	Specifies whether to make the object appear in every page of the report results when exporting the report to XML. Data type: Boolean
Repeat on Pages	Specifies whether to make the object appear in every page of the report results. Data type: Boolean
Suppress	Specifies whether to show the object in the design area and in the report results. All formulas and calculations will be skipped if the property is set to true. Data type: Boolean **Note:** When both the Invisible and Suppress properties of an object are set to true, Suppress has the higher priority.
Suppress When No Records	Specifies whether to display the object in the report results when no record is returned to its parent data container. Data type: Boolean
Border	
Border Color	Specifies the color of the border of the object. Choose a color from the drop-down list, enter a hexadecimal RGB value (for example, 0xCCAA00), choose/create a formula or edit an expression that returns a color, or select Custom from the drop-down list to specify the color. Data type: String
Border Thickness	Specifies the width of the border in inches or centimeters. Enter a numeric value to change the thickness. Data type: Float
Bottom Line	Specifies the line style of the bottom border of the object. Choose a style from the drop-down list. Data type: Enumeration
Left Line	Specifies the line style of the left border of the object. Choose a style from the drop-down list. Data type: Enumeration
Right Line	Specifies the line style of the right border of the object. Choose a style from the drop-down list. Data type: Enumeration
Top Line	Specifies the line style of the top border of the object. Choose a style from the drop-down list. Data type: Enumeration
Accessibility	
External CSS Class Selector	Specifies a class selector to be applied to the object when exported as HTML. Enter a valid class name from the CSS file. Data type: String
External ID	It is mapped to the HTML attribute *id*. This attribute specifies a name to the object, which must be unique in the report. Data type: String
Language	It is mapped to the HTML attribute *lang*. This attribute specifies the base language of the object's attribute values and text content. Data type: String

Table group

The properties of a table group in a page report are:

Property Name	Description
General	
Class Type	Indicates the class type of the object. This property is read only.
Group By	Shows the group field that the object is based on. If null, the object is grouped based on the whole dataset. This property is read only.
Instance Name	Shows the instance name of the object. This property is read only.
Others	
Current Block Index	The two properties, Current Block Index and Items per Block, work together to control the data of the object to be displayed in continuous page mode. Current Block Index specifies the index of the data block that will be displayed. 0 means the first block index, and 1 the second, and so on. Data type: Integer
Expand Detail Data	Specifies whether to expand the groups to show the details when the report is opened in Page Report Studio. Only works in continuous page mode. Data type: Boolean
Items per Block	The two properties, Current Block Index and Items per Block, work together to control the data of the object to be displayed in continuous page mode. Item per Block specifies the number of records in each data block. Data type: Integer
Group Layout	
Keep Group Together	Specifies whether to keep the whole group together. Data type: Boolean
TOC	
Anchor Display Value	Specifies a string or formula to display for the TOC entry for this object in the TOC Browser. By default it is the value of the specified object. The TOC Anchor property must be set to true for this property to take effect. Data type: String
TOC Anchor	Specifies whether to include the object in the TOC Browser for the report. A Boolean type value.
Security	
Cascade	Specifies whether to allow the security identifiers specified by the following three properties to view the group's detail or child groups. • **true** - The detail row, the group header row, and the group footer row will be displayed. • **false** - Only the group header and footer rows will be displayed. Data type: Boolean
Grant	Enables to set cached report bursting. The property accepts user names delimited by "|" as the value. You can specify which group of data can be viewed by which user. Data type: String
Groups	Enables to set cached report bursting. The property accepts group names delimited by "|" as the value. You can specify which group of data can be viewed by which group of users. Data type: String
Roles	Enables to set cached report bursting. The property accepts role names delimited by "|" as the value. You can specify which group of data can be viewed by which role. Data type: String

A table group may contain the following objects:

- Table group header row
- Table detail row
- Table group footer row

Table group header row

The properties of a table group header row in a page report are:

Property Name	Description
General	
Class Type	Indicates the class type of the object. This property is read only.
Data Inherit	Indicates whether the dataset for this object is inherited from another object. This property is read only.
Dataset	Indicates the dataset used by the object. This property is read only.
Instance Name	Shows the instance name of the object. This property is read only.
Geometry	
Height	Specifies the height of the object, in inches or centimeters. Enter a numeric value to change the height. Data type: Float
Width	Displays the width of the object, in inches or centimeters. This property is read only.
Color	
Background	Specifies the background color and fill effect of the object. Choose a color from the drop-down list, enter a hexadecimal RGB value (for example, 0xff0000), choose/create a formula or edit an expression that returns a color, or select Custom from the drop-down list to specify the color and fill effect. Data type: String
CSS	
Class	Specifies a CSS class to be applied to the object which is a valid class in the CSS file. Data type: String
Style	The property can be used in two ways. • Specifies a style to be applied to the object. Choose a style from the drop-down list which is available when a Style Group has been selected at the report tab level and when there are styles in the Style Group that can be applied to the object, or type in the style name. • Specifies a CSS selector to be applied to the object. Enter a valid class name from the CSS file. Data type: String
Others	
Export to CSV	Specifies whether to include the object when exporting the report to CSV. Data type: Boolean
Export to Page Report Result	Specifies whether to include the object when exporting the report to Page Report Result. Data type: Boolean
Export to Excel	Specifies whether to include the object when exporting the report to Excel. Data type: Boolean
Fill Whole Page	Specifies whether to make the row extended to the bottom of the page, so that the next row starts on a new page. Data type: Boolean
Invisible	Specifies whether to show the object in the design area and in the report results. All formulas and calculations will still be performed if the property is set to true. Data type: Boolean
Label	Specifies the popup text displayed when the mouse pointer hovers over the left edge of the row. Data type: String
On New Page	Specifies whether the row starts on a new page. This property is applied only when the report has page layout enabled. The default is false which means the row starts on a new page only if the previous page is filled. Data type: Boolean
Record Location	Specifies the calculation point for the properties which use formulas. Choose an option from the drop-down list. • **Default** - The values of the properties are calculated in the default location where they are located. • **Page Header** - The values of the properties are calculated in the page header. • **Page Footer** - The values of the properties are calculated in the page footer. Data type: Enumeration
Repeat	Specifies whether to make the group header appear in every page. Data type: Boolean
Suppress	Specifies whether to show the object in the design area and in the report results. All formulas and calculations will be skipped if the property is set to true. Data type: Boolean **Note:** When both the Invisible and Suppress properties of an object are set to true, Suppress has the higher priority.
Suppress When No Records	Specifies whether to display the object in the report results when no record is returned to its parent data container. Data type: Boolean
Border	
Border Color	Specifies the color of the border of the object. Choose a color from the drop-down list, enter a hexadecimal RGB value (for example, 0xCCAA00), choose/create a formula or edit an expression that returns a color, or select Custom from the drop-down list to specify the color. Data type: String
Border Thickness	Specifies the width of the border in inches or centimeters. Enter a numeric value to change the thickness. Data type: Float
Bottom Line	Specifies the line style of the bottom border of the object. Choose a style from the drop-down list. Data type: Enumeration
Left Line	Specifies the line style of the left border of the object. Choose a style from the drop-down list. Data type: Enumeration
Right Line	Specifies the line style of the right border of the object. Choose a style from the drop-down list. Data type: Enumeration
Top Line	Specifies the line style of the top border of the object. Choose a style from the drop-down list. Data type: Enumeration
Accessibility	
External CSS Class Selector	Specifies a class selector to be applied to the object when exported as HTML. Enter a valid class name from the CSS file. Data type: String
Language	It is mapped to the HTML attribute *lang*. This attribute specifies the base language of the object's attribute values and text content. Data type: String

Table detail row

The properties of a table detail row in a page report are:

Property Name	Description
General	
Class Type	Indicates the class type of the object. This property is read only.
Data Inherit	Indicates whether the dataset for this object is inherited from another object. This property is read only.
Dataset	Indicates the dataset used by the object. This property is read only.
Instance Name	Shows the instance name of the object. This property is read only.
Geometry	
Height	Specifies the height of the object, in inches or centimeters. Enter a numeric value to change the height. Data type: Float
Width	Displays the width of the object, in inches or centimeters. This property is read only.
Color	
Background	Specifies the background color and fill effect of the object. Choose a color from the drop-down list, enter a hexadecimal RGB value (for example, 0xff0000), choose/create a formula or edit an expression that returns a color, or select Custom from the drop-down list to specify the color and fill effect. Data type: String
CSS	
Class	Specifies a CSS class to be applied to the object which is a valid class in the CSS file. Data type: String
Style	The property can be used in two ways. • Specifies a style to be applied to the object. Choose a style from the drop-down list which is available when a Style Group has been selected at the report tab level and when there are styles in the Style Group that can be applied to the object, or type in the style name. • Specifies a CSS selector to be applied to the object. Enter a valid class name from the CSS file. Data type: String
Others	
Current Block Index	The two properties, Current Block Index and Items per Block, work together to control the data of the object to be displayed in continuous page mode. Current Block Index specifies the index of the data block that will be displayed. 0 means the first block index, and 1 the second, and so on. Data type: Integer
Export to CSV	Specifies whether to include the object when exporting the report to CSV. Data type: Boolean
Export to Page Report Result	Specifies whether to include the object when exporting the report to Page Report Result. Data type: Boolean
Export to Excel	Specifies whether to include the object when exporting the report to Excel. Data type: Boolean
Fill Whole Page	Specifies whether to make the row extended to the bottom of the page, so that the next row starts on a new page. Data type: Boolean
Invisible	Specifies whether to show the object in the design area and in the report results. All formulas and calculations will still be performed if the property is set to true. Data type: Boolean
Items per Block	The two properties, Current Block Index and Items per Block, work together to control the data of the object to be displayed in continuous page mode. Item per Block specifies the number of records in each data block. Data type: Integer
Label	Specifies the popup text displayed when the mouse pointer hovers over the left edge of the row. Data type: String
On New Page	Specifies whether the row starts on a new page. This property is applied only when the report has page layout enabled. The default is false which means the row starts on a new page only if the previous page is filled. Data type: Boolean
Record Location	Specifies the calculation point for the properties which use formulas. Choose an option from the drop-down list. • **Default** - The values of the properties are calculated in the default location where they are located. • **Page Header** - The values of the properties are calculated in the page header. • **Page Footer** - The values of the properties are calculated in the page footer. Data type: Enumeration
Suppress	Specifies whether to show the object in the design area and in the report results. All formulas and calculations will be skipped if the property is set to true. Data type: Boolean **Note:** When both the Invisible and Suppress properties of an object are set to true, Suppress has the higher priority.
Suppress When No Records	Specifies whether to display the object in the report results when no record is returned to its parent data container. Data type: Boolean
Border	
Border Color	Specifies the color of the border of the object. Choose a color from the drop-down list, enter a hexadecimal RGB value (for example, 0xCCAA00), choose/create a formula or edit an expression that returns a color, or select Custom from the drop-down list to specify the color. Data type: String
Border Thickness	Specifies the width of the border in inches or centimeters. Enter a numeric value to change the thickness. Data type: Float
Bottom Line	Specifies the line style of the bottom border of the object. Choose a style from the drop-down list Data type: Enumeration
Left Line	Specifies the line style of the left border of the object. Choose a style from the drop-down list. Data type: Enumeration
Right Line	Specifies the line style of the right border of the object. Choose a style from the drop-down list. Data type: Enumeration
Top Line	Specifies the line style of the top border of the object. Choose a style from the drop-down list. Data type: Enumeration
Accessibility	
External CSS Class Selector	Specifies a class selector to be applied to the object when exported as HTML. Enter a valid class name from the CSS file. Data type: String

Table group footer row

The properties of a table group footer row in a page report are:

Property Name	Description
General	
Class Type	Indicates the class type of the object. This property is read only.
Data Inherit	Indicates whether the dataset for this object is inherited from another object. This property is read only.
Dataset	Indicates the dataset used by the object. This property is read only.
Instance Name	Shows the instance name of the object. This property is read only.
Geometry	
Height	Specifies the height of the object, in inches or centimeters. Enter a numeric value to change the height. Data type: Float
Width	Displays the width of the object, in inches or centimeters. This property is read only.
Color	
Background	Specifies the background color and fill effect of the object. Choose a color from the drop-down list, enter a hexadecimal RGB value (for example, 0xff0000), choose/create a formula or edit an expression that returns a color, or select Custom from the drop-down list to specify the color and fill effect. Data type: String
CSS	
Class	Specifies a CSS class to be applied to the object which is a valid class in the CSS file. Data type: String
Style	The property can be used in two ways. • Specifies a style to be applied to the object. Choose a style from the drop-down list which is available when a Style Group has been selected at the report tab level and when there are styles in the Style Group that can be applied to the object, or type in the style name. • Specifies a CSS selector to be applied to the object. Enter a valid class name from the CSS file. Data type: String
Others	
Export to CSV	Specifies whether to include the object when exporting the report to CSV. Data type: Boolean
Export to Page Report Result	Specifies whether to include the object when exporting the report to Page Report Result. Data type: Boolean
Export to Excel	Specifies whether to include the object when exporting the report to Excel. Data type: Boolean
Fill Whole Page	Specifies whether to make the row extended to the bottom of the page, so that the next row starts on a new page. Data type: Boolean
Invisible	Specifies whether to show the object in the design area and in the report results. All formulas and calculations will still be performed if the property is set to true. Data type: Boolean
Label	Specifies the popup text displayed when the mouse pointer hovers over the left edge of the row. Data type: String
On New Page	Specifies whether the row starts on a new page. This property is applied only when the report has page layout enabled. The default is false which means the row starts on a new page only if the previous page is filled. Data type: Boolean
Record Location	Specifies the calculation point for the properties which use formulas. Choose an option from the drop-down list. • **Default** - The values of the properties are calculated in the default location where they are located. • **Page Header** - The values of the properties are calculated in the page header. • **Page Footer** - The values of the properties are calculated in the page footer. Data type: Enumeration
Suppress	Specifies whether to show the object in the design area and in the report results. All formulas and calculations will be skipped if the property is set to true. Data type: Boolean **Note:** When both the Invisible and Suppress properties of an object are set to true, Suppress has the higher priority.
Suppress When No Records	Specifies whether to display the object in the report results when no record is returned to its parent data container. Data type: Boolean
Border	
Border Color	Specifies the color of the border of the object. Choose a color from the drop-down list, enter a hexadecimal RGB value (for example, 0xCCAA00), choose/create a formula or edit an expression that returns a color, or select Custom from the drop-down list to specify the color. Data type: String
Border Thickness	Specifies the width of the border in inches or centimeters. Enter a numeric value to change the thickness. Data type: Float
Bottom Line	Specifies the line style of the bottom border of the object. Choose a style from the drop-down list. Data type: Enumeration
Left Line	Specifies the line style of the left border of the object. Choose a style from the drop-down list. Data type: Enumeration
Right Line	Specifies the line style of the right border of the object. Choose a style from the drop-down list. Data type: Enumeration
Top Line	Specifies the line style of the top border of the object. Choose a style from the drop-down list. Data type: Enumeration
Accessibility	
External CSS Class Selector	Specifies a class selector to be applied to the object when exported as HTML. Enter a valid class name from the CSS file. Data type: String
External ID	It is mapped to the HTML attribute *id*. This attribute specifies a name to the object, which must be unique in the report. Data type: String
External Style	It is mapped to the HTML attribute *style*. This attribute specifies style information for the object. Data type: String
Language	It is mapped to the HTML attribute *lang*. This attribute specifies the base language of the object's attribute values and text content. Data type: String

Table footer row

The properties of a table footer row in a page report are:

Property Name	Description
General	
Class Type	Indicates the class type of the object. This property is read only.
Data Inherit	Indicates whether the dataset for this object is inherited from another object. This property is read only.
Dataset	Indicates the dataset used by the object. This property is read only.
Instance Name	Shows the instance name of the object. This property is read only.
Geometry	
Height	Specifies the height of the object, in inches or centimeters. Enter a numeric value to change the height. Data type: Float
Width	Displays the width of the object, in inches or centimeters. This property is read only.
Color	
Background	Specifies the background color and fill effect of the object. Choose a color from the drop-down list, enter a hexadecimal RGB value (for example, 0xff0000), choose/create a formula or edit an expression that returns a color, or select Custom from the drop-down list to specify the color and fill effect. Data type: String
CSS	
Class	Specifies a CSS class to be applied to the object which is a valid class in the CSS file. Data type: String
Style	The property can be used in two ways. • Specifies a style to be applied to the object. Choose a style from the drop-down list which is available when a Style Group has been selected at the report tab level and when there are styles in the Style Group that can be applied to the object, or type in the style name. • Specifies a CSS selector to be applied to the object. Enter a valid class name from the CSS file. Data type: String
Others	
Export to CSV	Specifies whether to include the object when exporting the report to CSV. Data type: Boolean
Export to Page Report Result	Specifies whether to include the object when exporting the report to Page Report Result. Data type: Boolean
Export to Excel	Specifies whether to include the object when exporting the report to Excel. Data type: Boolean
Fill Whole Page	Specifies whether to make the row extended to the bottom of the page, so that the next row starts on a new page. Data type: Boolean
Invisible	Specifies whether to show the object in the design area and in the report results. All formulas and calculations will still be performed if the property is set to true. Data type: Boolean
Label	Specifies the popup text displayed when the mouse pointer hovers over the left edge of the row. Data type: String
On New Page	Specifies whether the row starts on a new page. This property is applied only when the report has page layout enabled. The default is false which means the row starts on a new page only if the previous page is filled. Data type: Boolean
Record Location	Specifies the calculation point for the properties which use formulas. Choose an option from the drop-down list. • **Default** - The values of the properties are calculated in the default location where they are located. • **Page Header** - The values of the properties are calculated in the page header. • **Page Footer** - The values of the properties are calculated in the page footer. Data type: Enumeration
Suppress	Specifies whether to show the object in the design area and in the report results. All formulas and calculations will be skipped if the property is set to true. Data type: Boolean **Note:** When both the Invisible and Suppress properties of an object are set to true, Suppress has the higher priority.
Suppress When No Records	Specifies whether to display the object in the report results when no record is returned to its parent data container. Data type: Boolean
Border	
Border Color	Specifies the color of the border of the object. Choose a color from the drop-down list, enter a hexadecimal RGB value (for example, 0xCCAA00), choose/create a formula or edit an expression that returns a color, or select Custom from the drop-down list to specify the color. Data type: String
Border Thickness	Specifies the width of the border in inches or centimeters. Enter a numeric value to change the thickness. Data type: Float
Bottom Line	Specifies the line style of the bottom border of the object. Choose a style from the drop-down list. Data type: Enumeration
Left Line	Specifies the line style of the left border of the object. Choose a style from the drop-down list. Data type: Enumeration
Right Line	Specifies the line style of the right border of the object. Choose a style from the drop-down list. Data type: Enumeration
Top Line	Specifies the line style of the top border of the object. Choose a style from the drop-down list. Data type: Enumeration
Accessibility	
External CSS Class Selector	Specifies a class selector to be applied to the object when exported as HTML. Enter a valid class name from the CSS file. Data type: String
External ID	It is mapped to the HTML attribute *id*. This attribute specifies a name to the object, which must be unique in the report. Data type: String
External Style	It is mapped to the HTML attribute *style*. This attribute specifies style information for the object. Data type: String
Language	It is mapped to the HTML attribute *lang*. This attribute specifies the base language of the object's attribute values and text content. Data type: String

Table column

There are four types of columns in a table of a page report, which are common column, detail column, group column and summary column. These table column objects have the same properties as follows:

Property Name	Description
General	
Class Type	Indicates the class type of the object. This property is read only.
Instance Name	Shows the instance name of the object. This property is read only.
Others	
Auto Fit	Specifies whether to adjust the width of the object according to the contents. If the property is set to true, the width of the object will be adjusted automatically only when: • The Position property for the field/label in this object is static or relative. • If the Position property for the field/label in this object is absolute, then the Auto Fit property for the field/label should be set to true. Data type: Boolean
Invisible	Specifies whether to show the object in the design area and in the report results. All formulas and calculations will still be performed if the property is set to true. Data type: Boolean

Table cell

The properties of a table cell in a page report are:

Property Name	Description
General	
Class Type	Indicates the class type of the object. This property is read only.
Data Inherit	Indicates whether the dataset for this object is inherited from another object. This property is read only.
Dataset	Indicates the dataset used by the object. This property is read only.
Instance Name	Shows the instance name of the object. This property is read only.
Geometry	
Height	Specifies the height of the object, in inches or centimeters. Enter a numeric value to change the height. Data type: Float
Width	Specifies the width of the object, in inches or centimeters. Enter a numeric value to change the width.
Color	
Background	Specifies the background color of the object. Choose a color from the drop-down list, enter a hexadecimal RGB value (for example, 0xff0000), choose/create a formula or edit an expression that returns a color, or select Custom from the drop-down list to specify the color. Data type: String
CSS	
Class	Specifies a CSS class to be applied to the object which is a valid class in the CSS file. Data type: String
Style	The property can be used in two ways. • Specifies a style to be applied to the object. Choose a style from the drop-down list which is available when a Style Group has been selected at the report tab level and when there are styles in the Style Group that can be applied to the object, or type in the style name. • Specifies a CSS selector to be applied to the object. Enter a valid class name from the CSS file. Data type: String
Others	
Export to CSV	Specifies whether to include the object when exporting the report to CSV. Data type: Boolean
Export to Page Report Result	Specifies whether to include the object when exporting the report to Page Report Result. Data type: Boolean
Export to Excel	Specifies whether to include the object when exporting the report to Excel. Data type: Boolean
Horizontal Alignment	Specifies the horizontal justification of the content in the table cell. Choose an option from the drop-down list. Data type: Enumeration **Note:** When the Position property of the object in the table cell is set to absolute, this property does not take effect.
Joining Merge	Specifies whether to make all the sequential rows included in the cell vertically merged in the report results. It is meaningful to set the property only when the cell is merged with other cells in the same column. Data type: Boolean
Record Location	Specifies the calculation point for the properties which use formulas. Choose an option from the drop-down list. • **Default** - The values of the properties are calculated in the default location where they are located. • **Page Header** - The values of the properties are calculated in the page header. • **Page Footer** - The values of the properties are calculated in the page footer. Data type: Enumeration
Repeat	Specifies whether to repeat the content of the cell in every page of the report results when the cell is split into pages. Available only for vertically merged cell. Data type: Boolean
Suppress When No Records	Specifies whether to display the object in the report results when no record is returned to its parent data container. Data type: Boolean
Vertical Alignment	Specifies the vertical justification of the content in the table cell. Choose an option from the drop-down list. Data type: Enumeration **Note:** When the Position property of the object in the table cell is set to absolute, this property does not take effect.
Border	
Border Color	Specifies the color of the border of the object. Choose a color from the drop-down list, enter a hexadecimal RGB value (for example, 0xCCAA00), choose/create a formula or edit an expression that returns a color, or select Custom from the drop-down list to specify the color. Data type: String
Border Thickness	Specifies the width of the border in inches or centimeters. Enter a numeric value to change the thickness. Data type: Float
Bottom Line	Specifies the line style of the bottom border of the object. Choose a style from the drop-down list. Data type: Enumeration
Left Line	Specifies the line style of the left border of the object. Choose a style from the drop-down list. Data type: Enumeration
Right Line	Specifies the line style of the right border of the object. Choose a style from the drop-down list. Data type: Enumeration
Top Line	Specifies the line style of the top border of the object. Choose a style from the drop-down list. Data type: Enumeration
Accessibility	
Abbr	It is mapped to the HTML attribute *abbr*. This attribute specifies an abbreviated form of the content of the object. The abbreviated text may be rendered by user agents when appropriate in place of the object's content. Data type: String
Axis	It is mapped to the HTML attribute *axis*. This attribute is used to place a cell into conceptual categories that can be considered to form axes in an n-dimensional space. The value of this attribute is a comma-separated list of category names. Data type: String
External CSS Class Selector	Specifies a class selector to be applied to the object when exported as HTML. Enter a valid class name from the CSS file. Data type: String
External ID	It is mapped to the HTML attribute *id*. This attribute specifies a name to the object, which must be unique in the report. Data type: String
Headers	It is mapped to the HTML attribute *headers*. This attribute specifies the list of header cells that provide header information for the current data cell. The value of this attribute is a space-separated list of cell names; those cells must be named by setting their *id* attribute. Data type: String
Scope	It is mapped to the HTML attribute *scope*. This attribute specifies the set of data cells for which the current header cell provides header information. Available values are: • **Row** - The current cell provides header information for the rest of the row that contains it. • **Column** - The current cell provides header information for the rest of the column that contains it. • **none** - The scope attribute will not be generated when exporting to HTML. Data type: String

Tabular

The properties of a tabular in a page report are:

Property Name	Description
General	
Class Type	Indicates the class type of the object. This property is read only.
Instance Name	Shows the instance name of the object. This property is read only.
Geometry	
Height	Specifies the height of the object, in inches or centimeters. Enter a numeric value to change the height. Data type: Float
Width	Specifies the width of the object, in inches or centimeters. Enter a numeric value to change the width. Data type: Float
X	Specifies the horizontal coordinate of the top left corner of the object, relative to its parent container. This property is ignored if the Position property is set to static. Enter a numeric value to change the position in inches or centimeters. Data type: Float
Y	Specifies the vertical coordinate of the top left corner of the object, relative to its parent container. This property is ignored if the Position property is set to static. Enter a numeric value to change the position in inches or centimeters. Data type: Float
Color	
Background	Specifies the background color of the object. Choose a color from the drop-down list, enter a hexadecimal RGB value (for example, 0xff0000), choose/create a formula or edit an expression that returns a color, or select Custom from the drop-down list to specify the color. Data type: String
CSS	
Class	Specifies a CSS class to be applied to the object which is a valid class in the CSS file. Data type: String
Style	The property can be used in two ways. • Specifies a style to be applied to the object. Choose a style from the drop-down list which is available when a Style Group has been selected at the report tab level and when there are styles in the Style Group that can be applied to the object, or type in the style name. • Specifies a CSS selector to be applied to the object. Enter a valid class name from the CSS file. Data type: String
Others	
Export to Applet	Specifies whether to include the object when exporting the report to Applet. Data type: Boolean
Export to CSV	Specifies whether to include the object when exporting the report to CSV. Data type: Boolean
Export to Excel	Specifies whether to include the object when exporting the report to Excel. Data type: Boolean
Export to HTML	Specifies whether to include the object when exporting the report to HTML. Data type: Boolean
Export to PDF	Specifies whether to include the object when exporting the report to PDF. Data type: Boolean
Export to PostScript	Specifies whether to include the object when exporting the report to PostScript. Data type: Boolean
Export to Report Result	Specifies whether to include the object when exporting the report to Report Result. Data type: Boolean
Export to RTF	Specifies whether to include the object when exporting the report to RTF. Data type: Boolean
Export to Text	Specifies whether to include the object when exporting the report to Text. Data type: Boolean
Export to XML	Specifies whether to include the object when exporting the report to XML. Data type: Boolean
Horizontal Auto Size	Specifies whether to make the width of the object automatically expanded according to the size of the components inserted. Data type: Boolean
Invisible	Specifies whether to show the object in the design area and in the report results. All formulas and calculations will still be performed if the property is set to true. Data type: Boolean
Position	Specifies the position of the object. Choose an option from the drop-down list. • **static** - The object is positioned at the location in which it is inserted. If selected, the X, Y and other position-related properties will be hidden or disabled. The position of the object is affected by preceding objects in the same container. • **absolute** - The object will be located at the position specified by dragging and dropping or by setting its X and Y property values. The position of the object is not affected by preceding objects in the same container. Data type: Enumeration
Suppress	Specifies whether to show the object in the design area and in the report results. All formulas and calculations will be skipped if the property is set to true. Data type: Boolean **Note:** When both the Invisible and Suppress properties of an object are set to true, Suppress has the higher priority.
Vertical Auto Size	Specifies whether to make the height of the object automatically expanded according to the size of the components inserted. Data type: Boolean
Excel	
Column Index	Specifies the X coordinate of the object relative to its parent container when exported to Excel or CSV, measured in cells. The Columned property at the report tab level must be set to true for this property to take effect. Data type: Integer
Row Index	Specifies the Y coordinate of the object relative to its parent container when exported to Excel or CSV, measured in cells. The Columned property at the report tab level must be set to true for this property to take effect. Data type: Integer
TOC	
Anchor Display Value	Specifies a string or formula to display for the TOC entry for this object in the TOC Browser. By default it is the value of the specified object. The TOC Anchor property must be set to true for this property to take effect. Data type: String
TOC Anchor	Specifies whether to include the object in the TOC Browser for the report. Data type: Boolean
Accessibility	
External CSS Class Selector	Specifies a class selector to be applied to the object when exported as HTML. Enter a valid class name from the CSS file. Data type: String

A tabular may contain these objects:

- Tabular row group
- Tabular cell

Tabular row group

The properties of a tabular row group in a page report are:

Property Name	Description
General	
Class Type	Indicates the class type of the object. This property is read only.
Instance Name	Shows the instance name of the object. This property is read only.
Others	
Cross Page	Specifies whether the row can be split across a page break. This property is applied only when the report has page layout enabled. Data type: Boolean
Fill Whole Page	Specifies whether to make the row extended to the bottom of the page, so that the next panel starts on a new page. Data type: Boolean
On New Page	Specifies whether the row starts on a new page. This property is only applied if the report has page layout enabled. The default is false which means the panel starts on a new page only if the previous page is filled. Data type: Boolean

Tabular cell

The properties of a tabular cell in a page report are:

Property Name	Description
General	
Class Type	Indicates the class type of the object. This property is read only.
Instance Name	Shows the instance name of the object. This property is read only.
Geometry	
Height	Specifies the height of the object, in inches or centimeters. Enter a numeric value to change the height. Data type: Float
Width	Specifies the width of the object, in inches or centimeters. Enter a numeric value to change the width. Data type: Float
X	Specifies the horizontal coordinate of the top left corner of the object, relative to its parent container. Enter a numeric value to change the position in inches or centimeters. Data type: Float
Y	Specifies the vertical coordinate of the top left corner of the object, relative to its parent container. Enter a numeric value to change the position in inches or centimeters. Data type: Float
Color	
Background	Specifies the background color of the object. Choose a color from the drop-down list, enter a hexadecimal RGB value (for example, 0xff0000), choose/create a formula or edit an expression that returns a color, or select Custom from the drop-down list to specify the color. Data type: String
CSS	
Class	Specifies a CSS class to be applied to the object which is a valid class in the CSS file. Data type: String
Style	The property can be used in two ways. • Specifies a style to be applied to the object. Choose a style from the drop-down list which is available when a Style Group has been selected at the report tab level and when there are styles in the Style Group that can be applied to the object, or type in the style name. • Specifies a CSS selector to be applied to the object. Enter a valid class name from the CSS file. Data type: String
Others	
Export to CSV	Specifies whether to include the object when exporting the report to CSV. Data type: Boolean
Export to Excel	Specifies whether to include the object when exporting the report to Excel. Data type: Boolean
Horizontal Alignment	Specifies the horizontal justification of the contents in the tabular cell. Choose an option from the drop-down list. Data type: Enumeration
Repeat Content	A cell may be extended to new pages for being affected by other cells even though it can be wholly put in the current page. In this case, if the property value is true, then its contents will appear on the new pages again. Data type: Boolean **Note:** If there are two or more cells that will be affected in the same row group and the values of the property for these cells are true, then only the contents of the last one will be repeated.
Vertical Alignment	Specifies the vertical justification of the contents in the tabular cell. Applied only when the web report runs in Web Report Studio. Choose an option from the drop-down list. Data type: Enumeration
Excel	
Column Index	Specifies the X coordinate of the object relative to its parent container when exported to Excel or CSV, measured in cells. The Columned property at the report tab level must be set to true for this property to take effect. Data type: Integer
Row Index	Specifies the Y coordinate of the object relative to its parent container when exported to Excel or CSV, measured in cells. The Columned property at the report tab level must be set to true for this property to take effect. Data type: Integer
Border	
Border Color	Specifies the color of the border of the object. Choose a color from the drop-down list, enter a hexadecimal RGB value (for example, 0xCCAA00), choose/create a formula or edit an expression that returns a color, or select Custom from the drop-down list to specify the color. Data type: String
Border Thickness	Specifies the width of the border in inches or centimeters. Enter a numeric value to change the thickness. Data type: Float
Bottom Line	Specifies the line style of the bottom border of the object. Choose a style from the drop-down list. Data type: Enumeration
Left Line	Specifies the line style of the left border of the object. Choose a style from the drop-down list. Data type: Enumeration
Right Line	Specifies the line style of the right border of the object. Choose a style from the drop-down list. Data type: Enumeration
Top Line	Specifies the line style of the top border of the object. Choose a style from the drop-down list. Data type: Enumeration
TOC	
Anchor Display Value	Specifies a string or formula to display for the TOC entry for this object in the TOC Browser. By default it is the value of the specified object. The TOC Anchor property must be set to true for this property to take effect. Data type: String
TOC Anchor	Specifies whether to include the object in the TOC Browser for the report. Data type: Boolean
Accessibility	
External CSS Class Selector	Specifies a class selector to be applied to the object when exported as HTML. Enter a valid class name from the CSS file. Data type: String

Text box

The properties of a text box in a page report are:

Property Name	Description
General	
Class Type	Indicates the class type of the object. This property is read only.
Instance Name	Shows the instance name of the object. This property is read only.
Geometry	
Height	Specifies the height of the object, in inches or centimeters. Enter a numeric value to change the height. Data type: Float
Width	Specifies the width of the object, in inches or centimeters. Enter a numeric value to change the width. Data type: Float
X	Specifies the horizontal coordinate of the top left corner of the object, relative to its parent container. This property is ignored if the Position property is set to static. Enter a numeric value to change the position in inches or centimeters. Data type: Float
Y	Specifies the vertical coordinate of the top left corner of the object, relative to its parent container. This property is ignored if the Position property is set to static. Enter a numeric value to change the position in inches or centimeters. Data type: Float
Color	
Background	Specifies the background color of the object. Choose a color from the drop-down list, enter a hexadecimal RGB value (for example, 0xff0000), choose/create a formula or edit an expression that returns a color, or select Custom from the drop-down list to specify the color. Data type: String
CSS	
Class	Specifies a CSS class to be applied to the object which is a valid class in the CSS file. Data type: String
Style	The property can be used in two ways. • Specifies a style to be applied to the object. Choose a style from the drop-down list which is available when a Style Group has been selected at the report tab level and when there are styles in the Style Group that can be applied to the object, or type in the style name. • Specifies a CSS selector to be applied to the object. Enter a valid class name from the CSS file. Data type: String
Others	
Export to Applet	Specifies whether to include the object when exporting the report to Applet. Data type: Boolean
Export to CSV	Specifies whether to include the object when exporting the report to CSV. Data type: Boolean
Export to Excel	Specifies whether to include the object when exporting the report to Excel. Data type: Boolean
Export to HTML	Specifies whether to include the object when exporting the report to HTML. Data type: Boolean
Export to PDF	Specifies whether to include the object when exporting the report to PDF. Data type: Boolean
Export to PostScript	Specifies whether to include the object when exporting the report to PostScript. Data type: Boolean
Export to Report Result	Specifies whether to include the object when exporting the report to Report Result. Data type: Boolean
Export to RTF	Specifies whether to include the object when exporting the report to RTF. Data type: Boolean
Export to Text	Specifies whether to include the object when exporting the report to Text. Data type: Boolean
Export to XML	Specifies whether to include the object when exporting the report to XML. Data type: Boolean
Invisible	Specifies whether to show the object in the design area and in the report results. All formulas and calculations will still be performed if the property is set to true. Data type: Boolean
Position	Specifies the position of the object. Choose an option from the drop-down list. • **static** - The object is positioned at the location in which it is inserted. If selected, the X, Y and other position-related properties will be hidden or disabled. The position of the object is affected by preceding objects in the same container. • **relative** - The object is positioned at an offset to the position at which it is inserted. The offset is determined by the X and Y property values. • **absolute** - The object will be located at the position specified by dragging and dropping or by setting its X and Y property values. The position of the object is not affected by preceding objects in the same container. Data type: Enumeration
Record Location	Specifies the calculation point for the properties which use formulas. Choose an option from the drop-down list. • **Default** - The values of the properties are calculated in the default location where they are located. • **Page Header** - The values of the properties are calculated in the page header. • **Page Footer** - The values of the properties are calculated in the page footer. Data type: Enumeration
Suppress	Specifies whether to show the object in the design area and in the report results. All formulas and calculations will be skipped if the property is set to true. Data type: Boolean **Note:** When both the Invisible and Suppress properties of an object are set to true, Suppress has the higher priority.
Suppress When No Records	Specifies whether to display the object in the report results when no record is returned to its parent data container. Data type: Boolean
Vertical Alignment	Specifies the vertical justification of the text in the object. Choose an option from the drop-down list. Data type: Enumeration
Vertical Auto Size	If true, the height of the object will be automatically adjusted according to the size of the contents. Data type: Boolean
Excel	
Column Index	Specifies the X coordinate of the object relative to its parent container when exported to Excel or CSV, measured in cells. The Columned property at the report tab level must be set to true for this property to take effect. Data type: Integer
Row Index	Specifies the Y coordinate of the object relative to its parent container when exported to Excel or CSV, measured in cells. The Columned property at the report tab level must be set to true for this property to take effect. Data type: Integer
Border	
Border Color	Specifies the color of the border of the object. Choose a color from the drop-down list, enter a hexadecimal RGB value (for example, 0xCCAA00), choose/create a formula or edit an expression that returns a color, or select Custom from the drop-down list to specify the color. Data type: String
Border Thickness	Specifies the width of the border in inches or centimeters. Enter a numeric value to change the thickness. Data type: Float
Bottom Line	Specifies the line style of the bottom border of the object. Choose a style from the drop-down list. Data type: Enumeration
Left Line	Specifies the line style of the left border of the object. Choose a style from the drop-down list. Data type: Enumeration
Right Line	Specifies the line style of the right border of the object. Choose a style from the drop-down list. Data type: Enumeration
Top Line	Specifies the line style of the top border of the object. Choose a style from the drop-down list. Data type: Enumeration
TOC	
Anchor Display Value	Specifies a string or formula to display for the TOC entry for this object in the TOC Browser. By default it is the value of the specified object. The TOC Anchor property must be set to true for this property to take effect. Data type: String
TOC Anchor	Specifies whether to include the object in the TOC Browser for the report. Data type: Boolean
Accessibility	
External CSS Class Selector	Specifies a class selector to be applied to the object when exported as HTML. Enter a valid class name from the CSS file. Data type: String

TOC

The properties of a TOC object in a page report are:

Property Name	Description
General	
Class Type	Indicates the class type of the object. This property is read only.
Instance Name	Shows the instance name of the object. This property is read only.
Root Label	Specifies the text of the label which will be displayed on the root of the TOC tree. Data type: String
Color	
Background	Specifies the background color of the object. Choose a color from the drop-down list, enter a hexadecimal RGB value (for example, 0xff0000), choose/create a formula or edit an expression that returns a color, or select Custom from the drop-down list to specify the color. Data type: String
Foreground	Specifies the foreground color of the object. Choose a color from the drop-down list, enter a hexadecimal RGB value (for example, 0xff0000), choose/create a formula or edit an expression that returns a color, or select Custom from the drop-down list to specify the color. Data type: String
CSS	
Class	Specifies a CSS class to be applied to the object which is a valid class in the CSS file. Data type: String
Style	The property can be used in two ways. • Specifies a style to be applied to the object. Choose a style from the drop-down list which is available when a Style Group has been selected at the report tab level and when there are styles in the Style Group that can be applied to the object, or type in the style name. • Specifies a CSS selector to be applied to the object. Enter a valid class name from the CSS file. Data type: String
Font	
Bold	Specifies whether to make the text bold. Data type: Boolean
Font Face	Specifies the font of the text. Choose an option from the drop-down list. Data type: Enumeration
Font Size	Specifies the font size of the text. Enter an integer value to change the size. Data type: Integer
Italic	Specifies whether to make the text italic. Data type: Boolean
Strikethrough	Specifies whether to add a strikethrough line to the text. Data type: Boolean
Underline	Specifies whether to underline the text. Data type: Boolean
Accessibility	
External CSS Class Selector	Specifies a class selector to be applied to the object when exported as HTML. Enter a valid class name from the CSS file. Data type: String
External ID	It is mapped to the HTML attribute *id*. This attribute specifies a name to the object, which must be unique in the report. Data type: String
External Style	It is mapped to the HTML attribute *style*. This attribute specifies style information for the object. Data type: String
External Title	It is mapped to the HTML attribute *title*. This attribute offers advisory information about the object. Data type: String
Language	It is mapped to the HTML attribute *lang*. This attribute specifies the base language of the object's attribute values and text content. Data type: String

UDO

As the name suggests, a UDO is an object defined by users. It has special properties defined by users. The properties listed in JReport Report Inspector correspond with the ones defined your UDO programs. Here are some default properties:

Property Name	Description
General	
Class Type	Indicates the class type of the object. This property is read only.
Instance Name	Shows the instance name of the object. This property is read only.
Geometry	
Height	Specifies the height of the object, in inches or centimeters. Enter a numeric value to change the height. Data type: Float
Width	Specifies the width of the object, in inches or centimeters. Enter a numeric value to change the width. Data type: Float
X	Specifies the horizontal coordinate of the top left corner of the object, relative to its parent container. This property is ignored if the Position property is set to static. Enter a numeric value to change the position in inches or centimeters. Data type: Float
Y	Specifies the vertical coordinate of the top left corner of the object, relative to its parent container. This property is ignored if the Position property is set to static. Enter a numeric value to change the position in inches or centimeters. Data type: Float
Color	
Background	Specifies the background color of the object. Choose a color from the drop-down list, enter a hexadecimal RGB value (for example, 0xff0000), choose/create a formula or edit an expression that returns a color, or select Custom from the drop-down list to specify the color. Data type: String
Foreground	Specifies the foreground color of the object. Choose a color from the drop-down list, enter a hexadecimal RGB value (for example, 0xff0000), choose/create a formula or edit an expression that returns a color, or select Custom from the drop-down list to specify the color. Data type: String
CSS	
Class	Specifies a CSS class to be applied to the object which is a valid class in the CSS file. Data type: String
Style	The property can be used in two ways. • Specifies a style to be applied to the object. Choose a style from the drop-down list which is available when a Style Group has been selected at the report tab level and when there are styles in the Style Group that can be applied to the object, or type in the style name. • Specifies a CSS selector to be applied to the object. Enter a valid class name from the CSS file. Data type: String
Others	
Export to Applet	Specifies whether to include the object when exporting the report to Applet. Data type: Boolean
Export to CSV	Specifies whether to include the object when exporting the report to CSV. If it is set to true, only the text value will be exported. Data type: Boolean
Export to Excel	Specifies whether to include the object when exporting the report to Excel. If it is set to true, only the text value will be exported. Data type: Boolean
Export to HTML	Specifies whether to include the object when exporting the report to HTML. Data type: Boolean
Export to PDF	Specifies whether to include the object when exporting the report to PDF. Data type: Boolean
Export to PostScript	Specifies whether to include the object when exporting the report to PostScript. Data type: Boolean
Export to Report Result	Specifies whether to include the object when exporting the report to Report Result. Data type: Boolean
Export to RTF	Specifies whether to include the object when exporting the report to RTF. Data type: Boolean
Export to Text	Specifies whether to include the object when exporting the report to Text. If it is set to true, only the text value will be exported. Data type: Boolean
Export to XML	Specifies whether to include the object when exporting the report to XML. If it is set to true, only the text value will be exported. Data type: Boolean
Invisible	Specifies whether to show the object in the design area and in the report results. All formulas and calculations will still be performed if the property is set to true. Data type: Boolean
Position	Specifies the position of the object. Choose an option from the drop-down list. • **static** - The object is positioned at the location in which it is inserted. The X and Y properties are ignored. The position of the object is affected by preceding objects in the same container. • **relative** - The object is positioned at an offset to the position at which it is inserted. The offset is determined by the X and Y property values. • **absolute** - The object will be located at the position specified by dragging and dropping or by setting its X and Y property values. The position of the object is not affected by preceding objects in the same container. Data type: Enumeration
Record Location	Specifies the calculation point for the properties which use formulas. Choose an option from the drop-down list. • **Default** - The values of the properties are calculated in the default location where they are located. • **Page Header** - The values of the properties are calculated in the page header. • **Page Footer** - The values of the properties are calculated in the page footer. Data type: Enumeration
Suppress	Specifies whether to show the object in the design area and in the report results. All formulas and calculations will be skipped if the property is set to true. Data type: Boolean **Note:** When both the Invisible and Suppress properties of an object are set to true, Suppress has the higher priority.
Suppress When No Records	Specifies whether to display the object in the report results when no record is returned to its parent data container. Data type: Boolean
Excel	
Column Index	Specifies the X coordinate of the object relative to its parent container when exported to Excel or CSV, measured in cells. The Columned property at the report tab level must be set to true for this property to take effect. Data type: Integer
Column Number	Specifies the number of columns which will be the object's width in the exported Excel file. The Columned property at the report tab level must be set to true for this property to take effect. Data type: Integer
Row Index	Specifies the Y coordinate of the object relative to its parent container when exported to Excel or CSV, measured in cells. The Columned property at the report tab level must be set to true for this property to take effect. Data type: Integer
Row Number	Specifies the number of rows which will be the object's height in the exported Excel file. The Columned property at the report tab level must be set to true for this property to take effect. Data type: Integer
Font	
Bold	Specifies whether to make the text bold. Data type: Boolean
Font Face	Specifies the font of the text. Choose an option from the drop-down list. Data type: Enumeration
Font Size	Specifies the font size of the text. Enter an integer value to change the size. Data type: Integer
Italic	Specifies whether to make the text italic. Data type: Boolean
Accessibility	
External CSS Class Selector	Specifies a class selector to be applied to the object when exported as HTML. Enter a valid class name from the CSS file. Data type: String

Notes:

- The above are the basic properties of a UDO and more will be added with those defined in your program.
- JReport UDO does not support RTF and PDF format output. That is, when exporting to RTF and PDF, a report with a UDO may not be rendered correctly. However, there is no such problem for HTML and PS.

Built-in UDOs

JReport provides you with two built-in UDOs: JHyperLink and JRotator.

JHyperLink

It operates in a similar way to the hyperlink in an HTML file (only for Windows NT). Here are its special properties:

Property Name	Description
Others	
Auto Fit	Specifies whether to adjust the width and height of the object according to the contents. Data type: Boolean
Display Value	Specifies the displayed text of the hyperlink. This property has lower priority than Display Image. Data type: String
Enable Hyperlink in Excel	Specifies whether to enable the link defined on the object when exporting the report to an Excel file. Data type: Boolean
Enable Hyperlink in HTML	Specifies whether to enable the link defined on the object when exporting the report to an HTML file. Data type: Boolean
Enable Hyperlink in PDF	Specifies whether to enable the link defined on the object when exporting the report to a PDF file. Data type: Boolean
Executer Class Name	If the specified URL is linked to a .rst file - the JReport Result export result file, you should select jet.udos.RPTExecuter. For other links, use jet.udos.IEExecuter. Data type: String
Horizontal Alignment	Specifies the horizontal justification of the content in the JHyperLink container. Choose an option from the drop-down list. Data type: Enumeration

HTML Style	Specifies the CSS style of the JHyperLink in HTML result and in Page Report Studio. The value is a String value containing only the body part of a CSS style definition. Make sure that you input the right css style string.
	For Example, if a full style definition is:
	`style="text-decoration:underline;color:#ffc000;font:italic small-caps bold 12pt serif;"`
	you need only input the content in the quotation marks as the HTML Style value:
	`text-decoration:underline;color:#ffc000;font:italic small-caps bold 12pt serif;`
	Data type: String

Target	Specifies the window or frame at which to target contents. Choose an option from the drop-down list. • **blank** - Loads the linked document into a new blank window. This window is not named. • **self** - Loads the linked document into the window in which the link is clicked (the active window). • **parent** - Loads the linked document into the immediate parent window of the active window. If there is no parent window, the active window is used. • **top** - Loads the linked document into the topmost window. Data type: Enumeration
Tooltips Text	Specifies the text that will be shown when you hover the mouse pointer over the JHyperLink in Designer view mode, in HTML result, or in Page Report Studio. Data type: String
URL	Specifies the URL for accessing the linked file or website page. Data type: String
Vertical Alignment	Specifies the vertical justification of the content in the JHyperLink container. Choose an option from the drop-down list. Data type: Enumeration
Word Wrap	Specifies whether to enable the word wrap function of JHyperLink in the exported files. Data type: Boolean
Image	
Alternate Text	Specifies the text that will be displayed instead if the image cannot be displayed. Data type: String
Display Image	Specifies the local path of the image displayed for the hyperlink (like a hotspot in HTML). This property has higher priority than Display Value. Data type: String
Original	Specifies from where the image is fetched when viewing or running the report on JReport Server. The property has following two options: • **true** - If selected, the image will not be delivered when publishing the report to JReport Server. JReport Server will find the image from the local directory specified by the property Display Image when you view the report. • **false** - If selected, the image will be delivered when publishing the report to JReport Server. JReport Server will obtain the image from the server side when you view or run the report. Data type: Boolean
Accessibility	
External AccessKey	It is mapped to the HTML attribute accesskey. This attribute specifies an access key to the object. Data type: String
External CSS Class Selector	Specifies a class selector to be applied to the object when exported as HTML. Enter a valid class name from the CSS file. Data type: String
HrefLang	It is mapped to the HTML attribute hreflang. This attribute specifies the base language of the resource designated by a link and may only be used when a link is specified. Data type: String

Example:

Insert two JHyperLink UDOs, and specify the following:

- URL: `http://www.jinfonet.com`
  Display Value: Click here.
- URL: `C:\docword.txt`
  Display Value: abc
  Display Image: `C:\JReport\Designer\Demo\SampleReports\Coffee.jpg`

When you point your mouse at either UDO, a hand will show you that it is a hyperlink. By clicking it, you will be taken to the destination. The application launched to open it (the linked file) is from your Windows NT system. If Display Value and Display Image both have been specified, the image has the higher priority to be displayed.

JRotator

It is a built-in UDO that can be rotated. Text and images can be held in JRotator. Following are its special properties:

Property Name	Description
Display Image	Specifies the local path of the image displayed in the JRotator. This property has higher priority than Display Value. Data type: String
Display Value	Specifies the text displayed in the JRotator. This property has lower priority than Display Image. Data type: String
Horizontal Alignment	Specifies the horizontal justification of the text in the JRotator container. Choose an option from the drop-down list. Data type: Enumeration
Rotate	Specifies the angle at which to rotate the JRotator, in degrees. • 0 - No rotation. • positive value - Rotates the JRotator clockwise. • negative value - Rotates the JRotator anticlockwise. Data type: Integer
Vertical Alignment	Specifies the vertical justification of the text in the JRotator container. Choose an option from the drop-down list. Data type: Enumeration

Web control

The properties of web controls are listed as follows:

- Filter control
- Multivalue container
- Navigation control
- Parameter control
- Parameter form control

Filter control

The properties of a filter control in a page report are:

Property Name	Description
General	
Class Type	Indicates the class type of the object. This property is read only.
Instance Name	Shows the instance name of the object. This property is read only.
Geometry	
Height	Specifies the height of the object, in inches or centimeters. Enter a numeric value to change the height. Data type: Float
Width	Specifies the width of the object, in inches or centimeters. Enter a numeric value to change the width. Data type: Float
X	Specifies the horizontal coordinate of the top left corner of the object, relative to its parent container. This property is ignored if the Position property is set to static. Enter a numeric value to change the position in inches or centimeters. Data type: Float
Y	Specifies the vertical coordinate of the top left corner of the object, relative to its parent container. This property is ignored if the Position property is set to static. Enter a numeric value to change the position in inches or centimeters. Data type: Float
Color	
Background	Specifies the background color of the object. Choose a color from the drop-down list, enter a hexadecimal RGB value (for example, 0xff0000), choose/create a formula or edit an expression that returns a color, or select Custom from the drop-down list to specify the color. Data type: String
Foreground	Specifies the foreground color of the object. Choose a color from the drop-down list, enter a hexadecimal RGB value (for example, 0xff0000), choose/create a formula or edit an expression that returns a color, or select Custom from the drop-down list to specify the color. Data type: String
CSS	
Class	Specifies a CSS class to be applied to the object which is a valid class in the CSS file. Data type: String
ID	Specifies the identifier of the object, which must be unique in the report. The ID property can be a style sheet selector. Data type: String
Style	The property can be used in two ways. • Specifies a style to be applied to the object. Choose a style from the drop-down list which is available when a Style Group has been selected at the report tab level and when there are styles in the Style Group that can be applied to the object, or type in the style name. • Specifies a CSS selector to be applied to the object. Enter a valid class name from the CSS file. Data type: String
Others	
Export to Applet	Specifies whether to include the object when exporting the report to Applet. Data type: Boolean
Export to CSV	Specifies whether to include the object when exporting the report to CSV. Data type: Boolean
Export to Excel	Specifies whether to include the object when exporting the report to Excel. Data type: Boolean
Export to HTML	Specifies whether to include the object when exporting the report to HTML. Data type: Boolean
Export to PDF	Specifies whether to include the object when exporting the report to PDF. Data type: Boolean
Export to PostScript	Specifies whether to include the object when exporting the report to PostScript. Data type: Boolean
Export to Report Result	Specifies whether to include the object when exporting the report to Report Result. Data type: Boolean
Export to RTF	Specifies whether to include the object when exporting the report to RTF. Data type: Boolean
Export to Text	Specifies whether to include the object when exporting the report to Text. Data type: Boolean
Export to XML	Specifies whether to include the object when exporting the report to XML. Data type: Boolean
Invisible	Specifies whether to show the object in the design area and in the report results. All formulas and calculations will still be performed if the property is set to true. Data type: Boolean
Position	Specifies the position of the object. Choose an option from the drop-down list. • **static** - The object is positioned at the location in which it is inserted. If selected, the X, Y and other position-related properties will be hidden or disabled. The position of the object is affected by preceding objects in the same container. • **relative** - The object is positioned at an offset to the position at which it is inserted. The offset is determined by the X and Y property values. • **absolute** - The object will be located at the position specified by dragging and dropping or by setting its X and Y property values. The position of the object is not affected by preceding objects in the same container. Data type: Enumeration
Excel	
Column Index	Specifies the X coordinate of the object relative to its parent container when exported to Excel or CSV, measured in cells. The Columned property at the report tab level must be set to true for this property to take effect. Data type: Integer
Row Index	Specifies the Y coordinate of the object relative to its parent container when exported to Excel or CSV, measured in cells. The Columned property at the report tab level must be set to true for this property to take effect. Data type: Integer
Border	
Border Color	Specifies the color of the border of the object. Choose a color from the drop-down list, enter a hexadecimal RGB value (for example, 0xCCAA00), choose/create a formula or edit an expression that returns a color, or select Custom from the drop-down list to specify the color. Data type: String
Border Thickness	Specifies the width of the border in inches or centimeters. Enter a numeric value to change the thickness. Data type: Float
Bottom Line	Specifies the line style of the bottom border of the object. Choose a style from the drop-down list. Data type: Enumeration
Left Line	Specifies the line style of the left border of the object. Choose a style from the drop-down list. Data type: Enumeration
Right Line	Specifies the line style of the right border of the object. Choose a style from the drop-down list. Data type: Enumeration
Top Line	Specifies the line style of the top border of the object. Choose a style from the drop-down list. Data type: Enumeration
Title	
Background	Specifies the background color of the title bar. Choose a color from the drop-down list, enter a hexadecimal RGB value (for example, 0xff0000), choose/create a formula or edit an expression that returns a color, or select Custom from the drop-down list to specify the color. Data type: String
Bold	Specifies whether to make the text bold in the title bar. Data type: Boolean
Font Face	Specifies the font of the text in the title bar. Choose an option from the drop-down list. Data type: Enumeration
Font Size	Specifies the font size of the text in the title bar. Enter an integer value to change the size. Data type: Integer
Foreground	Specifies the foreground color of the title bar. Choose a color from the drop-down list, enter a hexadecimal RGB value (for example, 0xff0000), choose/create a formula or edit an expression that returns a color, or select Custom from the drop-down list to specify the color. Data type: String
Horizontal Alignment	Specifies the horizontal justification of the text in the title bar. Choose an option from the drop-down list. Data type: Enumeration
Italic	Specifies whether to make the text italic in the title bar. Data type: Boolean
Map Name	Specifies whether to use the name of the DBField added to the filter control as the text displayed in the title bar. Data type: Boolean
Show Title	Specifies whether to show the title bar. It is meaningful to set all the other properties in the Title category when this property is set to true. Data type: Boolean
Text	Specifies the text displayed in the title bar. This property can be edited only when Map Name is set to false. Enter a string to change the text. Data type: String
Underline	Specifies whether to underline the text in the title bar. Data type: Boolean
Text Format	
Bold	Specifies whether to make the text bold. Data type: Boolean
Font Face	Specifies the font of the text. Choose an option from the drop-down list. Data type: Enumeration
Font Size	Specifies the font size of the text. Enter an integer value to change the size. Data type: Integer
Format	Specifies the display format of the text in the report result. It varies with data type and can be manually defined. Data type: String **Note:** For the BigDecimal type, to avoid precision loss, you should enter a prefix JRD when setting the format property value.
Italic	Specifies whether to make the text italic. Data type: Boolean
Underline	Specifies whether to underline the text. Data type: Boolean
Accessibility	
External CSS Class Selector	Specifies a class selector to be applied to the object when exported as HTML. Enter a valid class name from the CSS file. Data type: String
External ID	It is mapped to the HTML attribute *id*. This attribute specifies a name to the object, which must be unique in the report. Data type: String
External Style	It is mapped to the HTML attribute *style*. This attribute specifies style information for the object. Data type: String
External Title	It is mapped to the HTML attribute *title*. This attribute offers advisory information about the object. Data type: String
Language	It is mapped to the HTML attribute *lang*. This attribute specifies the base language of the object's attribute values and text content. Data type: String

Multivalue container

The properties of a multivalue container object in a page report are:

Property Name	Description
General	
Class Type	Indicates the class type of the object. This property is read only.
Data Inherit	Indicates whether the dataset for this object is inherited from another object. This property is read only.
Dataset	Indicates the dataset used by the object. This property is read only.
Instance Name	Shows the instance name of the object. This property is read only.
Geometry	
Height	Specifies the height of the object, in inches or centimeters. Enter a numeric value to change the height. Data type: Float
Width	Specifies the width of the object, in inches or centimeters. Enter a numeric value to change the width. Data type: Float
X	Specifies the horizontal coordinate of the top left corner of the object, relative to its parent container. This property is ignored if the Position property is set to static. Enter a numeric value to change the position in inches or centimeters. Data type: Float
Y	Specifies the vertical coordinate of the top left corner of the object, relative to its parent container. This property is ignored if the Position property is set to static. Enter a numeric value to change the position in inches or centimeters. Data type: Float
Color	
Background	Specifies the background color of the object. Choose a color from the drop-down list, enter a hexadecimal RGB value (for example, 0xff0000), choose/create a formula or edit an expression that returns a color, or select Custom from the drop-down list to specify the color. Data type: String
CSS	
Class	Specifies a CSS class to be applied to the object which is a valid class in the CSS file. Data type: String
ID	Specifies the identifier of the object, which must be unique in the report. The ID property can be a style sheet selector. Data type: String
Style	The property can be used in two ways. • Specifies a style to be applied to the object. Choose a style from the drop-down list which is available when a Style Group has been selected at the report tab level and when there are styles in the Style Group that can be applied to the object, or type in the style name. • Specifies a CSS selector to be applied to the object. Enter a valid class name from the CSS file. Data type: String
Others	
Export to Applet	Specifies whether to include the object when exporting the report to Applet. Data type: Boolean
Export to CSV	Specifies whether to include the object when exporting the report to CSV. Data type: Boolean
Export to Excel	Specifies whether to include the object when exporting the report to Excel. Data type: Boolean
Export to HTML	Specifies whether to include the object when exporting the report to HTML. Data type: Boolean
Export to PDF	Specifies whether to include the object when exporting the report to PDF. Data type: Boolean
Export to PostScript	Specifies whether to include the object when exporting the report to PostScript. Data type: Boolean
Export to Report Result	Specifies whether to include the object when exporting the report to Report Result. Data type: Boolean
Export to RTF	Specifies whether to include the object when exporting the report to RTF. Data type: Boolean
Export to Text	Specifies whether to include the object when exporting the report to Text. Data type: Boolean
Export to XML	Specifies whether to include the object when exporting the report to XML. Data type: Boolean
Invisible	Specifies whether to show the object in the design area and in the report results. All formulas and calculations will still be performed if the property is set to true. Data type: Boolean
Position	Specifies the position of the object. Choose an option from the drop-down list. • **static** - The object is positioned at the location in which it is inserted. If selected, the X, Y and other position-related properties will be hidden or disabled. The position of the object is affected by preceding objects in the same container. • **relative** - The object is positioned at an offset to the position at which it is inserted. The offset is determined by the X and Y property values. • **absolute** - The object will be located at the position specified by dragging and dropping or by setting its X and Y property values. The position of the object is not affected by preceding objects in the same container. Data type: Enumeration
Record Location	Specifies the calculation point for the properties which use formulas. Choose an option from the drop-down list. • **Default** - The values of the properties are calculated in the default location where they are located. • **Page Header** - The values of the properties are calculated in the page header. • **Page Footer** - The values of the properties are calculated in the page footer. Data type: Enumeration
Suppress	Specifies whether to show the object in the design area and in the report results. All formulas and calculations will be skipped if the property is set to true. Data type: Boolean **Note:** When both the Invisible and Suppress properties of an object are set to true, Suppress has the higher priority.
Suppress When No Records	Specifies whether to display the object in the report results when no record is returned to its parent data container. Data type: Boolean
Excel	
Column Index	Specifies the X coordinate of the object relative to its parent container when exported to Excel or CSV, measured in cells. The Columned property at the report tab level must be set to true for this property to take effect. Data type: Integer
Row Index	Specifies the Y coordinate of the object relative to its parent container when exported to Excel or CSV, measured in cells. The Columned property at the report tab level must be set to true for this property to take effect. Data type: Integer
Border	
Border Color	Specifies the color of the border of the object. Choose a color from the drop-down list, enter a hexadecimal RGB value (for example, 0xCCAA00), choose/create a formula or edit an expression that returns a color, or select Custom from the drop-down list to specify the color. Data type: String
Border Thickness	Specifies the width of the border in inches or centimeters. Enter a numeric value to change the thickness. Data type: Float
Bottom Line	Specifies the line style of the bottom border of the object. Choose a style from the drop-down list. Data type: Enumeration
Left Line	Specifies the line style of the left border of the object. Choose a style from the drop-down list. Data type: Enumeration
Right Line	Specifies the line style of the right border of the object. Choose a style from the drop-down list. Data type: Enumeration
Top Line	Specifies the line style of the top border of the object. Choose a style from the drop-down list. Data type: Enumeration
Text Format	
Bold	Specifies whether to make the text bold. Data type: Boolean
Font Face	Specifies the text font of the items in the multivalue container. Choose an option from the drop-down list. Data type: Enumeration
Font Size	Specifies the text font size of the items. Enter an integer value to change the size. Data type: Integer
Foreground	Specifies the color of the item text. Choose a color from the drop-down list, enter a hexadecimal RGB value (for example, 0xff0000) , choose/create a formula or edit an expression that returns a color, or select Custom from the drop-down list to specify the color and fill effect. Data type: String
Italic	Specifies whether to make the text italic. Data type: Boolean
Accessibility	
External CSS Class Selector	Specifies a class selector to be applied to the object when exported as HTML. Enter a valid class name from the CSS file. Data type: String
External Style	It is mapped to the HTML attribute *style*. This attribute specifies style information for the object. Data type: String
External Title	It is mapped to the HTML attribute *title*. This attribute offers advisory information about the object. Data type: String
Language	It is mapped to the HTML attribute *lang*. This attribute specifies the base language of the object's attribute values and text content. Data type: String

Navigation control

The properties of a navigation control in a page report are:

Property Name	Description
General	
Class Type	Indicates the class type of the object. This property is read only.
Instance Name	Shows the instance name of the object. This property is read only.
Geometry	
Height	Specifies the height of the object, in inches or centimeters. Enter a numeric value to change the height. Data type: Float
Width	Specifies the width of the object, in inches or centimeters. Enter a numeric value to change the width. Data type: Float
X	Specifies the horizontal coordinate of the top left corner of the object, relative to its parent container. This property is ignored if the Position property is set to static. Enter a numeric value to change the position in inches or centimeters. Data type: Float
Y	Specifies the vertical coordinate of the top left corner of the object, relative to its parent container. This property is ignored if the Position property is set to static. Enter a numeric value to change the position in inches or centimeters. Data type: Float
Color	
Background	Specifies the background color of the object. Choose a color from the drop-down list, enter a hexadecimal RGB value (for example, 0xff0000) , choose/create a formula or edit an expression that returns a color, or select Custom from the drop-down list to specify the color. Data type: String
CSS	
Class	Specifies a CSS class to be applied to the object which is a valid class in the CSS file. Data type: String
Style	The property can be used in two ways. • Specifies a style to be applied to the object. Choose a style from the drop-down list which is available when a Style Group has been selected at the report tab level and when there are styles in the Style Group that can be applied to the object, or type in the style name. • Specifies a CSS selector to be applied to the object. Enter a valid class name from the CSS file. Data type: String
Others	
Export to Applet	Specifies whether to include the object when exporting the report to Applet. Data type: Boolean
Export to CSV	Specifies whether to include the object when exporting the report to CSV. Data type: Boolean
Export to Excel	Specifies whether to include the object when exporting the report to Excel. Data type: Boolean
Export to HTML	Specifies whether to include the object when exporting the report to HTML. Data type: Boolean
Export to PDF	Specifies whether to include the object when exporting the report to PDF. Data type: Boolean
Export to PostScript	Specifies whether to include the object when exporting the report to PostScript. Data type: Boolean
Export to Report Result	Specifies whether to include the object when exporting the report to Report Result. Data type: Boolean
Export to RTF	Specifies whether to include the object when exporting the report to RTF. Data type: Boolean
Export to Text	Specifies whether to include the object when exporting the report to Text. Data type: Boolean
Export to XML	Specifies whether to include the object when exporting the report to XML. Data type: Boolean
Invisible	Specifies whether to show the object in the design area and in the report results. All formulas and calculations will still be performed if the property is set to true. Data type: Boolean
Position	Specifies the position of the object. Choose an option from the drop-down list. • **static** - The object is positioned at the location in which it is inserted. The X and Y properties are ignored. The position of the object is affected by preceding objects in the same container. • **relative** - The object is positioned at an offset to the position at which it is inserted. The offset is determined by the X and Y property values. • **absolute** - The object will be located at the position specified by dragging and dropping or by setting its X and Y property values. The position of the object is not affected by preceding objects in the same container. Data type: Enumeration
Excel	
Column Index	Specifies the X coordinate of the object relative to its parent container when exported to Excel or CSV, measured in cells. The Columned property at the report tab level must be set to true for this property to take effect. Data type: Integer
Row Index	Specifies the Y coordinate of the object relative to its parent container when exported to Excel or CSV, measured in cells. The Columned property at the report tab level must be set to true for this property to take effect. Data type: Integer
Border	
Border Color	Specifies the color of the border of the object. Choose a color from the drop-down list, enter a hexadecimal RGB value (for example, 0xCCAA00), choose/create a formula or edit an expression that returns a color, or select Custom from the drop-down list to specify the color. Data type: String
Border Thickness	Specifies the width of the border in inches or centimeters. Enter a numeric value to change the thickness. Data type: Float
Bottom Line	Specifies the line style of the bottom border of the object. Choose a style from the drop-down list. Data type: Enumeration
Left Line	Specifies the line style of the left border of the object. Choose a style from the drop-down list. Data type: Enumeration
Right Line	Specifies the line style of the right border of the object. Choose a style from the drop-down list. Data type: Enumeration
Top Line	Specifies the line style of the top border of the object. Choose a style from the drop-down list. Data type: Enumeration

A navigation control has a child object:

- Navigation item

Navigation item

The properties of a navigation item in a page report are:

Property Name	Description
General	
Class Type	Indicates the class type of the object. This property is read only.
Instance Name	Shows the instance name of the object. This property is read only.
Text Format	
Auto Fit	Specifies whether to adjust the width and height of the object according to the contents. Data type: Boolean
Bold	Specifies whether to make the text bold. Data type: Boolean
Font Face	Specifies the font of the text. Choose an option from the drop-down list. Data type: Enumeration
Font Size	Specifies the font size of the text. Enter an integer value to change the size. Data type: Integer
Horizontal Alignment	Specifies the horizontal justification of the text in the object. Choose an option from the drop-down list. Data type: Enumeration
Italic	Specifies whether to make the text italic. Data type: Boolean
Strikethrough	Specifies whether to add a strikethrough line to the text. Data type: Boolean
Text	Specifies the text in the object. Enter a string to display as the object text. Data type: String
Underline	Specifies whether to underline the text. Data type: Boolean
Vertical Alignment	Specifies the vertical justification of the text in the object. Choose an option from the drop-down list. Data type: Enumeration
Word Wrap	Specifies whether to wrap the text to the width. Data type: Boolean
Color	
Background	Specifies the background color of the object. Choose a color from the drop-down list, enter a hexadecimal RGB value (for example, 0xff0000), choose/create a formula or edit an expression that returns a color, or select Custom from the drop-down list to specify the color. Data type: String
Foreground	Specifies the foreground color of the object. Choose a color from the drop-down list, enter a hexadecimal RGB value (for example, 0xff0000), choose/create a formula or edit an expression that returns a color, or select Custom from the drop-down list to specify the color. Data type: String
Others	
Export to Applet	Specifies whether to include the object when exporting the report to Applet. Data type: Boolean
Export to CSV	Specifies whether to include the object when exporting the report to CSV. Data type: Boolean
Export to Excel	Specifies whether to include the object when exporting the report to Excel. Data type: Boolean
Export to HTML	Specifies whether to include the object when exporting the report to HTML. Data type: Boolean
Export to PDF	Specifies whether to include the object when exporting the report to PDF. Data type: Boolean
Export to PostScript	Specifies whether to include the object when exporting the report to PostScript. Data type: Boolean
Export to Report Result	Specifies whether to include the object when exporting the report to Report Result. Data type: Boolean
Export to RTF	Specifies whether to include the object when exporting the report to RTF. Data type: Boolean
Export to Text	Specifies whether to include the object when exporting the report to Text. Data type: Boolean
Export to XML	Specifies whether to include the object when exporting the report to XML. Data type: Boolean
Padding	
Bottom Padding	Specifies the space in inches or centimeters between the text and the bottom border of the object. Enter a numeric value to change the padding. Data type: Float
Left Padding	Specifies the space in inches or centimeters between the text and the left border of the object. Enter a numeric value to change the padding. Data type: Float
Right Padding	Specifies the space in inches or centimeters between the text and the right border of the object. Enter a numeric value to change the padding. Data type: Float
Top Padding	Specifies the space in inches or centimeters between the text and the top border of the object. Enter a numeric value to change the padding. Data type: Float
Border	
Border Color	Specifies the color of the border of the object. Choose a color from the drop-down list, enter a hexadecimal RGB value (for example, 0xCCAA00), choose/create a formula or edit an expression that returns a color, or select Custom from the drop-down list to specify the color. Data type: String
Border Thickness	Specifies the width of the border in inches or centimeters. Enter a numeric value to change the thickness. Data type: Float
Bottom Line	Specifies the line style of the bottom border of the object. Choose a style from the drop-down list. Data type: Enumeration
Left Line	Specifies the line style of the left border of the object. Choose a style from the drop-down list. Data type: Enumeration
Right Line	Specifies the line style of the right border of the object. Choose a style from the drop-down list. Data type: Enumeration
Shadow	Specifies whether to draw two shadowed borders, beneath and to the right of the object. Data type: Boolean
Shadow Color	Specifies the color of the shadow. Choose a color from the drop-down list, enter a hexadecimal RGB value (for example, 0xff0000), choose/create a formula or edit an expression that returns a color, or select Custom from the drop-down list to specify the color and fill effect. Data type: String
Top Line	Specifies the line style of the top border of the object. Choose a style from the drop-down list. Data type: Enumeration
Pattern	
Pattern Color	Specifies the color in which to draw a pattern to fill the object. Choose a color from the drop-down list, enter a hexadecimal RGB value (for example, 0xff0000), choose/create a formula or edit an expression that returns a color, or select Custom from the drop-down list to specify the color and fill effect. Data type: String
Pattern Style	Specifies the style of the pattern. Choose an option from the drop-down list. • **none** - No pattern will be applied to the object. • **50**% - The object will be filled in the Pattern Color of 50%-transparency. • **horizontal** - The pattern will be made of horizontal lines of the Pattern Color. • **vertical** - The pattern will be made of vertical lines of the Pattern Color. • **grid** - The pattern will be an overlapping of horizontal and vertical lines of the Pattern Color. • **diagonal** - The pattern will be made of diagonal lines of the Pattern Color. Data type: Enumeration

Parameter control

The properties of a parameter control in a page report are:

Property Name	Description
General	
Class Type	Indicates the class type of the object. This property is read only.
Instance Name	Shows the instance name of the object. This property is read only.
Geometry	
Height	Specifies the height of the object, in inches or centimeters. Enter a numeric value to change the height. Data type: Float
Width	Specifies the width of the object, in inches or centimeters. Enter a numeric value to change the width. Data type: Float
X	Specifies the horizontal coordinate of the top left corner of the object, relative to its parent container. This property is ignored if the Position property is set to static. Enter a numeric value to change the position in inches or centimeters. Data type: Float
Y	Specifies the vertical coordinate of the top left corner of the object, relative to its parent container. This property is ignored if the Position property is set to static. Enter a numeric value to change the position in inches or centimeters. Data type: Float
Color	
Background	Specifies the background color of the object. Choose a color from the drop-down list, enter a hexadecimal RGB value (for example, 0xff0000), choose/create a formula or edit an expression that returns a color, or select Custom from the drop-down list to specify the color. Data type: String
Foreground	Specifies the foreground color of the object. Choose a color from the drop-down list, enter a hexadecimal RGB value (for example, 0xff0000), choose/create a formula or edit an expression that returns a color, or select Custom from the drop-down list to specify the color. Data type: String
CSS	
Class	Specifies a CSS class to be applied to the object which is a valid class in the CSS file. Data type: String
ID	Specifies the identifier of the object, which must be unique in the report. The ID property can be a style sheet selector. Data type: String
Style	The property can be used in two ways. • Specifies a style to be applied to the object. Choose a style from the drop-down list which is available when a Style Group has been selected at the report tab level and when there are styles in the Style Group that can be applied to the object, or type in the style name. • Specifies a CSS selector to be applied to the object. Enter a valid class name from the CSS file. Data type: String
Excel	
Column Index	Specifies the X coordinate of the object relative to its parent container when exported to Excel or CSV, measured in cells. The Columned property at the report tab level must be set to true for this property to take effect. Data type: Integer
Row Index	Specifies the Y coordinate of the object relative to its parent container when exported to Excel or CSV, measured in cells. The Columned property at the report tab level must be set to true for this property to take effect. Data type: Integer
Border	
Border Color	Specifies the color of the border of the object. Choose a color from the drop-down list, enter a hexadecimal RGB value (for example, 0xCCAA00), choose/create a formula or edit an expression that returns a color, or select Custom from the drop-down list to specify the color. Data type: String
Border Thickness	Specifies the width of the border in inches or centimeters. Enter a numeric value to change the thickness. Data type: Float
Bottom Line	Specifies the line style of the bottom border of the object. Choose a style from the drop-down list. Data type: Enumeration
Left Line	Specifies the line style of the left border of the object. Choose a style from the drop-down list. Data type: Enumeration
Right Line	Specifies the line style of the right border of the object. Choose a style from the drop-down list. Data type: Enumeration
Shadow	Specifies whether to draw two shadowed borders, beneath and to the right of the object. Data type: Boolean
Shadow Color	Specifies the color of the shadow. Choose a color from the drop-down list, enter a hexadecimal RGB value (for example, 0xff0000), choose/create a formula or edit an expression that returns a color, or select Custom from the drop-down list to specify the color and fill effect. Data type: String
Top Line	Specifies the line style of the top border of the object. Choose a style from the drop-down list. Data type: Enumeration
Pattern	
Pattern Color	Specifies the color in which to draw a pattern to fill the object. Choose a color from the drop-down list, enter a hexadecimal RGB value (for example, 0xff0000), choose/create a formula or edit an expression that returns a color, or select Custom from the drop-down list to specify the color and fill effect. Data type: String
Pattern Style	Specifies the style of the pattern. Choose an option from the drop-down list. • **none** - No pattern will be applied to the object. • **50**% - The object will be filled in the Pattern Color of 50%-transparency. • **horizontal** - The pattern will be made of horizontal lines of the Pattern Color. • **vertical** - The pattern will be made of vertical lines of the Pattern Color. • **grid** - The pattern will be an overlapping of horizontal and vertical lines of the Pattern Color. • **diagonal** - The pattern will be made of diagonal lines of the Pattern Color. Data type: Enumeration
Text Format	
Bold	Specifies whether to make the text bold. Data type: Boolean
Font Face	Specifies the font of the text. Choose an option from the drop-down list. Data type: Enumeration
Font Size	Specifies the font size of the text. Enter an integer value to change the size. Data type: Integer
Format	Specifies the display format of the text in the report result. It varies with data type and can be manually defined. Data type: String **Note:** For the BigDecimal type, to avoid precision loss, you should enter a prefix JRD when setting the format property value.
Horizontal Alignment	Specifies the horizontal justification of the text in the object. Choose an option from the drop-down list. Data type: Enumeration
Italic	Specifies whether to make the text italic. Data type: Boolean
Underline	Specifies whether to underline the text. Data type: Boolean
Vertical Alignment	Specifies the vertical justification of the text in the object. Choose an option from the drop-down list. Data type: Enumeration
Others	
Export to Applet	Specifies whether to include the object when exporting the report to Applet. Data type: Boolean
Export to CSV	Specifies whether to include the object when exporting the report to CSV. Data type: Boolean
Export to Excel	Specifies whether to include the object when exporting the report to Excel. Data type: Boolean
Export to HTML	Specifies whether to include the object when exporting the report to HTML. Data type: Boolean
Export to PDF	Specifies whether to include the object when exporting the report to PDF. Data type: Boolean
Export to PostScript	Specifies whether to include the object when exporting the report to PostScript. Data type: Boolean
Export to Report Result	Specifies whether to include the object when exporting the report to Report Result. Data type: Boolean
Export to RTF	Specifies whether to include the object when exporting the report to RTF. Data type: Boolean
Export to Text	Specifies whether to include the object when exporting the report to Text. Data type: Boolean
Export to XML	Specifies whether to include the object when exporting the report to XML. Data type: Boolean
Invisible	Specifies whether to show the object in the design area and in the report results. All formulas and calculations will still be performed if the property is set to true. Data type: Boolean
Parameter	Specifies the value of the parameter. Data type: String
Position	Specifies the position of the object. Choose an option from the drop-down list. • **static** - The object is positioned at the location in which it is inserted. If selected, the X, Y and other position-related properties will be hidden or disabled. The position of the object is affected by preceding objects in the same container. • **relative** - The object is positioned at an offset to the position at which it is inserted. The offset is determined by the X and Y property values. • **absolute** - The object will be located at the position specified by dragging and dropping or by setting its X and Y property values. The position of the object is not affected by preceding objects in the same container. Data type: Enumeration
Accessibility	
External CSS Class Selector	Specifies a class selector to be applied to the object when exported as HTML. Enter a valid class name from the CSS file. Data type: String
External Style	It is mapped to the HTML attribute *style*. This attribute specifies style information for the object. Data type: String
External Title	It is mapped to the HTML attribute *title*. This attribute offers advisory information about the object. Data type: String
Language	It is mapped to the HTML attribute *lang*. This attribute specifies the base language of the object's attribute values and text content. Data type: String

Parameter form control

The properties of a parameter form control in a page report are:

Property Name	Description
General	
Class Type	Indicates the class type of the object. This property is read only.
Instance Name	Shows the instance name of the object. This property is read only.
Geometry	
Height	Specifies the height of the object, in inches or centimeters. Enter a numeric value to change the height. Data type: Float
Width	Specifies the width of the object, in inches or centimeters. Enter a numeric value to change the width. Data type: Float
X	Specifies the horizontal coordinate of the top left corner of the object, relative to its parent container. This property is ignored if the Position property is set to static. Enter a numeric value to change the position in inches or centimeters. Data type: Float
Y	Specifies the vertical coordinate of the top left corner of the object, relative to its parent container. This property is ignored if the Position property is set to static. Enter a numeric value to change the position in inches or centimeters. Data type: Float
Color	
Background	Specifies the background color of the object. Choose a color from the drop-down list, enter a hexadecimal RGB value (for example, 0xff0000), choose/create a formula or edit an expression that returns a color, or select Custom from the drop-down list to specify the color. Data type: String
CSS	
Class	Specifies a CSS class to be applied to the object which is a valid class in the CSS file. Data type: String
ID	Specifies the identifier of the object, which must be unique in the report. The ID property can be a style sheet selector. Data type: String
Style	The property can be used in two ways. • Specifies a style to be applied to the object. Choose a style from the drop-down list which is available when a Style Group has been selected at the report tab level and when there are styles in the Style Group that can be applied to the object, or type in the style name. • Specifies a CSS selector to be applied to the object. Enter a valid class name from the CSS file. Data type: String
Others	
Export to Applet	Specifies whether to include the object when exporting the report to Applet. Data type: Boolean
Export to CSV	Specifies whether to include the object when exporting the report to CSV. Data type: Boolean
Export to Excel	Specifies whether to include the object when exporting the report to Excel. Data type: Boolean
Export to HTML	Specifies whether to include the object when exporting the report to HTML. Data type: Boolean
Export to PDF	Specifies whether to include the object when exporting the report to PDF. Data type: Boolean
Export to PostScript	Specifies whether to include the object when exporting the report to PostScript. Data type: Boolean
Export to Report Result	Specifies whether to include the object when exporting the report to Report Result. Data type: Boolean
Export to RTF	Specifies whether to include the object when exporting the report to RTF. Data type: Boolean
Export to Text	Specifies whether to include the object when exporting the report to Text. Data type: Boolean
Export to XML	Specifies whether to include the object when exporting the report to XML. Data type: Boolean
Invisible	Specifies whether to show the object in the design area and in the report results. All formulas and calculations will still be performed if the property is set to true. Data type: Boolean
Position	Specifies the position of the object. Choose an option from the drop-down list. • **static** - The object is positioned at the location in which it is inserted. If selected, the X, Y and other position-related properties will be hidden or disabled. The position of the object is affected by preceding objects in the same container. • **relative** - The object is positioned at an offset to the position at which it is inserted. The offset is determined by the X and Y property values. • **absolute** - The object will be located at the position specified by dragging and dropping or by setting its X and Y property values. The position of the object is not affected by preceding objects in the same container. Data type: Enumeration
Title	
Background	Specifies the background color of the title bar. Choose a color from the drop-down list, enter a hexadecimal RGB value (for example, 0xff0000), choose/create a formula or edit an expression that returns a color, or select Custom from the drop-down list to specify the color. Data type: String
Bold	Specifies whether to make the text bold in the title bar. Data type: Boolean
Font Face	Specifies the font of the text in the title bar. Choose an option from the drop-down list. Data type: Enumeration
Font Size	Specifies the font size of the text in the title bar. Enter an integer value to change the size. Data type: Integer
Foreground	Specifies the foreground color of the title bar. Choose a color from the drop-down list, enter a hexadecimal RGB value (for example, 0xff0000), choose/create a formula or edit an expression that returns a color, or select Custom from the drop-down list to specify the color. Data type: String
Horizontal Alignment	Specifies the horizontal justification of the text in the title bar. Choose an option from the drop-down list. Data type: Enumeration
Italic	Specifies whether to make the text italic in the title bar. Data type: Boolean
Show Title	Specifies whether to show the title bar. It is meaningful to set all the other properties in the Title category when this property is set to true. Data type: Boolean
Text	Specifies the text displayed in the title bar. Enter a string to change the text. Data type: String
Underline	Specifies whether to underline the text in the title bar. Data type: Boolean
Excel	
Column Index	Specifies the X coordinate of the object relative to its parent container when exported to Excel or CSV, measured in cells. The Columned property at the report tab level must be set to true for this property to take effect. Data type: Integer
Row Index	Specifies the Y coordinate of the object relative to its parent container when exported to Excel or CSV, measured in cells. The Columned property at the report tab level must be set to true for this property to take effect. Data type: Integer
Border	
Border Color	Specifies the color of the border of the object. Choose a color from the drop-down list, enter a hexadecimal RGB value (for example, 0xCCAA00), choose/create a formula or edit an expression that returns a color, or select Custom from the drop-down list to specify the color. Data type: String
Border Thickness	Specifies the width of the border in inches or centimeters. Enter a numeric value to change the thickness. Data type: Float
Bottom Line	Specifies the line style of the bottom border of the object. Choose a style from the drop-down list. Data type: Enumeration
Left Line	Specifies the line style of the left border of the object. Choose a style from the drop-down list. Data type: Enumeration
Right Line	Specifies the line style of the right border of the object. Choose a style from the drop-down list. Data type: Enumeration
Top Line	Specifies the line style of the top border of the object. Choose a style from the drop-down list. Data type: Enumeration
Pattern	
Pattern Color	Specifies the color in which to draw a pattern to fill the object. Choose a color from the drop-down list, enter a hexadecimal RGB value (for example, 0xff0000), choose/create a formula or edit an expression that returns a color, or select Custom from the drop-down list to specify the color and fill effect. Data type: String
Pattern Style	Specifies the style of the pattern. Choose an option from the drop-down list. • **none** - No pattern will be applied to the object. • **50%** - The object will be filled in the Pattern Color of 50%-transparency. • **horizontal** - The pattern will be made of horizontal lines of the Pattern Color. • **vertical** - The pattern will be made of vertical lines of the Pattern Color. • **grid** - The pattern will be an overlapping of horizontal and vertical lines of the Pattern Color. • **diagonal** - The pattern will be made of diagonal lines of the Pattern Color. Data type: Enumeration
Accessibility	
External CSS Class Selector	Specifies a class selector to be applied to the object when exported as HTML. Enter a valid class name from the CSS file. Data type: String
External Style	It is mapped to the HTML attribute *style*. This attribute specifies style information for the object. Data type: String
Language	It is mapped to the HTML attribute *lang*. This attribute specifies the base language of the object's attribute values and text content. Data type: String

Properties for web report

The following are object properties for web report included in the Report Inspector, which are listed in alphabetical order:

- Chart
- Crosstab
- DBField
- Formula field
- Geographic map
- Image
- Label
- Multimedia object (OLE object)
- Page panel
- Parameter field
- Report body
- Special field
- Table
- Tabular
- TOC
- Web control
- Web report

Chart

A chart mainly consists of three parts: the platform, paper, and legend. The chart platform is a container for other chart elements, and the chart paper holds walls, axes, data markers, axis values, and category names.

The following topics are detailed introduction to properties of the elements:

- Chart platform
- Chart paper
- Chart legend
- Chart label

Chart platform

A chart platform is a container for other chart elements. You can use the properties listed in the following table to control the appearances or behavior of a chart platform. In the Report Inspector, a chart platform is displayed as ChartObject by default.

The properties of a chart platform are:

Property Name	Description
General	
Data Inherit	Indicates whether the dataset for this object is inherited from another object. This property is read only.
Geometry	
Height	Specifies the height of the object, in inches or centimeters. Enter a numeric value to change the height. Data type: Float
Width	Specifies the width of the object, in inches or centimeters. Enter a numeric value to change the width. Data type: Float
CSS	
Class	Specifies a CSS class to be applied to the object which is a valid class in the CSS file. Data type: String
Style	The property can be used in two ways. • Specifies a style to be applied to the object. Choose a style from the drop-down list which is available when a Style Group has been selected at the report tab level and when there are styles in the Style Group that can be applied to the object, or type in the style name. • Specifies a CSS selector to be applied to the object. Enter a valid class name from the CSS file. Data type: String
Others	
Alternative Text	Specifies the alternate text of the image, which will be shown when the image cannot be displayed. This property applies only when the chart is exported to HTML as an image chart. Enter a string to serve as the alternative text. Data type: String
Default for Filter	Specifies whether to show the data component as the default component in the Apply to drop-down list of the Filter dialog in Web Report Studio. Data type: Boolean **Note:** In a web report, only one data component's Default for Filter property can be set to true.
Export to Applet	Specifies whether to include the object when exporting the report to Applet. Data type: Boolean
Export to Excel	Specifies whether to include the object when exporting the report to Excel. Data type: Boolean
Export to HTML	Specifies whether to include the object when exporting the report to HTML. Data type: Boolean
Export to PDF	Specifies whether to include the object when exporting the report to PDF. Data type: Boolean
Export to PostScript	Specifies whether to include the object when exporting the report to PostScript. Data type: Boolean
Export to Report Result	Specifies whether to include the object when exporting the report to Report Result. Data type: Boolean
Export to RTF	Specifies whether to include the object when exporting the report to RTF. Data type: Boolean
Invisible	Specifies whether to show the object in the design area and in the report results. All formulas and calculations will still be performed if the property is set to true. Data type: Boolean
Invisible for Filter Dialogs	Specifies whether to show the data component in the Apply to drop-down list of the Filter dialog in Web Report Studio. This property cannot be edited when Default for Filter is set to true. Data type: Boolean
Logic Column	Specifies whether to show the object in the next visible table cell in the same row when the column which holds the object is hidden. Choose an option from the drop-down list. Data type: Enumeration **Notes:** • The next visible table cell should be completely empty, that is, the cell should not hold any content including blank space. • When the property for several objects in the same row is set to next visible column, and the columns holding these objects are all hidden, then only the object in the rightmost column will be shown in the next visible cell.
Suppress	Specifies whether to show the object in the design area and in the report results. All formulas and calculations will be skipped if the property is set to true. Data type: Boolean **Note:** When both the Invisible and Suppress properties of an object are set to true, Suppress has the higher priority.
Suppress When No Records	Specifies whether to display the object in the report results when no record is returned to its parent data container. Data type: Boolean
Excel	
Column Index	Specifies the X coordinate of the object relative to its parent container when exported to Excel or CSV, measured in cells. The Columned property at the report level must be set to true for this property to take effect. Data type: Integer
Column Number	Specifies the number of columns which will be the object's width in the exported Excel or CSV file. The Columned property at the report level must be set to true for this property to take effect. Data type: String
Row Index	Specifies the Y coordinate of the object relative to its parent container when exported to Excel or CSV, measured in cells. The Columned property at the report level must be set to true for this property to take effect. Data type: Integer
Row Number	Specifies the number of rows which will be the object's height in the exported Excel or CSV file. The Columned property at the report level must be set to true for this property to take effect. Data type: String
TOC	
Anchor Display Value	Specifies a string or formula to display for the TOC entry for this object in the TOC Browser. By default it is the value of the specified object. The TOC Anchor property must be set to true for this property to take effect. Data type: String
TOC Anchor	Specifies whether to include the object in the TOC Browser for the report. Data type: Boolean
Platform (unavailable to org chart)	
Anti-aliasing	Specifies whether to make the edges in a chart smooth. Data type: Boolean
Color When Null	Specifies the color when the value is null. This property is available to heat map only and is applied when the heat map is colored only by a summary. Choose a color from the drop-down list, enter a hexadecimal RGB value (for example, 0xff0000), or select Custom from the drop-down list to specify the color. Data Type: String
Color When Zero	Specifies the color mapping to the zero value. Available to heat map only. It works when the heat map is colored only by a summary and the Using Color When Zero property is set to true. Choose a color from the drop-down list, enter a hexadecimal RGB value (for example, 0xff0000), or select Custom from the drop-down list to specify the color. Data Type: String
End Color	Specifies the end color of the gradient colors. This property is available to heat map only and is applied when the heat map is colored only by a summary. Choose a color from the drop-down list, enter a hexadecimal RGB value (for example, 0xff0000), or select Custom from the drop-down list to specify the color. Data type: String
Minimum Tick Mark Space	Specifies the minimum space between two adjacent tick marks. Enter a numeric value to change the space. Data type: Float
Pattern List	Specifies patterns of the data markers. Click ... in the value cell and click the small squares in the color tray one by one to specify the patterns for the data markers. The patterns can be one or more of the following: Automatic, Color, Texture, and Gradient. For heat maps, the Pattern List property works only when using single color and using groups as the color-by fields. Data type: String
Reverse	Specifies whether to reverse the start color and the end color. This property is available to heat map only and is applied when the heat map is colored only by a summary. Data type: Boolean
Show Legend	Specifies whether to make the legend visible in a chart. Data type: Boolean
Start Color	Specifies the start color of the gradient colors. This property is available to heat map only and is applied when the heat map is colored only by a summary. Choose a color from the drop-down list, enter a hexadecimal RGB value (for example, 0xff0000), or select Custom from the drop-down list to specify the color. Data type: String
Using Color When Zero	Specifies whether to map the zero value to the color specified by the property Color When Zero. • If true and the minimum value < 0 < the maximum value, the colors will be from the start color to the color when zero first, and then from the color when zero to the end color. • If false, the colors will be from the start color to the end color directly. This property is available to heat map only and is applied when the heat map is colored only by a summary. Data type: Boolean
Background	
Border Color	Specifies the color of the platform border. Choose a color from the drop-down list, enter a hexadecimal RGB value (for example, 0xCCAA00), or select Custom from the drop-down list to specify the color. Data type: String
Border End Caps	Specifies the ending style of the platform border line. Choose an option from the drop-down list. Data type: Enumeration
Border Joint	Specifies the joint style of the platform border line. Choose an option from the drop-down list. Data type: Enumeration
Border Outlined	Specifies whether to show the platform border line in Outline Path. If the property is set to true, the platform border line will be shown in Outline Path; otherwise, in Fill Path. Data type: Boolean
Border Style	Specifies the line style of the platform border. Choose an option from the drop-down list. Data type: Enumeration
Border Thickness	Specifies the width of the platform border, in inches or centimeters. Enter a numeric value to change the thickness. Data type: Float
Border Transparency	Specifies the transparency of the platform border, in percent. Enter a numeric value to change the transparency. Data type: Integer
Border Type	Specifies the border type of the platform. Choose an option from the drop-down list. Data type: Enumeration
Border Variable Dash	Specifies whether to resize the dash automatically. • **true** - If selected, the dash size will be adjusted automatically, and the option Auto Adjust Dash in the Format Platform dialog will take effect. • **false** - If selected, the dash size will be of fixed size, and the option Fixed Dash Size in the Format Platform dialog will take effect. Data type: Boolean
Color	Specifies the color of the platform background, applied when the property Fill Type is set to color. Choose a color from the drop-down list, enter a hexadecimal RGB value (for example, 0xff0000), or select Custom from the drop-down list to specify the color and fill effect. Data type: String
Fill Transparency	Specifies the transparency of the platform background, in percent. Enter a numeric value to change the transparency. Data type: Integer
Fill Type	Specifies the fill pattern for the platform background. Choose an option from the drop-down list. Data type: Enumeration
Gradient End Color	Specifies the color of the point where the gradient ends, applied when the property Fill Type is set to gradient. Choose a color from the drop-down list, enter a hexadecimal RGB value (for example, 0xff0000), or select Custom from the drop-down list to specify the color and fill effect. Data type: String
Gradient End X	Specifies the horizontal position where the gradient ends, measured in a percentage of the platform width. Applied when the property Fill Type is set to gradient. Data type: Integer
Gradient End Y	Specifies the vertical position where the gradient ends, measured in a percentage of the platform height. Applied when the property Fill Type is set to gradient. Data type: Integer
Gradient Start Color	Specifies the color of the point where the gradient begins, applied when the property Fill Type is set to gradient. Choose a color from the drop-down list, enter a hexadecimal RGB value (for example, 0xff0000), or select Custom from the drop-down list to specify the color and fill effect. Data type: String
Gradient Start X	Specifies the horizontal position where the gradient begins, measured in a percentage of the platform width. Applied when the property Fill Type is set to gradient. Data type: Integer
Gradient Start Y	Specifies the vertical position where the gradient begins, measured in a percentage of the platform height. Applied when the property Fill Type is set to gradient. Data type: Integer
Gradient Style	Specifies the gradient style for the platform background, applied when the property Fill Type is set to gradient. Choose an option from the drop-down list. Data type: Enumeration
Image File	Specifies the file name of the image, a portion of which will be displayed as the platform background. Applied when Fill Type is set to image. Enter the file name with suffix in the value cell. If the image is outside of the current catalog, you should include the full path of the image correctly. Data type: String
Image Height	The four properties, Image X, Image Y, Image Width, and Image Height, together are used to define a portion of the image (the portion is a rectangle), which will be displayed as the platform background. They take effect when the property Fill Type is set to image. Image Height specifies the height of the image portion, measured in a percentage of the image height. Data type: Integer
Image Layout	Specifies the layout style for the image that will be displayed as the platform background, applied when the property Fill Type is set to image. Choose an option from the drop-down list. Data type: Enumeration
Image Width	The four properties, Image X, Image Y, Image Width, and Image Height, together are used to define a portion of the image (the portion is a rectangle), which will be displayed as the platform background. They take effect when the property Fill Type is set to image. Image Width specifies the width of the image portion, measured in a percentage of the image width. Data type: Integer
Image X	The four properties, Image X, Image Y, Image Width, and Image Height, together are used to define a portion of the image (the portion is a rectangle), which will be displayed as the platform background. They take effect when the property Fill Type is set to image. Image X specifies the distance between the left border of image and the portion that will contain the pattern, measured in a percentage of the image width. Data type: Integer
Image Y	The four properties, Image X, Image Y, Image Width, and Image Height, together are used to define a portion of the image (the portion is a rectangle), which will be displayed as the platform background. They take effect when the property Fill Type is set to image. Image Y specifies the distance between the top border of image and the portion that will contain the pattern, measured in a percentage of the image height. Data type: Integer
Inset Bottom	Specifies the bottom position of the background area, measured in a percentage of the chart height, from the bottom edge of the chart. Data type: Float
Inset Left	Specifies the left position of the background area, measured in a percentage of the chart width, from the left edge of the chart. Data type: Float
Inset Right	Specifies the right position of the background area, measured in a percentage of the chart width, from the right edge of the chart. Data type: Float
Inset Top	Specifies the top position of the background area, measured in a percentage of the chart height, from the top edge of the chart. Data type: Float
Radius	Specifies the radius for the joint of the platform border line. The property takes effect only when Border Joint is set to joint round. Enter a numeric value to change the radius. Data type: Float
Texture Background Color	Specifies the background color of the texture, applied when the property Fill Type is set to texture. Choose a color from the drop-down list, enter a hexadecimal RGB value (for example, 0xff0000), or select Custom from the drop-down list to specify the color. Data type: String
Texture Foreground Color	Specifies the foreground color of the texture, applied when the property Fill Type is set to texture. Choose a color from the drop-down list, enter a hexadecimal RGB value (for example, 0xff0000), or select Custom from the drop-down list to specify the color. Data type: String
Texture Style	Specifies the texture style of the platform background, applied when the property Fill Type is set to texture. Choose an option from the drop-down list. Data type: Enumeration
Data (unavailable to org chart)	
For heat map	
Color by Logarithm	Specifies whether to calculate the color-by summary based on logarithm function. Available to heat map only. Choose an option from the drop-down list: • **None** Returns the real value of the field instead of any logarithmic value. • **log x** Gets the logarithm of the field with base 10. • **ln x** Gets the logarithm of the field with base e. Data type: String
Size by Logarithm	Specifies whether to calculate the size-by summary based on logarithm function. Available to heat map only. Choose an option from the drop-down list: • **None** Returns the real value of the field instead of any logarithmic value. • **log x** Gets the logarithm of the field with base-10. • **ln x** Gets the logarithm of the field with base-e. Data type: String
For other chart types	
Category Data Mapping File	Specifies the data mapping file to the object for NLS use, which takes effect for category values shown on the category (X) axis and the hints of the data markers. For example, if the data mapping file name is Category_de_DE.properties, set the value as Category (without the language and locale part). Data type: String
Category End Offset	The four properties work together to control the range of the data being displayed on a chart: Category Start Offset, Category End Offset, Series Start Offset, and Series End Offset. Category End Offset specifies the ending offset of the categories. It does not apply to the charts which involve only one group. Data type: Integer
Category Format	Specifies the data format for the category axis (the X axis) to display the tick mark labels in the way you choose. Click ... in the value cell to specify the format in the Data Format dialog. Data type: String
Category Start Offset	The four properties work together to control the range of the data being displayed on a chart: Category Start Offset, Category End Offset, Series Start Offset, and Series End Offset. Category Start Offset specifies the starting offset of the categories. It does not apply to the charts which involve only one group. Data type: Integer
Category Value Encoding	Specifies the encoding format for values on the category axis. Choose an option from the drop-down list. Data type: String
Hint Percent Format	Specifies the format for each hint percentage to the total. Click ... in the value cell to specify the format in the Data Format dialog. Data type: String
Hint Value Format	Specifies the number format for the hint message. Click ... in the value cell to specify the format in the Data Format dialog. Data type: String
Hyperlink	Specifies whether to add a hyperlink that refers to another report or a website to the data markers of the chart. Data type: String
Hyperlink Target	Specifies the target window or frame in which to display the content the Hyperlink property specifies. Choose an option from the drop-down list. • **<Server Setting>** Loads the linked file according to setting of the Target of Detail Table Report and Links option in the Customize Profile panel of the Profile dialog in JReport Server. • **New Window** Loads the linked file into a new window. The window is not named. • **Whole Window** Loads the linked file into the full browser window. • **Same Frame** Loads the linked file into the same frame as the link. • **Parent Frame** Loads the linked file into the parent frame of the frame that contains the link. • **Other Frame** Loads the linked file into some other specified frame. If selected, you can also directly input the name of the frame you have defined in the text box. If the frame name does not exist, the linked file will be loaded into a new window. Data type: String
Motion Bar Format	Specifies the data format for the motion field values displayed on the motion bar. Click ... in the value cell to specify the format in the Data Format dialog. Data type: String
Overall Series	Specifies whether to calculate the top or bottom N category values based on the series values. Data type: Boolean
Primary Data Format	Specifies the data format for the primary value axis (the Y1 axis) to display the tick mark labels in the way you choose. Click ... in the value cell to specify the format in the Data Format dialog. Data type: String
Reverse Category	Specifies whether to reverse the sequence of the category field value. Data type: Boolean
Reverse Series	Specifies whether to reverse the sequence of the series field value. Data type: Boolean
Secondary Data Format	Specifies the data format for the secondary value axis (the Y2 axis) to display the tick mark labels in the way you choose. Click ... in the value cell to specify the format in the Data Format dialog. Data type: String
Series Data Mapping File	Specifies the data mapping file to the object for NLS use, which takes effect for series values shown in the legend and the hints of the data markers. Not available to bubble chart. For example, if the data mapping file name is Category_de_DE.properties, set the value as Category (without the language and locale part). Data type: String
Series End Offset	The four properties work together to control the range of the data being displayed on a chart: Category Start Offset, Category End Offset, Series Start Offset, and Series End Offset. Series End Offset specifies the ending offset of the series. Data type: Integer
Series Format	Specifies the data format for the series axis (the Z axis) to display the tick mark labels in the way you choose. Click ... in the value cell to specify the format in the Data Format dialog. Data type: String
Series Start Offset	The four properties work together to control the range of the data being displayed on a chart: Category Start Offset, Category End Offset, Series Start Offset, and Series End Offset. Series Start Offset specifies the starting offset of the series. Data type: Integer
Series Value Encoding	Specifies the encoding format for values on the series axis. Choose an option from the drop-down list. Data type: String
Sort Category	Specifies the sorting order for the category field values. Choose an option from the drop-down list. Data type: Enumeration
Sort Series	Specifies the sorting order for the series field values. Choose an option from the drop-down list. Data type: Enumeration
Swap Groups	Specifies to display values from different data fields by switching data between the category and series axes, the category and value axes of a chart. By default, this property is set to false, which means no switch will take place in the chart. • **Switching between the category and series axes** If you create a chart by adding fields to all of its axes, and set Swap Groups to true, data on the category and series axes will be switched. • **Switching between the category and value axes** If you create a chart by adding one field to the category axis and several to the value axis, and set Swap Groups to true, data on the category and value axes will be switched. Data type: Boolean **Note:** If the chart has only one field on the category axis and one on the value axis, even though the property value is set to true, there will be no switch take place on the two axes.
X HyperLink	Specifies whether to add a hyperlink that refers to another report or a website to the X axis labels. Data type: String
X Hyperlink Target	Specifies the target window or frame in which to display the content the X Hyperlink property specifies. Choose an option from the drop-down list. • **<Server Setting>** Loads the linked file according to setting of the Target of Detail Table Report and Links option in the Customize Profile panel of the Profile dialog in JReport Server. • **New Window** Loads the linked file into a new window. The window is not named. • **Whole Window** Loads the linked file into the full browser window. • **Same Frame** Loads the linked file into the same frame as the link. • **Parent Frame** Loads the linked file into the parent frame of the frame that contains the link. • **Other Frame** Loads the linked file into some other specified frame. If selected, you can also directly input the name of the frame you have defined in the text box. If the frame name does not exist, the linked file will be loaded into a new window. Data type: String
Z HyperLink	Specifies whether to add a hyperlink that refers to another report or a website to the Z axis labels. Data type: String
Z Hyperlink Target	Specifies the target window or frame in which to display the content the Z Hyperlink property specifies. Choose an option from the drop-down list. • **<Server Setting>** Loads the linked file according to setting of the Target of Detail Table Report and Links option in the Customize Profile panel of the Profile dialog in JReport Server. • **New Window** Loads the linked file into a new window. The window is not named. • **Whole Window** Loads the linked file into the full browser window. • **Same Frame** Loads the linked file into the same frame as the link. • **Parent Frame** Loads the linked file into the parent frame of the frame that contains the link. • **Other Frame** Loads the linked file into some other specified frame. If selected, you can also directly input the name of the frame you have defined in the text box. If the frame name does not exist, the linked file will be loaded into a new window. Data type: String

Chart paper

The properties of a chart paper are:

Property Name	Description
Geometry	
Height	Specifies the height of the object, in inches or centimeters. Enter a numeric value to change the height. Data type: Float
Width	Specifies the width of the object, in inches or centimeters. Enter a numeric value to change the width. Data type: Float
CSS	
Class	Specifies a CSS class to be applied to the object which is a valid class in the CSS file. Data type: String
Style	The property can be used in two ways. • Specifies a style to be applied to the object. Choose a style from the drop-down list which is available when a Style Group has been selected at the report tab level and when there are styles in the Style Group that can be applied to the object, or type in the style name. • Specifies a CSS selector to be applied to the object. Enter a valid class name from the CSS file. Data type: String
Coordinate Paper (unavailable to heat map and org charts)	
Graph Position	Specifies the position of the paper contents relative to the paper background. Choose an option from the drop-down list. Data type: Enumeration
Reverse Painting Order	Specifies whether to reverse the order of graphic objects in a combo chart. Data type: Boolean
Scale X	Specifies the percentage for elongating or shortening the X axis. Enter a numeric value to change the percentage. Data type: Float
Scale Y	Specifies the percentage for elongating or shortening the Y axis. Enter a numeric value to change the percentage. Data type: Float
Shadow Contents	Specifies whether to add shadows to all 2-D charts. Data type: Boolean
Background	
Border Color	Specifies the color of the paper border. Choose a color from the drop-down list, enter a hexadecimal RGB value (for example, 0xff0000), or select Custom from the drop-down list to specify the color. Data type: String
Border End Caps	Specifies the ending style of the paper border line. Choose an option from the drop-down list. Data type: Enumeration
Border Joint	Specifies the joint style of the paper border line. Choose an option from the drop-down list. Data type: Enumeration
Border Outlined	Specifies whether to show the paper border line in Outline Path. If the property is set to true, the paper border line will be shown in Outline Path; otherwise, in Fill Path. Data type: Boolean
Border Style	Specifies the line style of the paper border. Choose an option from the drop-down list. Data type: Enumeration
Border Thickness	Specifies the width of the paper border, in inches or centimeters. Enter a numeric value to change the thickness. Data type: Float
Border Transparency	Specifies the transparency of the paper border, in percent. Enter a numeric value to change the transparency. Data type: Integer
Border Type	Specifies the border type of the paper. Choose an option from the drop-down list. Data type: Enumeration
Border Variable Dash	Specifies whether to resize the dash automatically. • **true** - If selected, the dash size will be adjusted automatically, and the option Auto Adjust Dash in the Format Paper dialog will take effect. • **false** - If selected, the dash size will be of fixed size, and the option Fixed Dash Size in the Format Paper dialog will take effect. Data type: Boolean
Color	Specifies the color of the paper background, applied when the property Fill Type is set to color. Choose a color from the drop-down list, enter a hexadecimal RGB value (for example, 0xff0000), or select Custom from the drop-down list to specify the color. Data type: String
Fill Transparency	Specifies the transparency of the paper background, in percent. Enter a numeric value to change the transparency. Data type: Integer
Fill Type	Specifies the fill pattern for the paper background. Choose an option from the drop-down list. Data type: Enumeration
Gradient End Color	Specifies the color of the point where the gradient ends, applied when the property Fill Type is set to gradient. Choose a color from the drop-down list, enter a hexadecimal RGB value (for example, 0xff0000), or select Custom from the drop-down list to specify the color. Data type: String
Gradient End X	Specifies the horizontal position where the gradient ends, measured in a percentage of the paper width. Applied when the property Fill Type is set to gradient. Data type: Integer
Gradient End Y	Specifies the vertical position where the gradient ends, measured in a percentage of the paper height. Applied when the property Fill Type is set to gradient. Data type: Integer
Gradient Start Color	Specifies the color of the point where the gradient begins, applied when the property Fill Type is set to gradient. Choose a color from the drop-down list, enter a hexadecimal RGB value (for example, 0xff0000), or select Custom from the drop-down list to specify the color. Data type: String
Gradient Start X	Specifies the horizontal position where the gradient begins, measured in a percentage of the paper width. Applied when the property Fill Type is set to gradient. Data type: Integer
Gradient Start Y	Specifies the vertical position where the gradient begins, measured in a percentage of the paper height. Applied when the property Fill Type is set to gradient. Data type: Integer
Gradient Style	Specifies the gradient style for the paper background, applied when the property Fill Type is set to gradient. Choose an option from the drop-down list. Data type: Enumeration
Image File	Specifies the file name of the image, a portion of which will be displayed as the paper background. Applied when the property Fill Type is set to image. Enter the file name with suffix in the value cell. If the image is outside of the current catalog, you should include the full path of the image correctly. Data type: String
Image Height	The four properties, Image X, Image Y, Image Width, and Image Height, together are used to define a portion of the image (the portion is a rectangle), which will be displayed as the paper background. They take effect when the property Fill Type is set to image. Image Height specifies the height of the image portion, measured in a percentage of the image height. Data type: Integer
Image Layout	Specifies the layout style for the image that will be displayed as the paper background, applied when the property Fill Type is set to image. Choose an option from the drop-down list. Data type: Enumeration
Image Width	The four properties, Image X, Image Y, Image Width, and Image Height, together are used to define a portion of the image (the portion is a rectangle), which will be displayed as the paper background. They take effect when the property Fill Type is set to image. Image Width specifies the width of the image portion, measured in a percentage of the image width. Data type: Integer
Image X	The four properties, Image X, Image Y, Image Width, and Image Height, together are used to define a portion of the image (the portion is a rectangle), which will be displayed as the paper background. They take effect when the property Fill Type is set to image. Image X specifies the distance between the left border of image and the portion that will contain the pattern, measured in a percentage of the image width. Data type: Integer
Image Y	The four properties, Image X, Image Y, Image Width, and Image Height, together are used to define a portion of the image (the portion is a rectangle), which will be displayed as the paper background. They take effect when the property Fill Type is set to image. Image Y specifies the distance between the top border of image and the portion that will contain the pattern, measured in a percentage of the image height. Data type: Integer
Inset Bottom	Specifies the bottom position of the background area, measured in a percentage of the paper height, from the bottom edge of the paper. Data type: Float
Inset Left	Specifies the left position of the background area, measured in a percentage of the paper width, from the left edge of the paper. Data type: Float
Inset Right	Specifies the right position of the background area, measured in a percentage of the paper width, from the right edge of the paper. Data type: Float
Inset Top	Specifies the top position of the background area, measured in a percentage of the paper height, from the top edge of the paper. Data type: Float
Texture Background Color	Specifies the background color of the texture, applied when the property Fill Type is set to texture. Choose a color from the drop-down list, enter a hexadecimal RGB value (for example, 0xff0000), or select Custom from the drop-down list to specify the color. Data type: String
Texture Foreground Color	Specifies the foreground color of the texture, applied when the property Fill Type is set to texture. Choose a color from the drop-down list, enter a hexadecimal RGB value (for example, 0xff0000), or select Custom from the drop-down list to specify the color. Data type: String
Texture Style	Specifies the texture style of the paper background, applied when the property Fill Type is set to texture. Choose an option from the drop-down list. Data type: Enumeration
Others (available to org charts only)	
Layout Mode	Specifies how the org chart tree expands from the root node. Choose an option from the drop-down list. • **top down** The org chart tree expands from top to bottom. • **bottom up** The org chart tree expands from bottom to top. It is the reverse of the top down mode. • **left right** The org chart tree expands from left to right. Data type: Enumeration
Graph (unavailable to heat map and org charts)	
Border	Specifies whether to show the borders. Data type: Boolean
Border Color	Specifies the line color for the border, not applying to 2-D line charts. Choose a color from the drop-down list, enter a hexadecimal RGB value (for example, 0xff0000), or select Custom from the drop-down list to specify the color. Data type: String
Border End Caps	Specifies the ending style for the border, not applying to 2-D line charts. Choose an option from the drop-down list. Data type: Enumeration
Border Joint	Specifies the line joint style for the border, not applying to 2-D line charts. Choose an option from the drop-down list. Data type: Enumeration
Border Outlined	Specifies whether to show the border in outline form, not applying to 2-D line charts. If the property is set to true, the border will be shown in outline form; otherwise, in whole form. Data type: Boolean
Border Style	Specifies the line style for the border, not applying to 2-D line charts. Choose an option from the drop-down list. Data type: Enumeration
Border Thickness	Specifies the thickness for the border, not applying to 2-D line charts. Enter a numeric value to change the thickness. Data type: Float
Border Transparency	Specifies the line transparency for the border, in percent, not applying to 2-D line charts. Enter a numeric value to change the transparency. Data type: Integer
Border Variable Dash	Specifies whether to resize the dash automatically, not applying to 2-D line charts. If the property is set to true, the dash size will be adjusted automatically; otherwise, the dash size will be of fixed size. Data type: Boolean
Suppress Label When 0	If true, the data label whose value is 0 will not be displayed. This property is available to these 2-D chart types: bar, bench, line, area and pie. Data type: Boolean
For bar/bench chart	
Bar Gap	Specifies the distance between each bar/bench in a category. Applies to 2-D clustered bar/bench charts only. Enter a numeric value to change the gap. Data type: Integer
Bar Style	Specifies the style of the bars/benches. Choose an option from the drop-down list. It can be normal or cylinder, however, if you specify the style to be cylinder, it will take effect only when the property Use Depth is set to true. Data type: Enumeration
Bar Width	Specifies the total width of the bars/benches in each category, measured in a percentage of the unit width. Enter a numeric value to change the width. Data type: Integer
Depth	Specifies the depth of the bars/benches, applied when the value of the property Use Depth is set to true. Applies to 2-D bar/bench charts only. Enter a numeric value to change the depth. Data type: Float
Direction	Specifies the angle of the axis along the depth of the bars/benches. Applies to 2-D bar/bench charts only. Enter a numeric value to change the direction. Data type: Integer
Update Direction	Specifies the direction in which to paint the values when the chart is updated. Available to real time chart only. Choose an option from the drop-down list. • **Default** - If it is a bench chart, paints the values from top to bottom, otherwise, from right to left. • **Reverse** - If it is a bench chart, paints the values from bottom to top, otherwise, from left to right. Data type: Enumeration
Use Depth	Specifies whether to make the bars/benches visually three-dimensional. Applies to 2-D bar/bench charts only. Data type: Boolean **Note:** If the chart is a combo chart composed by bars and areas/lines, when you set the depth properties for the bars, they will be applied to the areas/lines as well, and vice versa.
Value Label Type	Specifies in which way the value labels will be displayed around the bars/benches. Applied only when Show Static Data Label is checked in the Format Bar dialog. Choose an option from the drop-down list. • **value** - Shows the value for each bar/bench. • **percent** - Shows the percentage of each bar/bench to the total. • **value and percent** - Shows the value and the percentage for each bar/bench. Data type: Enumeration
Vary Colors by Color List	Specifies whether or not to make the bar colors vary. It only takes effect on clustered bar/bench types and when the charts has no series field. Data type: Boolean
For line chart	
Area Pattern List	Specifies patterns for the areas which are formed by the chart axes and the lines. Applies to 2-D lines only. Click [...] in the value cell and click the small squares in the color tray one by one to specify the patterns for the areas. The patterns can be one or more of the following: Automatic, Color, Texture, and Gradient. Data type: String
Depth	Specifies the depth of the lines, applied when the property Use Depth is set to true. Enter a numeric value to change the depth. Data type: Float
Direction	Specifies the angle of the axis along the depth of the lines. Enter a numeric value to change the direction. Data type: Integer
Ignore Null Value	Specifies whether to skip null data values in order to draw continuous lines. • **true** - When a null data value appears, it is skipped and the line is drawn from the previous data value to the next data value directly. • **false** - When a null data value appears, draw it anyway and the line is broken at the point of the null data value. Data type: Boolean
Update Direction	Specifies the direction in which to paint the values when the chart is updated. Available to real time chart only. Choose an option from the drop-down list. • **Default** - Paints the values from right to left. • **Reverse** - Paints the values from left to right. Data type: Enumeration
Use Depth	Specifies whether to make the lines visually three-dimensional. Applies to 2-D line charts only. Data type: Boolean **Note:** If the chart is a combo chart composed by lines and areas/bars, when you set the depth properties for the lines, they will be applied to the areas/bars as well, and vice versa.
Value Label Type	Specifies in which way the value labels will be displayed around the lines. Applied only when Show Static Data Label is checked in the Format Line dialog. Available to 2-D line charts only. Choose an option from the drop-down list. • **value** - Shows the value for each line node. • **percent** - Shows the percentage of each line node to the total. • **value and percent** - Shows the value and the percentage for each line node. Data type: Enumeration
For area chart	
Depth	Specifies the depth of the areas, applied when the property Use Depth is set to true. Applies to 2-D area charts only. Enter a numeric value to change the depth. Data type: Float
Direction	Specifies the angle of the axis along the depth of the areas. Applies to 2-D area charts only. Enter a numeric value to change the direction. Data type: Integer
Ignore Null Value	Specifies whether to skip null data values in order to draw continuous areas. • **true** - When a null data value appears, it is skipped and the area is drawn from the previous data value to the next data value directly. • **false** - When a null data value appears, draw it anyway and the area is broken at the point of the null data value. Data type: Boolean
Update Direction	Specifies the direction in which to paint the values when the chart is updated. Available to real time chart only. Choose an option from the drop-down list. • **Default** - Paints the values from right to left. • **Reverse** - Paints the values from left to right. Data type: Enumeration
Use Depth	Specifies whether to make the areas visually three-dimensional. Applies to 2-D area charts only. Data type: Boolean **Note:** If the chart is a combo chart composed by areas and bars/lines, when you set the depth properties for the areas, they will be applied to the bars/lines as well, and vice versa.
Use Dropline	Specifies whether to show the lines that represent the data categories. Applies to 2-D area charts only. Data type: Boolean
Value Label Type	Specifies in which way the value labels will be displayed around the areas. Applied only when Show Static Data Label is checked in the Format Area dialog. Choose an option from the drop-down list. • **value** - Shows the value for each area node. • **percent** - Shows the percentage of each area node to the total. • **value and percent** - Shows the value and the percentage for each area node. Data type: Enumeration
For pie chart	
Angle X	Specifies the rotation angle of the chart around the X axis, in degrees. Enter a numeric value to change the rotation. Data type: Float
Angle Y	Specifies the rotation angle of the chart around the Y axis, in degrees. Enter a numeric value to change the rotation. Data type: Float
Donut Hole	Specifies the radius percentage of the donut hole to the total pie circle. Applies to donut chart only. Data type: Float
Pie Gap	Specifies the distance between every two adjacent pie sections. Enter a numeric value to change the gap. Data type: Float
Section Explode Gap	Specifies the distance between each pie section and the pie center. Enter a numeric value to change the gap. Data type: Float
Show Pie/Donut Name	Specifies whether to show the name of the pie/donut. Data type: Boolean
Value Label Type	Specifies in which way the value labels will be displayed around the pie. Applied only when Show Static Data Label is checked in the Format Pie dialog. Choose an option from the drop-down list. • **value** - Shows the value for the pie section. • **category name** - Shows the category name for the pie section. • **percent** - Shows the percentage of the section to the total. • **value and percent** - Shows the value and the percentage for the pie section. Data type: Enumeration
For indicator chart	
Horizontal Margin	Specifies the distance in inches or centimeters between the left indicators and the left border of the chart paper. Data type: Float
Range Radius	Specifies the relative size of an indicator in a percentage of the total indicator size. Data type: Integer
Vertical Margin	Specifies the distance in inches or centimeters between the top indicators and the top border of the chart paper. Data type: Float
For gauge chart	
Alert Color	Specifies the color for the alert range. Choose a color from the drop-down list, enter a hexadecimal RGB value (for example, 0xff0000), or select Custom from the drop-down list to specify the color. Data type: String
Alert Maximum Value	Specifies the maximum value for the alert range. Enter a numeric value to specify the value. Data type: Float
Alert Minimum Value	Specifies the minimum value for the alert range. Enter a numeric value to specify the value. Data type: Float
Arrow Style	Specifies the style for pointer arrows, applied when the property Pointer Style is set to Arrow. Applies to dial gauge charts only. Choose an option from the drop-down list. Data type: Enumeration
Curve Style	Specifies the style for the curve. Applies to dial gauge charts only. Choose an option from the drop-down list. Data type: Enumeration
Error Color	Specifies the color for the error range. Choose a color from the drop-down list, enter a hexadecimal RGB value (for example, 0xff0000), or select Custom from the drop-down list to specify the color. Data type: String
Error Maximum Value	Specifies the maximum value for the error range. Enter a numeric value to specify the value. Data type: Float
Error Minimum Value	Specifies the minimum value for the error range. Enter a numeric value to specify the value. Data type: Float
Fill Curve	Specifies whether to fill the dial with colors. Applies to dial gauge charts only. Data type: Boolean
Gauge Alert Gradient End Color	Specifies the color of the point where the gradient ends for the alert range, applied when Gauge Alert Type is set to gradient. Choose a color from the drop-down list, enter a hexadecimal RGB value (for example, 0xff0000), or select Custom from the drop-down list to specify the color. Data type: String
Gauge Alert Gradient End X	Specifies the horizontal position where the gradient ends, measured in a percentage of the alert range width. Applied when Gauge Alert Type is set to gradient. Data type: Integer
Gauge Alert Gradient End Y	Specifies the vertical position where the gradient ends, measured in a percentage of the alert range height. Applied when Gauge Alert Type is set to gradient. Data type: Integer
Gauge Alert Gradient Start Color	Specifies the color of the point where the gradient begins for the alert range, applied when Gauge Alert Type is set to gradient. Choose a color from the drop-down list, enter a hexadecimal RGB value (for example, 0xff0000), or select Custom from the drop-down list to specify the color. Data type: String
Gauge Alert Gradient Start X	Specifies the horizontal position where the gradient begins, measured in a percentage of the alert range width. Applied when Gauge Alert Type is set to gradient. Data type: Integer
Gauge Alert Gradient Start Y	Specifies the vertical position where the gradient begins, measured in a percentage of the alert range height. Applied when Gauge Alert Type is set to gradient. Data type: Integer
Gauge Alert Gradient Style	Specifies the gradient style for the alert range, applied when Gauge Alert Type is set to gradient. Choose an option from the drop-down list. Data type: Enumeration
Gauge Alert Transparency	Specifies the transparency of the alert range background, in percent. Enter a numeric value to change the transparency. Data type: Integer
Gauge Alert Type	Specifies the fill pattern for the alert range. Choose an option from the drop-down list. Data type: Enumeration
Gauge Bar Border Color	Specifies the color of the border of the object. Applies to bar gauge charts only. Choose a color from the drop-down list, enter a hexadecimal RGB value (for example, 0xff0000), or select Custom from the drop-down list to specify the color. Data type: String
Gauge Bar Border End Caps	Specifies the ending style of the scale border line. Applies to bar gauge charts only. Choose an option from the drop-down list. Data type: Enumeration
Gauge Bar Border Joint	Specifies the joint style of the scale border line. Applies to bar gauge charts only. Choose an option from the drop-down list. Data type: Enumeration
Gauge Bar Border Outlined	Specifies whether to show the scale border line in Outline Path. Applies to bar gauge charts only. If the property is set to true, the border line will be shown in Outline Path; otherwise, in Fill Path. Data type: Boolean
Gauge Bar Border Style	Specifies the line style of the scale borders. Applies to bar gauge charts only. Choose an option from the drop-down list. Data type: Enumeration
Gauge Bar Border Thickness	Specifies the width of the border in inches or centimeters. Applies to bar gauge charts only. Enter a numeric value to change the thickness. Data type: Float
Gauge Bar Border Transparency	Specifies the transparency of the borders, in percent. Applies to bar gauge charts only. Enter a numeric value to change the transparency. Data type: Integer
Gauge Bar Border Variable Dash	Specifies whether to resize the dash automatically. Applies to bar gauge charts only. If the property is set to true, the dash size will be adjusted automatically; otherwise, the dash size will be of fixed size. Data type: Boolean
Gauge Bar Layout	Specifies the layout for the scales. Applies to bar gauge charts only. Choose an option from the drop-down list. Data type: Enumeration
Gauge Error Gradient End Color	Specifies the color of the point where the gradient ends for the error range, applied when Gauge Error Type is set to gradient. Choose a color from the drop-down list, enter a hexadecimal RGB value (for example, 0xff0000), or select Custom from the drop-down list to specify the color. Data type: String
Gauge Error Gradient End X	Specifies the horizontal position where the gradient ends, measured in a percentage of the error range width. Applied when Gauge Error Type is set to gradient. Data type: Integer
Gauge Error Gradient End Y	Specifies the vertical position where the gradient ends, measured in a percentage of the error range height. Applied when Gauge Error Type is set to gradient. Data type: Integer
Gauge Error Gradient Start Color	Specifies the color of the point where the gradient begins for the error range, applied when Gauge Error Type is set to gradient. Choose a color from the drop-down list, enter a hexadecimal RGB value (for example, 0xff0000), or select Custom from the drop-down list to specify the color. Data type: String
Gauge Error Gradient Start X	Specifies the horizontal position where the gradient begins, measured in a percentage of the error range width. Applied when Gauge Error Type is set to gradient. Data type: Integer
Gauge Error Gradient Start Y	Specifies the vertical position where the gradient begins, measured in a percentage of the error range height. Applied when Gauge Error Type is set to gradient. Data type: Integer
Gauge Error Gradient Style	Specifies the gradient style for the error range, applied when Gauge Error Type is set to gradient. Choose an option from the drop-down list. Data type: Enumeration
Gauge Error Transparency	Specifies the transparency of the error range background, in percent. Enter a numeric value to change the transparency. Data type: Integer
Gauge Error Type	Specifies the fill pattern for the error range. Choose an option from the drop-down list. Data type: Enumeration
Gauge Normal Gradient End Color	Specifies the color of the point where the gradient ends for the normal range, applied when Gauge Normal Type is set to gradient. Choose a color from the drop-down list, enter a hexadecimal RGB value (for example, 0xff0000), or select Custom from the drop-down list to specify the color. Data type: String
Gauge Normal Gradient End X	Specifies the horizontal position where the gradient ends, measured in a percentage of the normal range width. Applied when Gauge Normal Type is set to gradient. Data type: Integer
Gauge Normal Gradient End Y	Specifies the vertical position where the gradient ends, measured in a percentage of the normal range height. Applied when Gauge Normal Type is set to gradient. Data type: Integer
Gauge Normal Gradient Start Color	Specifies the color of the point where the gradient begins for the normal range, applied when Gauge Normal Type is set to gradient. Choose a color from the drop-down list, enter a hexadecimal RGB value (for example, 0xff0000), or select Custom from the drop-down list to specify the color. Data type: String
Gauge Normal Gradient Start X	Specifies the horizontal position where the gradient begins, measured in a percentage of the normal range width. Applied when Gauge Normal Type is set to gradient. Data type: Integer
Gauge Normal Gradient Start Y	Specifies the vertical position where the gradient begins, measured in a percentage of the normal range height. Applied when Gauge Normal Type is set to gradient. Data type: Integer
Gauge Normal Gradient Style	Specifies the gradient style for the normal range, applied when Gauge Normal Type is set to gradient. Choose an option from the drop-down list. Data type: Enumeration
Gauge Normal Transparency	Specifies the transparency of the normal range background, in percent. Enter a numeric value to change the transparency. Data type: Integer
Gauge Normal Type	Specifies the fill pattern for the normal range. Choose an option from the drop-down list. Data type: Enumeration
Horizontal Margin	Specifies the distance in inches or centimeters between the left bubbles and the left border of the chart paper. Applies to bubble gauge charts only. Enter a numeric value to change the margin. Data type: Float
Normal Color	Specifies the color for the normal range. Choose a color from the drop-down list, enter a hexadecimal RGB value (for example, 0xff0000), or select Custom from the drop-down list to specify the color. Data type: String
Normal Maximum Value	Specifies the maximum value for the normal range. Enter a numeric value to specify the value. Data type: Float
Normal Minimum Value	Specifies the minimum value for the normal range. Enter a numeric value to specify the value. Data type: Float
Outer Inner Gap	Specifies the width of the dial. Applies to dial gauge charts only. Enter a numeric value to change the gap. Data type: Float
Pointer Style	Specifies the style for the pointer. Applies to dial gauge charts only. Choose an option from the drop-down list. Data type: Enumeration
Range Radius	Specifies the relative size of a dial or ball. Applies to dial and bubble gauge charts only. Enter a numeric value to change the size. Data type: Integer
Start Angle	Specifies the angle of the dial beginning point. Applies to dial gauge charts only. Enter a numeric value to change the angle. Data type: Integer
Stop Angle	Specifies the angle of the dial endpoint. Applies to dial gauge charts only. Enter a numeric value to change the angle. Data type: Integer
Vertical Margin	Specifies the distance in inches or centimeters between the top bubbles and the top border of the chart paper. Applies to bubble gauge charts only. Enter a numeric value to change the margin. Data type: Float
For bubble chart	
Use 3D Effect	Specifies whether to make the bubbles visually three-dimensional. Data type: Boolean
Cut Bubble Based on XY Area	Specifies whether to cut the bubbles when they are beyond the area formed by the X axis and Y axis. Data type: Boolean
For bullet chart	
Comparative Measure Color List	Specifies the fill effect of the comparative measures in the same data series. Click [...] in the value cell to specify the effect. Data type: String
Comparative Measure Width	Specifies the width of the comparative measures. Enter a numeric value to change the width. Data type: Integer
Featured Measure Width	Specifies the width of the featured measures. Enter a numeric value to change the width. Data type: Integer
Graph Direction	Specifies the direction to draw the bullet chart, horizontally or vertically. Choose an option from the drop-down list. Data type: Enumeration
Qualitative Ranges Color List	Specifies the fill effect of the qualitative ranges in the same data series. Click [...] in the value cell to specify the effect. Data type: String
Qualitative Ranges Width	Specifies the width of the qualitative ranges. Enter a numeric value to change the width. Data type: Integer
Special Featured Measure Graph Type	Specifies the graph type for the featured measures when the value (Y) axis does not start from the value zero. Choose an option from the drop-down list. Data type: Enumeration
Vary Colors by Color List	Specifies whether or not to make colors for the featured measures vary. Data type: Boolean
Hint (unavailable to org chart)	
Color	Specifies the color of the hint text, applied when the property Fill Type is set to color. Choose a color from the drop-down list, enter a hexadecimal RGB value (for example, 0xff0000), or select Custom from the drop-down list to specify the color. Data type: String
Fill Transparency	Specifies the transparency of the background, in percent. Enter a numeric value to change the transparency. Data type: Integer
Fill Type	Specifies the fill pattern for the hint text. Choose an option from the drop-down list. Data type: Enumeration
Font Effect	Specifies the special effect for the hint. Choose an option from the drop-down list. Data type: Integer
Font Face	Specifies the font face for the hint. Choose an option from the drop-down list. Data type: Enumeration
Font Rotation	Specifies the rotation angle of the hint around its center, in degrees. The default value is 0. Enter a numeric value to change the rotation. Data type: Float
Font Script	Specifies whether the hint will be in superscript or subscript form, or neither of them. Choose an option from the drop-down list. Data type: Enumeration
Font Shearing	Specifies the shearing transformation of the hint around its center. The default value is 0. Enter a numeric value to change the shearing. Data type: Float
Font Size	Specifies the font size for the hint. Enter an integer value to change the size. Data type: Integer
Font Strikethrough	Specifies the style for the line by which the hint text is struck through. Choose an option from the drop-down list. Data type: Enumeration
Font Style	Specifies the font style for the hint. Choose an option from the drop-down list. Data type: Enumeration
Font Underline	Specifies the underline style for the hint. Choose an option from the drop-down list. Data type: Enumeration
Gradient End Color	Specifies the color of the point where the gradient ends, applied when the property Fill Type is set to gradient. Choose a color from the drop-down list, enter a hexadecimal RGB value (for example, 0xff0000), or select Custom from the drop-down list to specify the color. Data type: String
Gradient End X	Specifies the horizontal position where the gradient ends, measured in a percentage of the hint width. Applied when the property Fill Type is set to gradient. Data type: Integer
Gradient End Y	Specifies the vertical position where the gradient ends, measured in a percentage of the hint height. Applied when the property Fill Type is set to gradient. Data type: Integer
Gradient Start Color	Specifies the color of the point where the gradient begins, applied when the property Fill Type is set to gradient. Choose a color from the drop-down list, enter a hexadecimal RGB value (for example, 0xff0000), or select Custom from the drop-down list to specify the color. Data type: String
Gradient Start X	Specifies the horizontal position where the gradient begins, measured in a percentage of the hint width. Applied when the property Fill Type is set to gradient. Data type: Integer
Gradient Start Y	Specifies the vertical position where the gradient begins, measured in a percentage of the hint height. Applied when the property Fill Type is set to gradient. Data type: Integer
Gradient Style	Specifies the gradient style for the hint text, applied when the property Fill Type is set to gradient. Choose an option from the drop-down list. Data type: Enumeration
Line Color	Specifies the color of the thin lines that are used to point to the static data labels. Applies to pie charts only. Choose a color from the drop-down list, enter a hexadecimal RGB value (for example, 0xff0000), or select Custom from the drop-down list to specify the color. Data type: String
Outline Color	Specifies the color for the hint text outline. Choose a color from the drop-down list, enter a hexadecimal RGB value (for example, 0xff0000), or select Custom from the drop-down list to specify the color. Data type: String
Outline End Caps	Specifies the ending style for the hint text outline. Choose an option from the drop-down list. Data type: Enumeration
Outline Joint	Specifies the joint style for the hint text outline. Choose an option from the drop-down list. Data type: Enumeration
Outline Outlined	Specifies whether to show the hint text outline in outline form. If the property is set to true, the outline will be shown in outline form; otherwise, in whole form. Data type: Boolean
Outline Style	Specifies the style for the hint text outline. Choose an option from the drop-down list. Data type: Enumeration
Outline Thickness	Specifies the thickness of the hint text outline. Enter a numeric value to change the thickness. Data type: Float
Outline Transparency	Specifies the transparency for the hint text outline, in percent. Enter a numeric value to change the transparency. Data type: Integer
Outline Variable Dash	Specifies whether to resize the dash automatically. If the property is set to true, the dash size will be adjusted automatically; otherwise, the dash size will be of fixed size. Data type: Boolean
Show Category and Series	Specifies whether to include the category and series values in the hints of the data markers. Data type: Boolean
Show Fields	Specifies the fields you want to show in the hint. Available to heat map only. Choose an option from the drop-down list: • **Auto** The fields shown in the hint are all the groups, size-by fields, and color-by fields. • **Customized** Opens the Show Fields dialog to customize the fields shown in the hint flexibly. The available fields are the same as the fields which can be inserted into the innermost rectangle. Data type: String
Show Tips	Specifies whether to show the hint that displays the value of a data marker when the mouse pointer points at the data marker in Designer view mode, in HTML result, or at runtime on JReport Server. If this property is set to false, all the other hint properties will be ignored. Data type: Boolean
Texture Background Color	Specifies the background color of the texture, applied when the property Fill Type is set to texture. Choose a color from the drop-down list, enter a hexadecimal RGB value (for example, 0xff0000), or select Custom from the drop-down list to specify the color. Data type: String
Texture Foreground Color	Specifies the foreground color of the texture, applied when the property Fill Type is set to texture. Choose a color from the drop-down list, enter a hexadecimal RGB value (for example, 0xff0000), or select Custom from the drop-down list to specify the color. Data type: String
Texture Style	Specifies the texture style for the hint text, applied when the property Fill Type is set to texture. Choose an option from the drop-down list. Data type: Enumeration
Category (X) Axis (unavailable to heat map and org charts)	
Axis Color	Specifies the color for the axis. Choose a color from the drop-down list, enter a hexadecimal RGB value (for example, 0xff0000), or select Custom from the drop-down list to specify the color. Data type: String
Axis End Caps	Specifies the ending style for the axis. Choose an option from the drop-down list. Data type: Enumeration
Axis Joint	Specifies the joint style for the axis. Choose an option from the drop-down list. Data type: Enumeration
Axis Outlined	Specifies whether to show the axis in outline form. If the property is set to true, the axis will be shown in outline form; otherwise, in whole form. Data type: Boolean
Axis Placement	Specifies the position of X axis across the Y axis. Choose an option from the drop-down list. • **origin** - The X axis will cross the Y axis at the origin. • **bottom** - The X axis will cross the Y axis at its minimum value point. Data type: Enumeration
Axis Style	Specifies the style for the axis. Choose an option from the drop-down list. Data type: Enumeration
Axis Thickness	Specifies the thickness for the axis. Enter a numeric value to change the thickness. Data type: Float
Axis Transparency	Specifies the transparency for the axis, in percent. Enter a numeric value to change the transparency. Data type: Integer
Axis Variable Dash	Specifies whether to resize the dash automatically. If the property is set to true, the dash size will be adjusted automatically; otherwise, the dash size will be of fixed size. Data type: Boolean
Gridline Color	Specifies the color for the gridlines. Choose a color from the drop-down list, enter a hexadecimal RGB value (for example, 0xff0000), or select Custom from the drop-down list to specify the color. Data type: String
Gridline End Caps	Specifies the ending style for the gridlines. Choose an option from the drop-down list. Data type: Enumeration
Gridline Joint	Specifies the joint style for the gridlines. Choose an option from the drop-down list. Data type: Enumeration
Gridline Outlined	Specifies whether to show the gridlines in outline form. If the property is set to true, the gridlines will be shown in outline form; otherwise, in whole form. Data type: Boolean
Gridline Style	Specifies the style for the gridlines. Choose an option from the drop-down list. Data type: Integer
Gridline Thickness	Specifies the thickness for the gridlines. Enter a numeric value to change the thickness. Data type: Float
Gridline Transparency	Specifies the transparency for the gridlines, in percent. Enter a numeric value to change the transparency. Data type: Integer
Gridline Variable Dash	Specifies whether to resize the dash automatically. If the property is set to true, the dash size will be adjusted automatically; otherwise, the dash size will be of fixed size. Data type: Boolean
Increment	Specifies the increased amount between two adjacent values on the axis. The default is to let the increased amount be determined by JReport Designer. Applies to bubble charts which use the axis to show numeric data only. Data type: Integer
Label Axis Gap	Specifies the distance between the labels and the axis. Enter a numeric value to change the gap. Data type: Float
Label Color	Specifies the color for the label text. Choose a color from the drop-down list, enter a hexadecimal RGB value (for example, 0xff0000), or select Custom from the drop-down list to specify the color. Data type: String
Label Every N Tick Marks	Specifies the frequency at which the tick marks will be labeled. Type 1 to label every tick mark, type 2 to label every two tick marks, type 3 to label every three tick marks, and so on. Data type: Integer
Label Fill Transparency	Specifies the transparency for the label text, in percent. Enter a numeric value to change the transparency. Data type: Integer
Label Fill Type	Specifies the fill pattern for the label text. Choose an option from the drop-down list. Data type: Enumeration
Label Font Automatic Orientation	Specifies whether to adjust the rotation angle of the label text on the axis automatically. • **true** - The rotation angle of the label text will be adjusted automatically according to the length of the text, in degrees. If the text can be completely displayed horizontally, the default rotation angle will be 0; if the text cannot be completely displayed horizontally, the default rotation angle will be 30 anticlockwise, and the cut off part will be shown as suspension points. • **false** - If selected, the rotation angle of the label text will be determined by value of the property Label Font Rotation. Data type: Boolean **Note:** The property Label Font Automatic Orientation takes higher priority over the property Label Word Wrap.
Label Font Effect	Specifies the special effect for the label text. Choose an option from the drop-down list. Data type: Enumeration
Label Font Face	Specifies the font face for the label text. Choose an option from the drop-down list. Data type: Enumeration
Label Font Rotation	Specifies the rotation angle of the label text around its center, in degrees. The default value is 0. Enter a numeric value to change the rotation. Data type: Float
Label Font Script	Specifies whether the label text will be in superscript or subscript form, or neither of them. Choose an option from the drop-down list. Data type: Enumeration
Label Font Shearing	Specifies the shearing transformation of the label text around its center. The default value is 0. Enter a numeric value to change the shearing. Data type: Float
Label Font Size	Specifies the font size for the label text. Enter an integer value to change the size. Data type: Integer
Label Font Strikethrough	Specifies the style for the line by which the label text is struck through. Choose an option from the drop-down list. Data type: Enumeration
Label Font Style	Specifies the font style for the label text. Choose an option from the drop-down list. Data type: Enumeration
Label Font Underline	Specifies the underline style for the label text. Choose an option from the drop-down list. Data type: Enumeration
Label Gradient End Color	Specifies the color of the point where the gradient ends, applied when the property Label Fill Type is set to gradient. Choose a color from the drop-down list, enter a hexadecimal RGB value (for example, 0xff0000), or select Custom from the drop-down list to specify the color. Data type: String
Label Gradient End X	Specifies the horizontal position where the gradient ends, measured in a percentage of the label text width. Applied when the property Label Fill Type is set to gradient. Data type: Integer
Label Gradient End Y	Specifies the vertical position of the gradient endpoint, measured in a percentage of the label text height. Applied when the property Label Fill Type is set to gradient. Data type: Integer
Label Gradient Start Color	Specifies the color of the point where the gradient begins, applied when the property Label Fill Type is set to gradient. Choose a color from the drop-down list, enter a hexadecimal RGB value (for example, 0xff0000), or select Custom from the drop-down list to specify the color. Data type: String
Label Gradient Start X	Specifies the horizontal position where the gradient begins, measured in a percentage of the label text width. Applied when the property Label Fill Type is set to gradient. Data type: Integer
Label Gradient Start Y	Specifies the vertical position where the gradient begins, measured in a percentage of the label text height. Applied when the property Label Fill Type is set to gradient. Data type: Integer
Label Gradient Style	Specifies the gradient style for the label text, applied when the property Label Fill Type is set to gradient. Choose an option from the drop-down list. Data type: Enumeration
Label Outline Color	Specifies the color for the label text outline. Choose a color from the drop-down list, enter a hexadecimal RGB value (for example, 0xff0000), or select Custom from the drop-down list to specify the color. Data type: String
Label Outline End Caps	Specifies the ending style for the label text outline. Choose an option from the drop-down list. Data type: Enumeration
Label Outline Joint	Specifies the joint style for the label text outline. Choose an option from the drop-down list. Data type: Enumeration
Label Outline Outlined	Specifies whether to show the label text outline in outline form. If the property is set to true, the outline will be shown in outline form; otherwise, in whole form. Data type: Boolean
Label Outline Style	Specifies the style for the label text outline. Choose an option from the drop-down list. Data type: Enumeration
Label Outline Thickness	Specifies the thickness for the label text outline. Enter a numeric value to change the thickness. Data type: Float
Label Outline Transparency	Specifies the transparency for the label text outline, in percent. Enter a numeric value to change the transparency. Data type: Integer
Label Outline Variable Dash	Specifies whether to resize the dash automatically. If the property is set to true, the dash size will be adjusted automatically; otherwise, the dash size will be of fixed size. Data type: Boolean
Label Position	Specifies whether to show the label text outside or inside the axis. Takes effect only when Use Depth is set to false. Data type: Enumeration
Label Texture Background Color	Specifies the background color of the texture, applied when the property Label Fill Type is set to texture. Choose a color from the drop-down list, enter a hexadecimal RGB value (for example, 0xff0000), or select Custom from the drop-down list to specify the color. Data type: String
Label Texture Foreground Color	Specifies the foreground color of the texture, applied when the property Label Fill Type is set to texture. Choose a color from the drop-down list, enter a hexadecimal RGB value (for example, 0xff0000), or select Custom from the drop-down list to specify the color. Data type: String
Label Texture Style	Specifies the texture style for the label text, applied when the property Label Fill Type is set to texture. Choose an option from the drop-down list. Data type: Enumeration
Label Word Wrap	Specifies whether to enable the word wrap function for the label text, not applying to bubble charts and gauge charts. If the text contains special characters (such as ",", "." and space), it will be broken at the position of one of the special characters. Data type: Boolean
Major Tick Mark Length	Specifies the length of the major tick marks. Enter a numeric value to change the length. Data type: Float
Major Tick Mark Line Color	Specifies the color of the major tick marks. Choose a color from the drop-down list, enter a hexadecimal RGB value (for example, 0xff0000), or select Custom from the drop-down list to specify the color. Data type: String
Major Tick Mark Line End Caps	Specifies the ending style for the major tick marks. Choose an option from the drop-down list. Data type: Enumeration
Major Tick Mark Line Joint	Specifies the joint style for the major tick marks. Choose an option from the drop-down list. Data type: Enumeration
Major Tick Mark Line Outlined	Specifies whether to show the major tick marks in outline form. If the property is set to true, the tick marks will be shown in outline form; otherwise, in whole form. Data type: Boolean
Major Tick Mark Line Style	Specifies the line type for the major tick marks. Choose an option from the drop-down list. Data type: Enumeration
Major Tick Mark Line Thickness	Specifies the thickness of the major tick marks. Enter a numeric value to change the thickness. Data type: Float
Major Tick Mark Line Transparency	Specifies the transparency of the major tick marks, in percent. Enter a numeric value to change the transparency. Data type: Integer
Major Tick Mark Line Variable Dash	Specifies whether to adjust the dash size automatically. Applied when Major Tick Mark Line Style is not set to solid line. • **true** - The dash size will be adjusted automatically. • **false** - The dash size will be of fixed size. Data type: Boolean
Major Tick Mark Type	Specifies the position of the major tick marks relative to the axis. Choose an option from the drop-down list. Data type: Enumeration
Maximum Value	Specifies the maximum value of the data appearing on the X axis. Applies to bubble charts which use the axis to show numeric data only. Data type: Float
Minimum Value	Specifies the minimum value of the data appearing on the X axis. Applies to bubble charts which use the axis to show numeric data only. Data type: Float
Minor Tick Mark Length	Specifies the length of the minor tick marks. Enter a numeric value to change the length. Data type: Float
Minor Tick Mark Type	Specifies the position of the minor tick marks relative to the axis. Choose an option from the drop-down list. Data type: Enumeration
Number of Labels	Specifies the number of tick mark labels that will be shown. The default is to let JReport Designer determine the number of labels to be shown. Data type: Integer
Number of Tick Marks	Specifies the number of tick marks that will be shown. The default is to let the number be determined by JReport Designer. Applies to bubble charts only. Data type: Integer
Number of Visible Values	Specifies how many data items will be displayed on the scrollbar and the axis. Applied only when Scrollable Chart is set to true for 2-D charts of bar, bench, line, and area types. Data type: Integer
Percentage of Scrolling Area Height	Specifies the percentage the scrollbar occupies the whole size of the chart. Applied only when Scrollable Chart is set to true for 2-D charts of bar, bench, line, and area types. Data type: Integer
Position Axis	Specifies the position at which the data point and label will be displayed on the X axis. • **on tick marks** - The data point and label will be aligned with each tick mark on the X axis. • **between tick marks** - The data point and label will be displayed between each two tick marks on the X axis. Data type: Enumeration
Scrollable Chart	Specifies whether to make the chart scrollable. If true, a scrollbar will be added in the chart, with which you can control the visible value range on the axis. Available to 2-D charts of bar, bench, line, and area types. Data type: Boolean
Show Axis	Specifies whether to show the axis. Data type: Boolean
Show Axis Label Tips	Specifies whether to show the complete label text when the mouse pointer points at a label on the axis in Designer view mode, in HTML result, or at runtime on JReport Server. Data type: Boolean
Show Chart in Scrolling Area	Specifies whether to show the thumbnail chart on the scrollbar. Applied only when Scrollable Chart is set to true for 2-D charts of bar, bench, line and area types. Data type: Boolean
Show Gridline	Specifies whether to show the gridlines perpendicular to the axis. Data type: Boolean
Show Tick Mark Labels	Specifies whether to show the tick mark labels. Data type: Boolean
Start Value	Specifies the position where the X axis label starts to be shown, applied when the property Use Start Value is set to true. Applies to 2-D charts that have a wall only. Data type: Float
Use Best Effect	Specifies whether to hide the label which would overlap with the one in front of it. Data type: Boolean
Use Start Value	Specifies whether to set the start value of X axis. This property applies to bubble charts, and only if the value on the category axis is numeric. Data type: Boolean
Value (Y) Axis (unavailable to heat map and org charts)	
Auto Scale on Y Axis	Specifies whether to recalculate the scale of the axis when the value of the property Number of Visible Values is changed. Applied only when Scrollable Chart is set to true for 2-D charts of bar, bench, line, and area types. Data type: Boolean
Axis Color	Specifies the color for the axis. Choose a color from the drop-down list, enter a hexadecimal RGB value (for example, 0xff0000), or select Custom from the drop-down list to specify the color. Data type: String
Axis End Caps	Specifies the ending style for the axis. Choose an option from the drop-down list. Data type: Enumeration
Axis Joint	Specifies the joint style for the axis. Choose an option from the drop-down list. Data type: Enumeration
Axis Outlined	Specifies whether to show the axis in outline form. If the property is set to true, the axis will be shown in outline form; otherwise, in whole form. Data type: Boolean
Axis Placement	Specifies the position of the Y axis across the X axis. Choose an option from the drop-down list. • **origin** - The Y axis will cross the X axis at the origin. • **bottom** - The Y axis will cross the X axis at its minimum value point. Data type: Enumeration
Axis Style	Specifies the style for the axis. Choose an option from the drop-down list. Data type: Enumeration
Axis Thickness	Specifies the thickness for the axis. Enter a numeric value to change the thickness. Data type: Float
Axis Transparency	Specifies the transparency for the axis, in percent. Enter a numeric value to change the transparency. Data type: Integer
Axis Variable Dash	Specifies whether to resize the dash automatically. If the property is set to true, the dash size will be adjusted automatically; otherwise, the dash size will be of fixed size. Data type: Boolean

Gridline Color	Specifies the color for the gridlines. Choose a color from the drop-down list, enter a hexadecimal RGB value (for example, 0xff0000), or select Custom from the drop-down list to specify the color. Data type: String
Gridline End Caps	Specifies the ending style for the gridlines. Choose an option from the drop-down list. Data type: Enumeration
Gridline Joint	Specifies the joint style for the gridlines. Choose an option from the drop-down list. Data type: Enumeration
Gridline Outlined	Specifies whether to show the gridlines in outline form. If the property is set to true, the gridlines will be shown in outline form; otherwise, in whole form. Data type: Boolean
Gridline Style	Specifies the style for the gridlines. Choose an option from the drop-down list. Data type: Enumeration
Gridline Thickness	Specifies the thickness for the gridlines. Enter a numeric value to change the thickness. Data type: Float
Gridline Transparency	Specifies the transparency for the gridlines, in percent. Enter a numeric value to change the transparency. Data type: Integer
Gridline Variable Dash	Specifies whether to resize the dash automatically. If the property is set to true, the dash size will be adjusted automatically; otherwise, the dash size will be of fixed size. Data type: Boolean
Increment	Specifies the increased amount between two adjacent values on the axis. The default is to let the increased amount be determined by JReport Designer. Data type: Integer
Label Axis Gap	Specifies the distance between the labels and the axis. Enter a numeric value to change the gap. Data type: Float
Label Color	Specifies the color for the label text. Choose a color from the drop-down list, enter a hexadecimal RGB value (for example, 0xff0000), or select Custom from the drop-down list to specify the color. Data type: String
Label Every N Tick Marks	Specifies the frequency at which the tick marks will be labeled. Type 1 to label every tick mark, type 2 to label every two tick marks, type 3 to label every three tick marks, and so on. Data type: Integer
Label Fill Transparency	Specifies the transparency for the label text, in percent. Enter a numeric value to change the transparency. Data type: Integer
Label Fill Type	Specifies the fill pattern for the label text. Choose an option from the drop-down list. Data type: Enumeration
Label Font Automatic Orientation	Specifies whether to adjust the rotation angle of the label text on the axis automatically. • **true** - The rotation angle of the label text will be adjusted automatically according to the length of the text, in degrees. If the text can be completely displayed horizontally, the default rotation angle will be 0; if the text cannot be completely displayed horizontally, the default rotation angle will be 30 anticlockwise, and the cut off part will be shown as suspension points. • **false** - If selected, the rotation angle of the label text will be determined by value of the property Label Font Rotation. Data type: Boolean **Note:** The property Label Font Automatic Orientation takes higher priority over the property Label Word Wrap.
Label Font Effect	Specifies the special effect for the label text. Choose an option from the drop-down list. Data type: Enumeration
Label Font Face	Specifies the font face for the label text. Choose an option from the drop-down list. Data type: Enumeration
Label Font Rotation	Specifies the rotation angle of the label text around its center, in degrees. The default value is 0. Enter a numeric value to change the rotation. Data type: Float
Label Font Script	Specifies whether the label text will in superscript or subscript form, or neither of them. Choose an option from the drop-down list. Data type: Enumeration
Label Font Shearing	Specifies the shearing transformation of the label text around its center. The default value is 0. Enter a numeric value to change the shearing. Data type: Float
Label Font Size	Specifies the font size for the label text. Enter an integer value to change the size. Data type: Integer
Label Font Strikethrough	Specifies the style for the line by which the label text is struck through. Choose an option from the drop-down list. Data type: Enumeration
Label Font Style	Specifies the font style for the label text. Choose an option from the drop-down list. Data type: Enumeration
Label Font Underline	Specifies the underline style for the label text. Choose an option from the drop-down list. Data type: Enumeration
Label Gradient End Color	Specifies the color of the point where the gradient ends, applied when the property Label Fill Type is set to gradient. Choose a color from the drop-down list, enter a hexadecimal RGB value (for example, 0xff0000), or select Custom from the drop-down list to specify the color. Data type: String
Label Gradient End X	Specifies the horizontal position where the gradient ends, measured in a percentage of the label text width. Applied when the property Label Fill Type is set to gradient. Data type: Integer
Label Gradient End Y	Specifies the vertical position of the gradient endpoint, measured in a percentage of the label text height. Applied when the property Label Fill Type is set to gradient. Data type: Integer
Label Gradient Start Color	Specifies the color of the point where the gradient begins, applied when the property Label Fill Type is set to gradient. Choose a color from the drop-down list, enter a hexadecimal RGB value (for example, 0xff0000), or select Custom from the drop-down list to specify the color. Data type: String
Label Gradient Start X	Specifies the horizontal position where the gradient begins, measured in a percentage of the label text width. Applied when the property Label Fill Type is set to gradient. Data type: Integer
Label Gradient Start Y	Specifies the vertical position where the gradient begins, measured in a percentage of the label text height. Applied when the property Label Fill Type is set to gradient. Data type: Integer
Label Gradient Style	Specifies the gradient style for the label text, applied when the property Label Fill Type is set to gradient. Choose an option from the drop-down list. Data type: Integer
Label Outline Color	Specifies the color for the label text outline. Choose a color from the drop-down list, enter a hexadecimal RGB value (for example, 0xff0000), or select Custom from the drop-down list to specify the color. Data type: String
Label Outline End Caps	Specifies the ending style for the label text outline. Choose an option from the drop-down list. Data type: Enumeration
Label Outline Joint	Specifies the joint style for the label text outline. Choose an option from the drop-down list. Data type: Enumeration
Label Outline Outlined	Specifies whether to show the label text outline in outline form. If the property is set to true, the outline will be shown in outline form; otherwise, in whole form. Data type: Boolean
Label Outline Style	Specifies the style for the label text outline. Choose an option from the drop-down list. Data type: Enumeration
Label Outline Thickness	Specifies the thickness for the label text outline. Enter a numeric value to change the thickness. Data type: Float
Label Outline Transparency	Specifies the transparency for the label text outline, in percent. Enter a numeric value to change the transparency. Data type: Integer
Label Outline Variable Dash	Specifies whether to resize the dash automatically. If the property is set to true, the dash size will be adjusted automatically; otherwise, the dash size will be of fixed size. Data type: Boolean
Label Position	Specifies whether to show the label text outside or inside the axis. Takes effect only when the property Use Depth is set to false. Data type: Enumeration
Label Texture Background Color	Specifies the background color of the texture, applied when the property Label Fill Type is set to texture. Choose a color from the drop-down list, enter a hexadecimal RGB value (for example, 0xff0000), or select Custom from the drop-down list to specify the color. Data type: String
Label Texture Foreground Color	Specifies the foreground color of the texture, applied when the property Label Fill Type is set to texture. Choose a color from the drop-down list, enter a hexadecimal RGB value (for example, 0xff0000), or select Custom from the drop-down list to specify the color. Data type: String
Label Texture Style	Specifies the texture style for the label text, applied when the property Label Fill Type is set to texture. Choose an option from the drop-down list. Data type: Enumeration
Label Word Wrap	Specifies whether to enable the word wrap function for the label text, not applying to bubble charts and gauge charts. If the text contains special characters (such as ",", "." and space), it will be broken at the position of one of the special characters. Data type: Boolean
Major Tick Mark Length	Specifies the length of the major tick marks. Enter a numeric value to change the length. Data type: Float
Major Tick Mark Line Color	Specifies the color of the major tick marks. Choose a color from the drop-down list, enter a hexadecimal RGB value (for example, 0xff0000), or select Custom from the drop-down list to specify the color. Data type: String
Major Tick Mark Line End Caps	Specifies the ending style for the major tick marks. Choose an option from the drop-down list. Data type: Enumeration
Major Tick Mark Line Joint	Specifies the joint style for the major tick marks. Choose an option from the drop-down list. Data type: Enumeration
Major Tick Mark Line Outlined	Specifies whether to show the major tick marks in outline form. If the property is set to true, the tick marks will be shown in outline form; otherwise, in whole form. Data type: Boolean
Major Tick Mark Line Style	Specifies the line type for the major tick marks. Choose an option from the drop-down list. Data type: Enumeration
Major Tick Mark Line Thickness	Specifies the thickness of the major tick marks. Enter a numeric value to change the thickness. Data type: Float
Major Tick Mark Line Transparency	Specifies the transparency of the major tick marks, in percent. Enter a numeric value to change the transparency. Data type: Integer
Major Tick Mark Line Variable Dash	Specifies whether to adjust the dash size automatically. Applied when Major Tick Mark Line Style is not set to solid line. • **true** - The dash size will be adjusted automatically. • **false** - The dash size will be of fixed size. Data type: Boolean
Major Tick Mark Type	Specifies the position of the major tick marks relative to the axis. Choose an option from the drop-down list. Data type: Enumeration
Maximum Value	Specifies the maximum value of the data appearing on the Y axis. Data type: Float
Minimum Value	Specifies the minimum value of the data appearing on the Y axis. Data type: Float
Minor Tick Mark Length	Specifies the length of the minor tick marks. Enter a numeric value to change the length. Data type: Float
Minor Tick Mark Type	Specifies the position of the minor tick marks relative to the axis. Choose an option from the drop-down list. Data type: Enumeration
Number of Labels	Specifies the number of tick mark labels that will be shown. The default is to let JReport Designer determine the number of labels to be shown. Data type: Integer
Number of Tick Marks	Specifies the number of tick marks that will be shown. The default is to let the number be determined by JReport Designer. Data type: Integer
Show Axis	Specifies whether to show the axis. Data type: Boolean
Show Axis Label Tips	Specifies whether to show the complete label text when the mouse pointer points at a label on the axis in Designer view mode, in HTML result, or at runtime on JReport Server. Data type: Boolean
Show Gridline	Specifies whether to show the gridlines perpendicular to the axis. Data type: Boolean
Show Percent	Specifies whether to show the value labels on the axis in percent. Only applies to bullet, bar/bench, line, and area chart that are not 100% stacked type. Data type: Boolean
Show Tick Mark Labels	Specifies whether to show the tick mark labels. Data type: Boolean
Start Value	Specifies the position where the axis label starts to be shown, applied when the property Use Start Value is set to true. Applies to 2-D charts that have a wall only. Data type: Float
Use Best Effect	Specifies whether to hide the label which would overlap with the one in front of it. Data type: Boolean
Use Start Value	Specifies whether to set the start value of the axis. Applies to 2-D charts that have a wall only. Data type: Boolean
Series (Z) Axis (available to bubble gauge charts only)	
Axis Color	Specifies the color for the axis. Choose a color from the drop-down list, enter a hexadecimal RGB value (for example, 0xff0000), or select Custom from the drop-down list to specify the color. Data type: String
Axis End Caps	Specifies the ending style for the axis. Choose an option from the drop-down list. Data type: Enumeration
Axis Joint	Specifies the joint style for the axis. Choose an option from the drop-down list. Data type: Enumeration
Axis Outlined	Specifies whether to show the axis in outline form. If the property is set to true, the axis will be shown in outline form; otherwise, in whole form. Data type: Boolean
Axis Placement	Specifies the position of the Z axis. Choose an option from the drop-down list. Data type: Enumeration
Axis Style	Specifies the style for the axis. Choose an option from the drop-down list. Data type: Enumeration
Axis Thickness	Specifies the thickness for the axis. Enter a numeric value to change the thickness. Data type: Float
Axis Transparency	Specifies the transparency for the axis, in percent. Enter a numeric value to change the transparency. Data type: Integer
Axis Variable Dash	Specifies whether to resize the dash automatically. If the property is set to true, the dash size will be adjusted automatically; otherwise, the dash size will be of fixed size. Data type: Boolean
Gridline Color	Specifies the color of the gridlines. Choose a color from the drop-down list, enter a hexadecimal RGB value (for example, 0xff0000), or select Custom from the drop-down list to specify the color. Data type: String
Gridline End Caps	Specifies the ending style for the gridlines. Choose an option from the drop-down list. Data type: Enumeration
Gridline Joint	Specifies the joint style for the gridlines. Choose an option from the drop-down list. Data type: Enumeration
Gridline Outlined	Specifies whether to show the gridlines in outline form. If the property is set to true, the gridlines will be shown in outline form; otherwise, in whole form. Data type: Boolean
Gridline Style	Specifies the style for the gridlines. Choose an option from the drop-down list. Data type: Enumeration
Gridline Thickness	Specifies the thickness for the gridlines. Enter a numeric value to change the thickness. Data type: Float
Gridline Transparency	Specifies the gridline transparency, in percent. Enter a numeric value to change the transparency. Data type: Integer
Gridline Variable Dash	Specifies whether to resize the dash automatically. If the property is set to true, the dash size will be adjusted automatically; otherwise, the dash size will be of fixed size. Data type: Boolean
Label Axis Gap	Specifies the distance between the labels and the axis. Enter a numeric value to change the gap. Data type: Float
Label Color	Specifies the color for the label text. Choose a color from the drop-down list, enter a hexadecimal RGB value (for example, 0xff0000), or select Custom from the drop-down list to specify the color. Data type: String
Label Every N Tick Marks	Specifies the frequency at which the tick marks will be labeled. Type 1 to label every tick mark, type 2 to label every two tick marks, type 3 to label every three tick marks, and so on. Data type: Integer
Label Fill Transparency	Specifies the transparency for the label text, in percent. Enter a numeric value to change the transparency. Data type: Integer
Label Fill Type	Specifies the fill pattern for the label text. Choose an option from the drop-down list. Data type: Enumeration
Label Font Automatic Orientation	Specifies whether to adjust the rotation angle of the label text on the axis automatically. • **true** - The rotation angle of the label text will be adjusted automatically according to the length of the text, in degrees. If the text can be completely displayed horizontally, the default rotation angle will be 0; if the text cannot be completely displayed horizontally, the default rotation angle will be 30 anticlockwise, and the cut off part will be shown as suspension points. • **false** - If selected, the rotation angle of the label text will be determined by value of the property Label Font Rotation. Data type: Boolean **Note:** The property Label Font Automatic Orientation takes higher priority over the property Label Word Wrap.
Label Font Effect	Specifies the special effect for the label text. Choose an option from the drop-down list. Data type: Enumeration
Label Font Face	Specifies the font face for the label text. Choose an option from the drop-down list. Data type: Enumeration
Label Font Rotation	Specifies the rotation angle of the label text around its center, in degrees. The default value is 0. Data type: Float
Label Font Script	Specifies whether the label text will in superscript or subscript form, or neither of them. Choose an option from the drop-down list. Data type: Enumeration
Label Font Shearing	Specifies the shearing transformation of the label text around its center. The default value is 0. Data type: Float
Label Font Size	Specifies the font size for the label text. Enter an integer value to change the size. Data type: Integer
Label Font Strikethrough	Specifies the style for the line by which the label text is struck through. Choose an option from the drop-down list. Data type: Enumeration
Label Font Style	Specifies the font style for the label text. Choose an option from the drop-down list. Data type: Enumeration
Label Font Underline	Specifies the underline style for the label text. Choose an option from the drop-down list. Data type: Enumeration
Label Gradient End Color	Specifies the color of the point where the gradient ends, applied when the property Label Fill Type is set to gradient. Choose a color from the drop-down list, enter a hexadecimal RGB value (for example, 0xff0000), or select Custom from the drop-down list to specify the color. Data type: String
Label Gradient End X	Specifies the horizontal position where the gradient ends, measured in a percentage of the label text width. Applied when the property Label Fill Type is set to gradient. Data type: Integer
Label Gradient End Y	Specifies the vertical position of the gradient endpoint, measured in a percentage of the label text height. Applied when the property Label Fill Type is set to gradient. Data type: Integer
Label Gradient Start Color	Specifies the color of the point where the gradient begins, applied when the property Label Fill Type is set to gradient. Choose a color from the drop-down list, enter a hexadecimal RGB value (for example, 0xff0000), or select Custom from the drop-down list to specify the color. Data type: String
Label Gradient Start X	Specifies the horizontal position where the gradient begins, measured in a percentage of the label text width. Applied when the property Label Fill Type is set to gradient. Data type: Integer
Label Gradient Start Y	Specifies the vertical position where the gradient begins, measured in a percentage of the label text height. Applied when the property Label Fill Type is set to gradient. Data type: Integer
Label Gradient Style	Specifies the gradient style for the label text, applied when the property Label Fill Type is set to gradient. Choose an option from the drop-down list. Data type: Enumeration
Label Outline Color	Specifies the color for the label text outline. Choose a color from the drop-down list, enter a hexadecimal RGB value (for example, 0xff0000), or select Custom from the drop-down list to specify the color. Data type: String
Label Outline End Caps	Specifies the ending style for the label text outline. Choose an option from the drop-down list. Data type: Enumeration
Label Outline Joint	Specifies the joint style for the label text outline. Choose an option from the drop-down list. Data type: Enumeration
Label Outline Outlined	Specifies whether to show the label text outline in outline form. If the property is set to true, the outline will be shown in outline form; otherwise, in whole form. Data type: Boolean
Label Outline Style	Specifies the style for the label text outline. Choose an option from the drop-down list. Data type: Enumeration
Label Outline Thickness	Specifies the thickness for the label text outline. Enter a numeric value to change the thickness. Data type: Float
Label Outline Transparency	Specifies the transparency for the label text outline, in percent. Enter a numeric value to change the transparency. Data type: Integer
Label Outline Variable Dash	Specifies whether to resize the dash automatically. If the property is set to true, the dash size will be adjusted automatically; otherwise, the dash size will be of fixed size. Data type: Boolean
Label Texture Background Color	Specifies the background color of the texture, applied when the property Label Fill Type is set to texture. Choose a color from the drop-down list, enter a hexadecimal RGB value (for example, 0xff0000), or select Custom from the drop-down list to specify the color. Data type: String
Label Texture Foreground Color	Specifies the foreground color of the texture, applied when the property Label Fill Type is set to texture. Choose a color from the drop-down list, enter a hexadecimal RGB value (for example, 0xff0000), or select Custom from the drop-down list to specify the color. Data type: String
Label Texture Style	Specifies the texture style for the label text, applied when the property Label Fill Type is set to texture. Choose an option from the drop-down list. Data type: Enumeration
Label Word Wrap	Specifies whether to enable the word wrap function for the label text, not applying to radar charts, bubble charts, scatter charts and gauge charts. If the text contains special characters (such as ",", "." and space), it will be broken at the position of one of the special characters. Data type: Boolean
Major Tick Mark Length	Specifies the length of the major tick marks. Enter a numeric value to change the length. Data type: Float
Major Tick Mark Line Color	Specifies the color of the major tick marks. Choose a color from the drop-down list, enter a hexadecimal RGB value (for example, 0xff0000), or select Custom from the drop-down list to specify the color. Data type: String
Major Tick Mark Line End Caps	Specifies the ending style for the major tick marks. Choose an option from the drop-down list. Data type: Enumeration
Major Tick Mark Line Joint	Specifies the joint style for the major tick marks. Choose an option from the drop-down list. Data type: Enumeration
Major Tick Mark Line Outlined	Specifies whether to show the major tick marks in outline form. If the property is set to true, the tick marks will be shown in outline form; otherwise, in whole form. Data type: Boolean
Major Tick Mark Line Style	Specifies the line type for the major tick marks. Choose an option from the drop-down list. Data type: Enumeration
Major Tick Mark Line Thickness	Specifies the thickness of the major tick marks. Enter a numeric value to change the thickness. Data type: Float
Major Tick Mark Line Transparency	Specifies the transparency of the major tick marks, in percent. Enter a numeric value to change the transparency. Data type: Integer
Major Tick Mark Line Variable Dash	Specifies whether to adjust the dash size automatically. Applied when Major Tick Mark Line Style is not set to solid line. • **true** - The dash size will be adjusted automatically. • **false** - The dash size will be of fixed size. Data type: Boolean
Major Tick Mark Type	Specifies the position of the major tick marks relative to the axis. Choose an option from the drop-down list. Data type: Enumeration
Minor Tick Mark Length	Specifies the length of the minor tick marks. Enter a numeric value to change the length. Data type: Float
Minor Tick Mark Type	Specifies the position of the minor tick marks relative to the axis. Choose an option from the drop-down list. Data type: Enumeration
Number of Labels	Specifies the number of tick mark labels that will be shown. The default is to let JReport Designer determine the number of labels to be shown. Data type: Integer
Show Axis	Specifies whether to show the axis. Data type: Boolean
Show Axis Label Tips	Specifies whether to show the complete label text when the mouse pointer points at a label on the axis in Designer view mode, in HTML result, or at runtime on JReport Server. Data type: Boolean
Show Gridline	Specifies whether to show the gridlines perpendicular to the axis. Data type: Boolean
Show Tick Mark Labels	Specifies whether to show the major tick mark labels. Data type: Boolean
Use Best Effect	Specifies whether to hide the label which would overlap with the one in front of it. Data type: Boolean
Wall (unavailable to gauge, heat map and org chart)	
Border Color	Specifies the color of the border of the object. Choose a color from the drop-down list, enter a hexadecimal RGB value (for example, 0xff0000), or select Custom from the drop-down list to specify the color. Data type: String
Border End Caps	Specifies the ending style of the wall border line. Choose an option from the drop-down list. Data type: Enumeration
Border Joint	Specifies the joint style of the wall border line. Choose an option from the drop-down list. Data type: Enumeration
Border Outlined	Specifies whether to show the wall border line in Outline Path. If the property is set to true, the wall border line will be shown in Outline Path; otherwise, in Fill Path. Data type: Boolean
Border Style	Specifies the line style of the wall borders. Choose an option from the drop-down list. Data type: Enumeration
Border Thickness	Specifies the width of the border in inches or centimeters. Enter a numeric value to change the thickness. Data type: Float
Border Transparency	Specifies the transparency of the wall borders, in percent. Enter a numeric value to change the transparency. Data type: Integer
Border Variable Dash	Specifies whether to resize the dash automatically. If the property is set to true, the dash size will be adjusted automatically; otherwise, the dash size will be of fixed size. Data type: Boolean
Bound Color	Specifies the color of the bound wall, applied when the property Bound Fill Type is set to color. Choose a color from the drop-down list, enter a hexadecimal RGB value (for example, 0xff0000), or select Custom from the drop-down list to specify the color. Data type: String
Bound Fill Transparency	Specifies the transparency of the background, in percent. Enter a numeric value to change the transparency. Data type: Integer
Bound Fill Type	Specifies the fill pattern for the bound wall. Choose an option from the drop-down list. Data type: Enumeration
Bound Gradient End Color	Specifies the color of the point where the gradient ends, applied when the property Bound Fill Type is set to gradient. Choose a color from the drop-down list, enter a hexadecimal RGB value (for example, 0xff0000), or select Custom from the drop-down list to specify the color. Data type: String
Bound Gradient End X	Specifies the horizontal position where the gradient ends, measured in a percentage of the bound wall width. Applied when the property Bound Fill Type is set to gradient. Data type: Integer
Bound Gradient End Y	Specifies the vertical position where the gradient ends, measured in a percentage of the bound wall height. Applied when the property Bound Fill Type is set to gradient. Data type: Integer
Bound Gradient Start Color	Specifies the color of the point where the gradient begins, applied when the property Bound Fill Type is set to gradient. Choose a color from the drop-down list, enter a hexadecimal RGB value (for example, 0xff0000), or select Custom from the drop-down list to specify the color. Data type: String
Bound Gradient Start X	Specifies the horizontal position where the gradient begins, measured in a percentage of the bound wall width. Applied when the property Bound Fill Type is set to gradient. Data type: Integer
Bound Gradient Start Y	Specifies the vertical position where the gradient begins, measured in a percentage of the bound wall height. Applied when the property Bound Fill Type is set to gradient. Data type: Integer
Bound Gradient Style	Specifies the gradient style for the bound wall, applied when the property Bound Fill Type is set to gradient. Choose an option from the drop-down list. Data type: Enumeration
Bound Texture Background Color	Specifies the background color of the texture, applied when the property Bound Fill Type is set to texture. Choose a color from the drop-down list, enter a hexadecimal RGB value (for example, 0xff0000), or select Custom from the drop-down list to specify the color. Data type: String
Bound Texture Foreground Color	Specifies the foreground color of the texture, applied when the property Bound Fill Type is set to texture. Choose a color from the drop-down list, enter a hexadecimal RGB value (for example, 0xff0000), or select Custom from the drop-down list to specify the color. Data type: String
Bound Texture Style	Specifies the texture style for the bound wall, applied when the property Bound Fill Type is set to texture. Choose an option from the drop-down list. Data type: Enumeration
Color	Specifies the color of the wall, applied when the property Fill Type is set to color. Choose a color from the drop-down list, enter a hexadecimal RGB value (for example, 0xff0000), or select Custom from the drop-down list to specify the color. Data type: String
Fill Transparency	Specifies the transparency of the background, in percent. Enter a numeric value to change the transparency. Data type: Integer
Fill Type	Specifies the fill pattern for the wall. Choose an option from the drop-down list. Data type: Enumeration
Gradient End Color	Specifies the color of the point where the gradient ends, applied when the property Fill Type is set to gradient. Choose a color from the drop-down list, enter a hexadecimal RGB value (for example, 0xff0000), or select Custom from the drop-down list to specify the color. Data type: String
Gradient End X	Specifies the horizontal position where the gradient ends, measured in a percentage of the wall width. Applied when the property Fill Type is set to gradient. Data type: Integer
Gradient End Y	Specifies the vertical position where the gradient ends, measured in a percentage of the wall height. Applied when the property Fill Type is set to gradient. Data type: Integer
Gradient Start Color	Specifies the color of the point where the gradient begins, applied when the property Fill Type is set to gradient. Choose a color from the drop-down list, enter a hexadecimal RGB value (for example, 0xff0000), or select Custom from the drop-down list to specify the color. Data type: String
Gradient Start X	Specifies the horizontal position where the gradient begins, measured in a percentage of the wall width. Applied when the property Fill Type is set to gradient. Data type: Integer
Gradient Start Y	Specifies the vertical position where the gradient begins, measured in a percentage of the wall height. Applied when the property Fill Type is set to gradient. Data type: Integer
Gradient Style	Specifies the gradient style for the wall, applied when the property Fill Type is set to gradient. Choose an option from the drop-down list. Data type: Enumeration
Image File	Specifies the file name of the image, a portion of which will be displayed as the wall background. Applied when the property Fill Type is set to image. Enter the file name with suffix in the value cell. If the image is outside of the current catalog, you should include the full path of the image correctly. Data type: String
Image Height	The four properties, Image X, Image Y, Image Width and Image Height, together are used to define a portion of the image (the portion is a rectangle), which will be displayed as the wall background. They take effect when the property Fill Type is set to image. Image Height specifies the height of the image portion, measured in a percentage of the image height. Data type: Integer
Image Layout	Specifies the layout style for the image that will be displayed as the wall background, applied when the property Fill Type is set to image. Choose an option from the drop-down list. Data type: Enumeration
Image Width	The four properties, Image X, Image Y, Image Width and Image Height, together are used to define a portion of the image (the portion is a rectangle), which will be displayed as the wall background. They take effect when the property Fill Type is set to image. Image Width specifies the width of the image portion, measured in a percentage of the image width. Data type: Integer
Image X	The four properties, Image X, Image Y, Image Width and Image Height, together are used to define a portion of the image (the portion is a rectangle), which will be displayed as the wall background. They take effect when the property Fill Type is set to image. Image X specifies the distance between the left border of image and the portion that will contain the pattern, measured in a percentage of the image width. Data type: Integer
Image Y	The four properties, Image X, Image Y, Image Width and Image Height, together are used to define a portion of the image (the portion is a rectangle), which will be displayed as the wall background. They take effect when the property Fill Type is set to image. Image Y specifies the distance between the top border of image and the portion that will contain the pattern, measured in a percentage of the image height. Data type: Integer
Show Wall	Specifies whether to show the wall. Data type: Boolean
Texture Background Color	Specifies the background color of the texture, applied when the property Fill Type is set to texture. Choose a color from the drop-down list, enter a hexadecimal RGB value (for example, 0xff0000), or select Custom from the drop-down list to specify the color. Data type: String
Texture Foreground Color	Specifies the foreground color of the texture, applied when the property Fill Type is set to texture. Choose a color from the drop-down list, enter a hexadecimal RGB value (for example, 0xff0000), or select Custom from the drop-down list to specify the color. Data type: String
Texture Style	Specifies the texture style for the wall, applied when the property Fill Type is set to texture. Choose an option from the drop-down list. Data type: Enumeration
Threshold Line (unavailable to gauge, bubble, heat map and org charts)	
Fill Threshold1	Specifies whether to fill the first threshold area with a semi-transparent color. Data type: Boolean
Fill Threshold2	Specifies whether to fill the second threshold area with a semi-transparent color. Data type: Boolean
Fill Transparency	Specifies the transparency of the background, in percent. Enter a numeric value to change the transparency. Data type: Integer
Show Threshold Line1	Specifies whether to show the first threshold line. Data type: Boolean
Show Threshold Line2	Specifies whether to show the second threshold line. Data type: Boolean
Threshold Line Color1	Specifies the color for the first threshold line. Choose a color from the drop-down list, enter a hexadecimal RGB value (for example, 0xff0000), or select Custom from the drop-down list to specify the color. Data type: String
Threshold Line Color2	Specifies the color for the second threshold line. Choose a color from the drop-down list, enter a hexadecimal RGB value (for example, 0xff0000), or select Custom from the drop-down list to specify the color. Data type: String
Threshold Line Style	Specifies the style for the threshold lines. Choose an option from the drop-down list. Data type: Enumeration
Threshold Value1	Specifies the value for the first threshold line. Data type: Float
Threshold Value2	Specifies the value for the second threshold line. Data type: Float
MeanLine (unavailable to gauge, bubble, heat map and org charts)	
Color	Specifies the color of the mean line. Choose a color from the drop-down list, enter a hexadecimal RGB value (for example, 0xff0000), or select Custom from the drop-down list to specify the color. Data type: String
Node Style	Specifies the style of the nodes that represent the average values. Choose an option from the drop-down list. Data type: Enumeration
Show	Specifies whether to show a line that represents the average value of each category group. Not applies to bubble charts. Data type: Boolean
Type	Specifies the type of the mean line. Choose an option from the drop-down list. Data type: Enumeration

Notes:

- For combo chart, the graph properties will respectively apply to the corresponding chart types.
- A combo chart may contain two value axes, namely, the Y1 and Y2 axes. The properties of the Y1 or Y2 axis are the same as those of the Y axis.

The following are the properties of the two child objects of the chart paper in an org chart:

- Org chart node
- Org chart line

The following are the properties of the two child objects of the chart paper in a heat map:

- Heat map rectangle
- Heat map rectangle title

Org chart node

The properties of each node in an org chart are:

Property Name	Description
Geometry	
Height	Specifies the height of the object, in inches or centimeters. Enter a numeric value to change the height. Data type: Float
Width	Specifies the width of the object, in inches or centimeters. Enter a numeric value to change the width. Data type: Float
Color	
Background	Specifies the background color of the object. Choose a color from the drop-down list, enter a hexadecimal RGB value (for example, 0xff0000), or select Custom from the drop-down list to specify the color. Data type: String
CSS	
Class	Specifies a CSS class to be applied to the object which is a valid class in the CSS file. Data type: String
Style	The property can be used in two ways. • Specifies a style to be applied to the object. Choose a style from the drop-down list which is available when a Style Group has been selected at the report tab level and when there are styles in the Style Group that can be applied to the object, or type in the style name. • Specifies a CSS selector to be applied to the object. Enter a valid class name from the CSS file. Data type: String
Others	
Minimum Leaf Distance	Specifies the minimum distance between two adjacent leaf nodes on the same level. For top-down and bottom-up layouts, it means horizontal distance. For left-right layout, it means vertical distance. Enter a numeric value to specify the distance. Data type: Float
Minimum Level Distance	Specifies the minimum distance between two adjacent levels. For top-down and bottom-up layouts, it means vertical distance. For left-right layout, it means horizontal distance. At runtime, if the vlaue of Minimum Level Distance is less than that of Line Thickness on org chart line, the former will be overridden by the latter. Data type: Float
Record Location	Specifies the calculation point for the properties which use formulas. Choose an option from the drop-down list. • **Default** - The values of the properties are calculated in the default location where they are located. • **Page Header** - The values of the properties are calculated in the page header. • **Page Footer** - The values of the properties are calculated in the page footer. Data type: Enumeration
Vertical Alignment	Specifies the vertical justification of the objects in each node. Choose an option from the drop-down list. Data type: Enumeration
Border	
Border Color	Specifies the color of the border of the object. Choose a color from the drop-down list, enter a hexadecimal RGB value (for example, 0xff0000), or select Custom from the drop-down list to specify the color. Data type: String
Border Thickness	Specifies the width of the border in inches or centimeters. Enter a numeric value to change the thickness. Data type: Float
Bottom Line	Specifies the line style of the bottom border of the object. Choose a style from the drop-down list. Data type: Enumeration
Left Line	Specifies the line style of the left border of the object. Choose a style from the drop-down list. Data type: Enumeration
Right Line	Specifies the line style of the right border of the object. Choose a style from the drop-down list. Data type: Enumeration
Top Line	Specifies the line style of the top border of the object. Choose a style from the drop-down list. Data type: Enumeration

Org chart line

The properties of the line object in an org chart are:

Property Name	Description
General	
Class Type	Indicates the class type of the object. This property is read only.
Instance Name	Shows the instance name of the object. This property is read only.
Line Property	
Line Color	Specifies the color of the line. Choose a color from the drop-down list, enter a hexadecimal RGB value (for example, 0xff0000), or select Custom from the drop-down list to specify the color and fill effect. Data type: String
Line Style	Specifies the style of the line. Choose an option from the drop-down list. Data type: Enumeration
Line Thickness	Specifies the thickness of the line. Enter a numeric value to change the thickness. Data type: Float

Heat map rectangle

The properties of the rectangles in a heat map are:

Property	Description
CSS	
Class	Specifies a CSS class to be applied to the object which is a valid class in the CSS file. Data type: String
Style	The property can be used in two ways. • Specifies a style to be applied to the object. Choose a style from the drop-down list which is available when a Style Group has been selected at the report tab level and when there are styles in the Style Group that can be applied to the object, or type in the style name. • Specifies a CSS selector to be applied to the object. Enter a valid class name from the CSS file. Data type: String
Others	
Link	Specifies to link the object to another report, a website, an e-mail address or a Blob data type field. Click ... in the value cell to set the link target. For details, see Binding a link to a label. Data type: String
Vertical Alignment	Specifies the vertical justification of the text in the object. Choose an option from the drop-down list. Data type: Enumeration
Separator	
Color	Specifies the color of the separator line. Choose a color from the drop-down list, enter a hexadecimal RGB value (for example, 0xff0000), or select Custom from the drop-down list to specify the color. Data type: String
Line Style	Specifies the line style of separator. Choose an option from the drop-down list. Data type: Enumeration
Thickness	Specifies the line width of the separator, in inches or centimeters. Enter a numeric value to change the thickness. Data type: Float
Transparency	Specifies the transparency of separators, in percent. Enter a numeric value to change the transparency. Data type: Integer
TOC	
Anchor Display Value	Specifies a string or formula to display for the TOC entry for this object in the TOC Browser. By default it is the value of the specified object. The TOC Anchor property must be set to true for this property to take effect. Data type: String
TOC Anchor	Specifies whether to include the object in the TOC Browser for the report. Data type: Boolean

Heat map rectangle title

The properties of the rectangle titles in a heat map are:

Text Format	
Bold	Specifies whether to make the text bold. Data type: Boolean
Font Face	Specifies the font of the text. Choose an option from the drop-down list. Data type: Enumeration
Font Size	Specifies the font size of the text. Enter an integer value to change the size. Data type: Integer
Format	Indicates the data format depending on the data type of the group field.
Horizontal Alignment	Specifies the horizontal justification of the text in the object. Choose an option from the drop-down list. Data type: Enumeration
Italic	Specifies whether to make the text italic. Data type: Boolean
Strikethrough	Specifies whether to add a strikethrough line to the text. Data type: Boolean
Underline	Specifies whether to underline the text. Data type: Boolean
Vertical Alignment	Specifies the vertical justification of the text in the object. Choose an option from the drop-down list. Data type: Enumeration
Word Wrap	Specifies whether to wrap the text to the width. Data type: Boolean
Geometry	
Height	Specifies the height of the object, in inches or centimeters. Enter a numeric value to change the height. Enabled when it is a horizontal banded object. Data type: Float
Color	
Background	Specifies the background color of the object. Choose a color from the drop-down list, enter a hexadecimal RGB value (for example, 0xff0000), or select Custom from the drop-down list to specify the color. Data type: String
Foreground	Specifies the foreground color of the object. Choose a color from the drop-down list, enter a hexadecimal RGB value (for example, 0xff0000), or select Custom from the drop-down list to specify the color. Data type: String
CSS	
Class	Specifies a CSS class to be applied to the object which is a valid class in the CSS file. Data type: String
ID	Specifies the identifier of the object, which must be unique in the report. The ID property can be a style sheet selector. Data type: String
Style	The property can be used in two ways. • Specifies a style to be applied to the object. Choose a style from the drop-down list which is available when a Style Group has been selected at the report tab level and when there are styles in the Style Group that can be applied to the object, or type in the style name. • Specifies a CSS selector to be applied to the object. Enter a valid class name from the CSS file. Data type: String
Padding	
Bottom Padding	Specifies the space in inches or centimeters between the text and the bottom border of the object. Enter a numeric value to change the padding. Data type: Float
Left Padding	Specifies the space in inches or centimeters between the text and the left border of the object. Enter a numeric value to change the padding. Data type: Float
Right Padding	Specifies the space in inches or centimeters between the text and the right border of the object. Enter a numeric value to change the padding. Data type: Float
Top Padding	Specifies the space in inches or centimeters between the text and the top border of the object. Enter a numeric value to change the padding. Data type: Float
Border	
Border Color	Specifies the color of the border of the object. Choose a color from the drop-down list, enter a hexadecimal RGB value (for example, 0xff0000), or select Custom from the drop-down list to specify the color. Data type: String
Border Thickness	Specifies the width of the border in inches or centimeters. Enter a numeric value to change the thickness. Data type: Float
Bottom Line	Specifies the line style of the bottom border of the object. Choose a style from the drop-down list. Data type: Enumeration
Left Line	Specifies the line style of the left border of the object. Choose a style from the drop-down list. Data type: Enumeration
Right Line	Specifies the line style of the right border of the object. Choose a style from the drop-down list. Data type: Enumeration
Shadow	Specifies whether to draw two shadowed borders, beneath and to the right of the object. Data type: Boolean
Shadow Color	Specifies the color of the shadow. Choose a color from the drop-down list, enter a hexadecimal RGB value (for example, 0xff0000), or select Custom from the drop-down list to specify the color and fill effect. Data type: String
Top Line	Specifies the line style of the top border of the object. Choose a style from the drop-down list. Data type: Enumeration
Pattern	
Pattern Color	Specifies the color in which to draw a pattern to fill the object. Choose a color from the drop-down list, enter a hexadecimal RGB value (for example, 0xff0000), or select Custom from the drop-down list to specify the color and fill effect. Data type: String
Pattern Style	Specifies the style of the pattern. Choose an option from the drop-down list. • **none** - No pattern will be applied to the object. • **50%** - The object will be filled in the Pattern Color of 50%-transparency. • **horizontal** - The pattern will be made of horizontal lines of the Pattern Color. • **vertical** - The pattern will be made of vertical lines of the Pattern Color. • **grid** - The pattern will be an overlapping of horizontal and vertical lines of the Pattern Color. • **diagonal** - The pattern will be made of diagonal lines of the Pattern Color. Data type: Enumeration
Others	
Invisible	Specifies whether to show the object in the design area and in the report results. All formulas and calculations will still be performed if the property is set to true. Data type: Boolean

Chart legend

The properties of a chart legend are:

Property	Description
Geometry	
Height	Specifies the height of the object, in inches or centimeters. Enter a numeric value to change the height. Data type: Float
Width	Specifies the width of the object, in inches or centimeters. Enter a numeric value to change the width. Data type: Float
CSS	
Class	Specifies a CSS class to be applied to the object which is a valid class in the CSS file. Data type: String
Style	The property can be used in two ways. • Specifies a style to be applied to the object. Choose a style from the drop-down list which is available when a Style Group has been selected at the report tab level and when there are styles in the Style Group that can be applied to the object, or type in the style name. • Specifies a CSS selector to be applied to the object. Enter a valid class name from the CSS file. Data type: String
Legend	
Auto Size	Specifies whether to resize the legend automatically. Data type: Boolean
Bottom Margin	Specifies the distance in inches or centimeters between the legend labels and the bottom border of the legend. Enter a numeric value to change the margin. Data type: Float
Data Mapping File	Specifies the data mapping file to the object for NLS use. For example, if the data mapping file name is Category_de_DE.properties, set the value as Category (without the language and locale part). Data type: String
Label Format Source	Specifies the data format option for the chart legend labels. Choose an option from the drop-down list. • **Axis Data Format** - If selected, the data format of the legend labels will follow that of the axis data. If the chart has series axis, the legend label format will be the same as the series axis data format; if not, the category axis data format will be used in the legend labels. • **Legend Data Format** - If selected, the data format of the legend labels will be controlled by the property Legend Data Format. Unavailable to heat maps that are colored only by numerics. Data type: Enumeration
Label Horizontal Spacing	Specifies the horizontal distance between two adjacent legend labels. Enter a numeric value to change the distance. Unavailable to heat maps that are colored only by numerics. Data type: Float
Label Vertical Spacing	Specifies the vertical distance between two adjacent legend labels. Enter a numeric value to change the distance. Unavailable to heat maps that are colored only by numerics. Data type: Float
Left Margin	Specifies the distance in inches or centimeters between the legend labels and the left border of the legend. Enter a numeric value to change the margin. Data type: Float
Legend Data Format	Specifies the data format of the legend labels, applied when Label Format Source is set to Legend Data Format. Click ... in the value cell to specify the format in the Data Format dialog. Data type: String
Placement	Specifies the position of the legend in the platform. Choose an option from the drop-down list. Data type: Enumeration
Right Margin	Specifies the distance in inches or centimeters between the legend labels and the right border of the legend. Enter a numeric value to change the margin. Data type: Float
Secondary Placement	Specifies the position of the legend further based on the setting of the property Placement. Choose an option from the drop-down list.
Show Scrollbar	Specifies whether to show a scrollbar on the legend to fully view the legend content when the content does not fit into the legend. Data type: Boolean
Show Tips	Specifies whether to show the corresponding data information when the mouse pointer points at a target in the chart legend in Designer view mode, in HTML result, or at server runtime. Data type: Boolean
Top Margin	Specifies the distance in inches or centimeters between the legend labels and the top border of the legend. Enter a numeric value to change the margin. Data type: Float
Truncate	Specifies whether to truncate the legend entry label text when the text overflow the labels. Data type: Boolean
Background	
Border Color	Specifies the color of the legend border. Choose a color from the drop-down list, enter a hexadecimal RGB value (for example, 0xff0000), or select Custom from the drop-down list to specify the color. Data type: String
Border End Caps	Specifies the ending style of the legend border line. Choose an option from the drop-down list. Data type: Enumeration
Border Joint	Specifies the joint style of the legend border line. Choose an option from the drop-down list. Data type: Enumeration
Border Outlined	Specifies whether to show the legend border line in Outline Path. If the property is set to true, the legend border line will be shown in Outline Path; otherwise, in Fill Path. Data type: Boolean
Border Style	Specifies the line style of the legend border. Choose an option from the drop-down list. Data type: Enumeration
Border Thickness	Specifies the width of the legend border, in inches or centimeters. Enter a numeric value to change the thickness. Data type: Float
Border Transparency	Specifies the transparency of the legend border, in percent. Enter a numeric value to change the transparency. Data type: Integer
Border Type	Specifies the border type of the legend. Choose an option from the drop-down list. Data type: Enumeration
Border Variable Dash	Specifies whether to resize the dash automatically. • **true** - If selected, the dash size will be adjusted automatically, and the option Auto Adjust Dash in the Format Legend dialog will be effect. • **false** - If selected, the dash size will be of fixed size, and the option Fixed Dash Size in the Format Legend dialog will take effect. Data type: Boolean
Color	Specifies the color of the legend background, applied when the property Fill Type is set to color. Choose a color from the drop-down list, enter a hexadecimal RGB value (for example, 0xff0000), or select Custom from the drop-down list to specify the color. Data type: String
Fill Transparency	Specifies the legend fill transparency, in percent. Enter a numeric value to change the transparency. Data type: Integer
Fill Type	Specifies the fill pattern for the legend background. Choose an option from the drop-down list. Data type: Enumeration
Gradient End Color	Specifies the color of the point where the gradient ends, applied when the property Fill Type is set to gradient. Choose a color from the drop-down list, enter a hexadecimal RGB value (for example, 0xff0000), or select Custom from the drop-down list to specify the color. Data type: String
Gradient End X	Specifies the horizontal position where the gradient ends, measured in a percentage of the legend width. Applied when the property Fill Type is set to gradient. Data type: Integer
Gradient End Y	Specifies the vertical position where the gradient ends, measured in a percentage of the legend height. Applied when the property Fill Type is set to gradient. Data type: Integer
Gradient Start Color	Specifies the color of the point where the gradient begins, applied when the property Fill Type is set to gradient. Choose a color from the drop-down list, enter a hexadecimal RGB value (for example, 0xff0000), or select Custom from the drop-down list to specify the color. Data type: String
Gradient Start X	Specifies the horizontal position where the gradient begins, measured in a percentage of the legend width. Applied when the property Fill Type is set to gradient. Data type: Integer
Gradient Start Y	Specifies the vertical position where the gradient begins, measured in a percentage of the legend height. Applied when the property Fill Type is set to gradient. Data type: Integer
Gradient Style	Specifies the gradient style for the legend background, applied when the property Fill Type is set to gradient. Choose an option from the drop-down list. Data type: Enumeration
Image File	Specifies the name of the image, a portion of which will be displayed as the legend background. Applied when the property Fill Type is set to image. Enter the file name with suffix in the value cell. If the image is outside of the current catalog, you should include the full path of the image correctly. Data type: String
Image Height	The four properties, Image X, Image Y, Image Width, and Image Height, together are used to define a portion of the image (the portion is a rectangle), which will be displayed as the legend background. They take effect when the property Fill Type is set to image. Image Height specifies the height of the image portion, measured in a percentage of the image height. Data type: Integer
Image Layout	Specifies the layout style for the image that will be displayed as the legend background, applied when the property Fill Type is set to image. Choose an option from the drop-down list. Data type: Enumeration
Image Width	The four properties, Image X, Image Y, Image Width, and Image Height, together are used to define a portion of the image (the portion is a rectangle), which will be displayed as the legend background. They take effect when the property Fill Type is set to image. Image Width specifies the width of the image portion, measured in a percentage of the image width. Data type: Integer
Image X	The four properties, Image X, Image Y, Image Width, and Image Height, together are used to define a portion of the image (the portion is a rectangle), which will be displayed as the legend background. They take effect when the property Fill Type is set to image. Image X specifies the distance between the left border of image and the portion that will contain the pattern, measured in a percentage of the image width. Data type: Integer
Image Y	The four properties, Image X, Image Y, Image Width, and Image Height, together are used to define a portion of the image (the portion is a rectangle), which will be displayed as the legend background. They take effect when the property Fill Type is set to image. Image Y specifies the distance between the top border of image and the portion that will contain the pattern, measured in a percentage of the image height. Data type: Integer
Inset Bottom	Specifies the bottom position of the background area, measured in a percentage of the legend height, from the bottom edge of the legend. Data type: Float
Inset Left	Specifies the left position of the background area, measured in a percentage of the legend width, from the left edge of the legend. Data type: Float
Inset Right	Specifies the right position of the background area, measured in a percentage of the legend width, from the right edge of the legend. Data type: Float
Inset Top	Specifies the top position of the background area, measured in a percentage of the legend height, from the top edge of the legend. Data type: Float
Texture Background Color	Specifies the background color of the texture, applied when the property Fill Type is set to texture. Choose a color from the drop-down list, enter a hexadecimal RGB value (for example, 0xff0000), or select Custom from the drop-down list to specify the color. Data type: String
Texture Foreground Color	Specifies the foreground color of the texture, applied when the property Fill Type is set to texture. Choose a color from the drop-down list, enter a hexadecimal RGB value (for example, 0xff0000), or select Custom from the drop-down list to specify the color. Data type: String
Texture Style	Specifies the texture style of the legend background, applied when the property Fill Type is set to texture. Choose an option from the drop-down list. Data type: Enumeration
Label	
Color	Specifies the color of the label, applied when the property Fill Type is set to color. Choose a color from the drop-down list, enter a hexadecimal RGB value (for example, 0xff0000), or select Custom from the drop-down list to specify the color. Data type: String
Fill Transparency	Specifies the transparency of the background, in percent. Enter a numeric value to change the transparency. Data type: Integer
Fill Type	Specifies the fill pattern for the label. Choose an option from the drop-down list. Data type: Enumeration
Font Effect	Specifies the special effect for the label text. Choose an option from the drop-down list. Data type: Enumeration
Font Face	Specifies the font face for the legend labels. Choose an option from the drop-down list. Data type: Enumeration
Font Rotation	Specifies the rotation angle of each legend label around its center, in degrees. The default value is 0. Enter a numeric value to change the rotation. Data type: Float
Font Script	Specifies whether the label text will in superscript or subscript form, or neither of them. Choose an option from the drop-down list. Data type: Enumeration
Font Shearing	Specifies the shearing transformation of each legend label around its center. The default value is 0. Enter a numeric value to change the shearing. Data type: Float
Font Size	Specifies the font size for the legend labels. Enter an integer value to change the size. Data type: Integer
Font Strikethrough	Specifies the style of the line by which the label texts are struck through. Choose an option from the drop-down list. Data type: Enumeration
Font Style	Specifies the font style for the legend labels. Choose an option from the drop-down list. Data type: Enumeration
Font Underline	Specifies the underline style for the legend labels. Choose an option from the drop-down list. Data type: Enumeration
Gradient End Color	Specifies the color of the point where the gradient ends, applied when the property Fill Type is set to gradient. Choose a color from the drop-down list, enter a hexadecimal RGB value (for example, 0xff0000), or select Custom from the drop-down list to specify the color. Data type: String
Gradient End X	Specifies the horizontal position where the gradient ends, measured in a percentage of the label text width. Applied when the property Fill Type is set to gradient. Data type: Integer
Gradient End Y	Specifies the vertical position where the gradient ends, measured in a percentage of the label text height. Applied when the property Fill Type is set to gradient. Data type: Integer
Gradient Start Color	Specifies the color of the point where the gradient begins, applied when the property Fill Type is set to gradient. Choose a color from the drop-down list, enter a hexadecimal RGB value (for example, 0xff0000), or select Custom from the drop-down list to specify the color. Data type: String
Gradient Start X	Specifies the horizontal position where the gradient begins, measured in a percentage of the label text width. Applied when the property Fill Type is set to gradient. Data type: Integer
Gradient Start Y	Specifies the vertical position where the gradient begins, measured in a percentage of the label text height. Applied when the property Fill Type is set to gradient. Data type: Integer
Gradient Style	Specifies the gradient style for the label, applied when the property Fill Type is set to gradient. Choose an option from the drop-down list. Choose an option from the drop-down list. Data type: Enumeration
Outline Color	Specifies the color for the label text outline. Choose a color from the drop-down list, enter a hexadecimal RGB value (for example, 0xff0000), or select Custom from the drop-down list to specify the color. Data type: String
Outline End Caps	Specifies the ending style for the label text outline. Choose an option from the drop-down list. Data type: Enumeration
Outline Joint	Specifies the joint style for the label text outline. Choose an option from the drop-down list. Data type: Enumeration
Outline Outlined	Specifies whether to show the label text outline in outline form. If the property is set to true, the outline will be shown in outline form; otherwise, in whole form. Data type: Boolean
Outline Style	Specifies the style for the label text outline. Choose an option from the drop-down list. Data type: Enumeration
Outline Thickness	Specifies the thickness for the label text outline. Enter a numeric value to change the thickness. Data type: Float
Outline Transparency	Specifies the transparency for the label text outline, in percent. Enter a numeric value to change the transparency. Data type: Integer
Outline Variable Dash	Specifies whether to resize the dash automatically. If the property is set to true, the dash size will be adjusted automatically; otherwise, the dash size will be of fixed size. Data type: Boolean
Percent Format	Specifies the format of the percentage values on the legend. Applied when Show Percent is set to true. Click ... in the value cell to specify the format in the Data Format dialog. Data type: String
Reverse Labels	Specifies whether to reverse the order of the labels in the legend object. Data type: Boolean
Show Percent	Specifies whether to show the percentage of each entry value to the total on the legend. Applies to pie, bar and bench charts that have only one value axis. Data type: Boolean
Show Values	Specifies whether to show the values for the legend entries. Applies to pie, bar and bench charts that have only one value axis. Data type: Boolean
Text of Total Value	Specifies the text of the total label on the legend. Applies to pie, bar and bench charts that have only one value axis. Data type: String
Texture Background Color	Specifies the background color of the texture, applied when the property Fill Type is set to texture. Choose a color from the drop-down list, enter a hexadecimal RGB value (for example, 0xff0000), or select Custom from the drop-down list to specify the color. Data type: String
Texture Foreground Color	Specifies the foreground color of the texture, applied when the property Fill Type is set to texture. Choose a color from the drop-down list, enter a hexadecimal RGB value (for example, 0xff0000), or select Custom from the drop-down list to specify the color. Data type: String
Texture Style	Specifies the texture style for the label, applied when the property Fill Type is set to texture. Choose an option from the drop-down list. Data type: Enumeration
Value Format	Specifies the number format of the legend entry values. Applied when Show Values is set to true. Click ... in the value cell to specify the format in the Data Format dialog. Data type: String
Word Wrap	Specifies whether to enable the word wrap function for the label text; not applying to radar charts, bubble charts, scatter charts and gauge charts. If the text contains special characters (such as ":", " " and space), it will be broken at the position of one of the special characters. Data type: Boolean
Icon	
Border Color	Specifies the color of the icon borders. Choose a color from the drop-down list, enter a hexadecimal RGB value (for example, 0xff0000), or select Custom from the drop-down list to specify the color. Data type: String
Border End Caps	Specifies the ending style of the icon border line. Choose an option from the drop-down list. Data type: Enumeration
Border Joint	Specifies the joint style of the icon border line. Choose an option from the drop-down list. Data type: Enumeration
Border Outlined	Specifies whether to show the icon border line in Outline Path. If the property is set to true, the icon border line will be shown in Outline Path; otherwise, in Fill Path. Data type: Boolean
Border Style	Specifies the line style of the icon borders. Choose an option from the drop-down list. Data type: Enumeration
Border Thickness	Specifies the width of the icon borders in inches or centimeters. Enter a numeric value to change the thickness. Data type: Float
Border Transparency	Specifies the transparency of the icon borders, in percent. Enter a numeric value to change the transparency. Data type: Integer
Height	Specifies the height of the icon, in inches or centimeters. Enter a numeric value to change the width. Data type: Float
Icon Alignment	Specifies the relative position of the icon to the legend. Choose an option from the drop-down list. Data type: Enumeration
Icon Text Gap	Specifies the distance between each legend label and icon. Enter a numeric value to change the gap. Data type: Float
Width	Specifies the width of the icon, in inches or centimeters. Enter a numeric value to change the width. Data type: Float

Chart label

The properties of a chart label are:

Property Name	Description
Geometry	
Height	Specifies the height of the object, in inches or centimeters. Enter a numeric value to change the height. Data type: Float
Width	Specifies the width of the object, in inches or centimeters. Enter a numeric value to change the width. Data type: Float
CSS	
Class	Specifies a CSS class to be applied to the object which is a valid class in the CSS file. Data type: String
Style	The property can be used in two ways. • Specifies a style to be applied to the object. Choose a style from the drop-down list which is available when a Style Group has been selected at the report tab level and when there are styles in the Style Group that can be applied to the object, or type in the style name. • Specifies a CSS selector to be applied to the object. Enter a valid class name from the CSS file. Data type: String
Label	
Bottom Margin	Specifies the distance in inches or centimeters between the label text and the bottom border of the label. The four properties, Bottom Margin, Left Margin, Right Margin and Top Margin work together with the property Text Alignment. Enter a numeric value to change the margin. Data type: Float
Color	Specifies the color of the label, applied when the property Fill Type is set to color. Choose a color from the drop-down list, enter a hexadecimal RGB value (for example, 0xff0000), or select Custom from the drop-down list to specify the color. Data type: String
Fill Transparency	Specifies the transparency of the background, in percent. Enter a numeric value to change the transparency. Data type: Integer
Fill Type	Specifies the fill pattern for the label. Choose an option from the drop-down list. Data type: Integer
Font Effect	Specifies the special effect for the label text. Choose an option from the drop-down list. Data type: Enumeration
Font Face	Specifies the font face of the label text. Choose an option from the drop-down list. Data type: Enumeration
Font Rotation	Specifies the rotation angle of the label text around its center, in degrees. The default value is 0. Enter a numeric value to change the rotation. Data type: Float
Font Script	Specifies whether the label text will be in superscript or subscript form, or neither of them. Choose an option from the drop-down list. Data type: Enumeration
Font Shearing	Specifies the shearing transformation of the label text around its center. The default value is 0. Enter a numeric value to change the shearing. Data type: Float
Font Size	Specifies the font size of the label text. Enter an integer value to change the size. Data type: Integer
Font Strikethrough	Specifies the style of the line by which the label text is struck through. Choose an option from the drop-down list. Data type: Enumeration
Font Style	Specifies the font style for the label text. Choose an option from the drop-down list. Data type: Enumeration
Font Underline	Specifies the underline style for the label text. Choose an option from the drop-down list. Data type: Enumeration
Gradient End Color	Specifies the color of the point where the gradient ends, applied when the property Fill Type is set to gradient. Choose a color from the drop-down list, enter a hexadecimal RGB value (for example, 0xff0000), or select Custom from the drop-down list to specify the color. Data type: String
Gradient End X	Specifies the horizontal position where the gradient ends, measured in a percentage of the label text width. Applied when the property Fill Type is set to gradient. Data type: Integer
Gradient End Y	Specifies the vertical position where the gradient ends, measured in a percentage of the label text height. Applied when the property Fill Type is set to gradient. Data type: Integer
Gradient Start Color	Specifies the color of the point where the gradient begins, applied when the property Fill Type is set to gradient. Choose a color from the drop-down list, enter a hexadecimal RGB value (for example, 0xff0000), or select Custom from the drop-down list to specify the color. Data type: String
Gradient Start X	Specifies the horizontal position where the gradient begins, measured in a percentage of the label text width. Applied when the property Fill Type is set to gradient. Data type: Integer
Gradient Start Y	Specifies the vertical position where the gradient begins, measured in a percentage of the label text height. Applied when the property Fill Type is set to gradient. Data type: Integer
Gradient Style	Specifies the gradient style for the label, applied when the property Fill Type is set to gradient. Choose an option from the drop-down list. Data type: Enumeration
Left Margin	Specifies the distance in inches or centimeters between the label text and the left border of the label. The four properties, Bottom Margin, Left Margin, Right Margin and Top Margin work together with the property Text Alignment. Enter a numeric value to change the margin. Data type: Float
Outline Color	Specifies the color for the label text outline. Choose a color from the drop-down list, enter a hexadecimal RGB value (for example, 0xff0000), or select Custom from the drop-down list to specify the color. Data type: String
Outline End Caps	Specifies the ending style for the label text outline. Choose an option from the drop-down list. Data type: Integer
Outline Joint	Specifies the joint style for the label text outline. Choose an option from the drop-down list. Data type: Enumeration
Outline Outlined	Specifies whether to show the label text outline in outline form. If the property is set to true, the outline will be shown in outline form; otherwise, in whole form. Data type: Boolean
Outline Style	Specifies the style for the label text outline. Choose an option from the drop-down list. Data type: Enumeration
Outline Thickness	Specifies the thickness for the label text outline. Data type: Float
Outline Transparency	Specifies the transparency for the label text outline, in percent. Enter a numeric value to change the transparency. Data type: Integer
Outline Variable Dash	Specifies whether to resize the dash automatically. If the property is set to true, the dash size will be adjusted automatically; otherwise, the dash size will be of fixed size. Data type: Boolean
Right Margin	Specifies the distance in inches or centimeters between the label text and the right border of the label. The four properties, Bottom Margin, Left Margin, Right Margin and Top Margin work together with the property Text Alignment. Enter a numeric value to change the margin. Data type: Float
Text	Specifies the text of the label. Enter a string to change the text. Data type: String
Text Alignment	Specifies the alignment style of the label text. It works together with the four properties, Bottom Margin, Left Margin, Right Margin and Top Margin. Choose an option from the drop-down list. Data type: Enumeration
Texture Background Color	Specifies the background color of the texture, applied when the property Fill Type is set to texture. Choose a color from the drop-down list, enter a hexadecimal RGB value (for example, 0xff0000), or select Custom from the drop-down list to specify the color. Data type: String
Texture Foreground Color	Specifies the foreground color of the texture, applied when the property Fill Type is set to texture. Choose a color from the drop-down list, enter a hexadecimal RGB value (for example, 0xff0000), or select Custom from the drop-down list to specify the color. Data type: String
Texture Style	Specifies the texture style for the label, applied when the property Fill Type is set to texture. Choose an option from the drop-down list. Data type: Enumeration
Top Margin	Specifies the distance in inches or centimeters between the label text and the top border of the label. The four properties, Bottom Margin, Left Margin, Right Margin and Top Margin work together with the property Text Alignment. Enter a numeric value to change the margin. Data type: Float
Word Wrap	Specifies whether to enable the word wrap function for the label text, not applying to radar charts, bubble charts, scatter charts and gauge charts. If the text contains special characters (such as ",", "." and space), it will be broken at the position of one of the special characters. Data type: Boolean
Background	
Border Color	Specifies the color of the label border. Choose a color from the drop-down list, enter a hexadecimal RGB value (0xCCAA00), or select Custom from the drop-down list to specify the color. Data type: String
Border End Caps	Specifies the ending style of the label border line. Choose an option from the drop-down list. Data type: Enumeration
Border Joint	Specifies the joint style of the label border line. Choose an option from the drop-down list. Data type: Enumeration
Border Outlined	Specifies whether to show the label border line in Outline Path. If the property is set to true, the label border line will be shown in Outline Path; otherwise, in Fill Path. Data type: Boolean
Border Style	Specifies the line style of the label border. Choose an option from the drop-down list. Data type: Enumeration
Border Thickness	Specifies the width of the label border, in inches or centimeters. Enter a numeric value to change the thickness. Data type: Float
Border Transparency	Specifies the transparency of the label border, in percent. Enter a numeric value to change the transparency. Data type: Integer
Border Type	Specifies the border type of the label. Choose an option from the drop-down list. Data type: Enumeration
Border Variable Dash	Specifies whether to resize the dash automatically. • **true** - If selected, the dash size will be adjusted automatically, and the option Auto Adjust Dash in the Format Label dialog will take effect. • **false** - If selected, the dash size will be of fixed size, and the option Fixed Dash Size in the Format Label dialog will take effect. Data type: Boolean
Color	Specifies the color of the label background, applied when the property Fill Type is set to color. Choose a color from the drop-down list, enter a hexadecimal RGB value (for example, 0xff0000), or select Custom from the drop-down list to specify the color. Data type: String
Fill Transparency	Specifies the transparency of the label background, in percent. Enter a numeric value to change the transparency. Data type: Integer
Fill Type	Specifies the fill pattern for the label background. Choose an option from the drop-down list. Data type: Enumeration
Gradient End Color	Specifies the color of the point where the gradient ends, applied when the property Fill Type is set to gradient. Choose a color from the drop-down list, enter a hexadecimal RGB value (for example, 0xff0000), or select Custom from the drop-down list to specify the color. Data type: String
Gradient End X	Specifies the horizontal position where the gradient ends, measured in a percentage of the label width. Applied when the property Fill Type is set to gradient. Data type: Integer
Gradient End Y	Specifies the vertical position where the gradient ends, measured in a percentage of the label height. Applied when the property Fill Type is set to gradient. Data type: Integer
Gradient Start Color	Specifies the color of the point where the gradient begins, applied when the property Fill Type is set to gradient. Choose a color from the drop-down list, enter a hexadecimal RGB value (for example, 0xff0000), or select Custom from the drop-down list to specify the color. Data type: String
Gradient Start X	Specifies the horizontal position where the gradient begins, measured in a percentage of the label width. Applied when the property Fill Type is set to gradient. Data type: Integer
Gradient Start Y	Specifies the vertical position where the gradient begins, measured in a percentage of the label height. Applied when the property Fill Type is set to gradient. Data type: Integer
Gradient Style	Specifies the gradient style for the label background, applied when the property Fill Type is set to gradient. Choose an option from the drop-down list. Data type: Enumeration
Image File	Specifies the file name of the image, a portion of which will be displayed as the label background. Applied when the property Fill Type is set to image. Enter the file name with suffix in the value cell. If the image is outside of the current catalog, you should include the full path of the image correctly. Data type: String
Image Height	The four properties, Image X, Image Y, Image Width, and Image Height, together are used to define a portion of the image (the portion is a rectangle), which will be displayed as the label background. They take effect when the property Fill Type is set to image. Image Height specifies the height of the image portion, measured in a percentage of the image height. Data type: Integer
Image Layout	Specifies the layout style for the image that will be displayed as the label background, applied when the property Fill Type is set to image. Choose an option from the drop-down list. Data type: Integer
Image Width	The four properties, Image X, Image Y, Image Width, and Image Height, together are used to define a portion of the image (the portion is a rectangle), which will be displayed as the label background. They take effect when the property Fill Type is set to image. Image Width specifies the width of the image portion, measured in a percentage of the image width. Data type: Integer
Image X	The four properties, Image X, Image Y, Image Width, and Image Height, together are used to define a portion of the image (the portion is a rectangle), which will be displayed as the label background. They take effect when the property Fill Type is set to image. Image X specifies the distance between the left border of image and the portion that will contain the pattern, measured in a percentage of the image width. Data type: Integer
Image Y	The four properties, Image X, Image Y, Image Width, and Image Height, together are used to define a portion of the image (the portion is a rectangle), which will be displayed as the label background. They take effect when the property Fill Type is set to image. Image Y specifies the distance between the top border of image and the portion that will contain the pattern, measured in a percentage of the image height. Data type: Integer
Inset Bottom	Specifies the bottom position of the background area, measured in a percentage of the label height, from the bottom edge of the label. Data type: Float
Inset Left	Specifies the left position of the background area, measured in a percentage of the label width, from the left edge of the label. Data type: Float
Inset Right	Specifies the right position of the background area, measured in a percentage of the label width, from the right edge of the label. Data type: Float
Inset Top	Specifies the top position of the background area, measured in a percentage of the label height, from the top edge of the label. Data type: Float
Texture Background Color	Specifies the background color of the texture, applied when the property Fill Type is set to texture. Choose a color from the drop-down list, enter a hexadecimal RGB value (for example, 0xff0000), or select Custom from the drop-down list to specify the color. Data type: String
Texture Foreground Color	Specifies the foreground color of the texture, applied when the property Fill Type is set to texture. Choose a color from the drop-down list, enter a hexadecimal RGB value (for example, 0xff0000), or select Custom from the drop-down list to specify the color. Data type: String
Texture Style	Specifies the texture style for the label background, applied when the property Fill Type is set to texture. Choose an option from the drop-down list. Data type: Enumeration
Icon	
Border Color	Specifies the color of the border of the object. Choose a color from the drop-down list, enter a hexadecimal RGB value (0xCCAA00), or select Custom from the drop-down list to specify the color. Data type: String
Border End Caps	Specifies the ending style of the icon border line. Choose an option from the drop-down list. Data type: Enumeration
Border Joint	Specifies the joint style of the icon border line. Choose an option from the drop-down list. Data type: Enumeration
Border Outlined	Specifies whether to show the icon border line in Outline Path. If the property is set to true, the icon border line will be shown in Outline Path; otherwise, in Fill Path. Data type: Boolean
Border Style	Specifies the line style of the icon borders. Choose an option from the drop-down list. Data type: Enumeration
Border Thickness	Specifies the width of the border in inches or centimeters. Enter a numeric value to change the thickness. Data type: Float
Border Transparency	Specifies the transparency of the icon borders, in percent. Enter a numeric value to change the transparency. Data type: Integer
Border Variable Dash	Specifies whether to resize the dash automatically. If the property is set to true, the dash size will be adjusted automatically; otherwise, the dash size will be of fixed size. Data type: Boolean
Color	Specifies the color of the icon background, applied when the property Fill Type is set to color. Choose a color from the drop-down list, enter a hexadecimal RGB value (for example, 0xff0000), or select Custom from the drop-down list to specify the color. Data type: String
Fill Transparency	Specifies the transparency of the icon background, in percent. Enter a numeric value to change the transparency. Data type: Integer
Fill Type	Specifies the fill pattern for the icon. Choose an option from the drop-down list. Data type: Enumeration
Gradient End Color	Specifies the color of the point where the gradient ends, applied when the property Fill Type is set to gradient. Choose a color from the drop-down list, enter a hexadecimal RGB value (for example, 0xff0000), or select Custom from the drop-down list to specify the color. Data type: String
Gradient End X	Specifies the horizontal position where the gradient ends, measured in a percentage of the icon width. Applied when the property Fill Type is set to gradient. Data type: Integer
Gradient End Y	Specifies the vertical position where the gradient ends, measured in a percentage of the icon height. Applied when the property Fill Type is set to gradient. Data type: Integer
Gradient Start Color	Specifies the color of the point where the gradient begins, applied when the property Fill Type is set to gradient. Choose a color from the drop-down list, enter a hexadecimal RGB value (for example, 0xff0000), or select Custom from the drop-down list to specify the color. Data type: String
Gradient Start X	Specifies the horizontal position where the gradient begins, measured in a percentage of the icon width. Applied when the property Fill Type is set to gradient. Data type: Integer
Gradient Start Y	Specifies the vertical position where the gradient begins, measured in a percentage of the icon height. Applied when the property Fill Type is set to gradient. Data type: Integer
Gradient Style	Specifies the gradient style for the icon, applied when the property Fill Type is set to gradient. Choose an option from the drop-down list. Data type: Enumeration
Height	Specifies the height of the icon, in inches or centimeters. Enter a numeric value to change the width. Data type: Float
Icon Alignment	Specifies the relative position of the icon to the label. Choose an option from the drop-down list. Data type: Enumeration
Icon Style	Specifies the style of the icon attached to the label. Choose an option from the drop-down list. Data type: Enumeration
Icon Text Gap	Specifies the distance between the icon and the label. Data type: Float
Image File	Specifies the file name of the image, a portion of which will be displayed as the icon background. Applied when the property Fill Type is set to image. Enter the file name with suffix in the value cell. If the image is outside of the current catalog, you should include the full path of the image correctly. Data type: String
Image Height	The four properties, Image X, Image Y, Image Width, and Image Height, together are used to define a portion of the image (the portion is a rectangle), which will be displayed as the icon background. They take effect when the property Fill Type is set to image. Image Height specifies the height of the image portion, measured in a percentage of the image height. Data type: Integer
Image Layout	Specifies the layout style for the image that will be displayed as the icon background, applied when the property Fill Type is set to image. Choose an option from the drop-down list. Data type: Enumeration
Image Width	The four properties, Image X, Image Y, Image Width, and Image Height, together are used to define a portion of the image (the portion is a rectangle), which will be displayed as the icon background. They take effect when the property Fill Type is set to image. Image Width specifies the width of the image portion, measured in a percentage of the image width. Data type: Integer
Image X	The four properties, Image X, Image Y, Image Width, and Image Height, together are used to define a portion of the image (the portion is a rectangle), which will be displayed as the icon background. They take effect when the property Fill Type is set to image. Image X specifies the distance between the left border of image and the portion that will contain the pattern, measured in a percentage of the image width. Data type: Integer
Image Y	The four properties, Image X, Image Y, Image Width, and Image Height, together are used to define a portion of the image (the portion is a rectangle), which will be displayed as the icon background. They take effect when the property Fill Type is set to image. Image Y specifies the distance between the top border of image and the portion that will contain the pattern, measured in a percentage of the image height. Data type: Integer
Texture Background Color	Specifies the background color of the texture, applied when the property Fill Type is set to texture. Choose a color from the drop-down list, enter a hexadecimal RGB value (for example, 0xff0000), or select Custom from the drop-down list to specify the color. Data type: String
Texture Foreground Color	Specifies the foreground color of the texture, applied when the property Fill Type is set to texture. Choose a color from the drop-down list, enter a hexadecimal RGB value (for example, 0xff0000), or select Custom from the drop-down list to specify the color. Data type: String
Texture Style	Specifies the texture style for the icon, applied when the property Fill Type is set to texture. Choose an option from the drop-down list. Data type: Enumeration
Width	Specifies the width of the icon, in inches or centimeters. Enter a numeric value to change the width. Data type: Float

Crosstab

The properties of a crosstab in a web report are:

Property Name	Description
Color	
Background	Specifies the background color of the object. Choose a color from the drop-down list, enter a hexadecimal RGB value (for example, 0xff0000), or select Custom from the drop-down list to specify the color. Data type: String
Alternating Line Color	
Direction	Specifies whether the color defined by the property Pattern List will apply to the fields in the same rows or columns in the crosstab. • **vertical** - The value of the property Pattern List will take effect on the fields in the same columns, except for the total columns. • **horizontal** - The value of the property Pattern List will take effect on the fields in the same rows, except for the total rows. Data type: Enumeration
Enable	Specifies whether to enable setting alternating line color for the crosstab. Only when it is set to true, the settings for the properties Direction and Pattern List can take effect. Data type: Boolean **Note:** The setting of the alternating line color function for crosstab can only take effect when all the aggregate fields in the crosstab have transparent background.
Pattern List	Specifies the color pattern for the fields in the same rows or columns in the crosstab. Click ... in the value cell and click the small squares in the color tray one by one to specify the patterns. The patterns can be one or more of the following: Automatic, Color, Texture, and Gradient. Data type: String
CSS	
Class	Specifies a CSS class to be applied to the object which is a valid class in the CSS file. Data type: String
Style	The property can be used in two ways. • Specifies a style to be applied to the object. Choose a style from the drop-down list which is available when a Style Group has been selected at the report tab level and when there are styles in the Style Group that can be applied to the object, or type in the style name. • Specifies a CSS selector to be applied to the object. Enter a valid class name from the CSS file. Data type: String
Others	
Default for Filter	Specifies whether to show the data component as the default component in the Apply to drop-down list of the Filter dialog in Web Report Studio. Data type: Boolean **Note:** In a web report, only one data component's Default for Filter property can be set to true.
Enable Navigation	Specifies whether to enable scrollbars on the crosstab when the parent container cannot fully display the data of the crosstab. When this property is set to true, you can further specify the height and width of the frame for holding the crosstab, by setting the two properties Navigation Height and Navigation Width. Data type: Boolean
Export to Applet	Specifies whether to include the object when exporting the report to Applet. Data type: Boolean
Export to CSV	Specifies whether to include the object when exporting the report to CSV. Data type: Boolean
Export to Excel	Specifies whether to include the object when exporting the report to Excel. Data type: Boolean
Export to HTML	Specifies whether to include the object when exporting the report to HTML. Data type: Boolean
Export to PDF	Specifies whether to include the object when exporting the report to PDF. Data type: Boolean
Export to PostScript	Specifies whether to include the object when exporting the report to PostScript. Data type: Boolean
Export to Report Result	Specifies whether to include the object when exporting the report to Report Result. Data type: Boolean
Export to RTF	Specifies whether to include the object when exporting the report to RTF. Data type: Boolean
Export to Text	Specifies whether to include the object when exporting the report to Text. Data type: Boolean
Export to XML	Specifies whether to include the object when exporting the report to XML. Data type: Boolean
Invisible	Specifies whether to show the object in the design area and in the report results. All formulas and calculations will still be performed if the property is set to true. Data type: Boolean
Invisible for Filter Dialogs	Specifies whether to show the data component in the Apply to drop-down list of the Filter dialog in Web Report Studio. This property cannot be edited when Default for Filter is set to true. Data type: Boolean
Navigation Height	Specifies the height of the frame for holding the crosstab. Enter a numeric value to specify the height. If no value is given here, JReport will calculate the height at runtime. This property takes effect only when Enable Navigation is set to true. Data type: Float
Navigation Width	Specifies the width of the frame for holding the crosstab. Enter a numeric value to specify the width. If no value is given here, JReport will calculate the width at runtime. This property takes effect only when Enable Navigation is set to true. Data type: Float
Record Location	Specifies the calculation point for the properties which use formulas. Choose an option from the drop-down list. • **Default** - The values of the properties are calculated in the default location where they are located. • **Page Header** - The values of the properties are calculated in the page header. • **Page Footer** - The values of the properties are calculated in the page footer. Data type: Enumeration
Suppress	Specifies whether to show the object in the design area and in the report results. All formulas and calculations will be skipped if the property is set to true. Data type: Boolean **Note:** When both the Invisible and Suppress properties of an object are set to true, Suppress has the higher priority.
Suppress When No Records	Specifies whether to display the object in the report results when no record is returned to its parent data container. Data type: Boolean
Excel	
Column Index	Specifies the X coordinate of the object relative to its parent container when exported to Excel or CSV, measured in cells. The Columned property at the report level must be set to true for this property to take effect. Data type: Integer
Row Index	Specifies the Y coordinate of the object relative to its parent container when exported to Excel or CSV, measured in cells. The Columned property at the report level must be set to true for this property to take effect. Data type: Integer
Border	
Border Color	Specifies the color of the border of the object. Choose a color from the drop-down list, enter a hexadecimal RGB value (for example, 0xff0000), or select Custom from the drop-down list to specify the color. Data type: String
Has Border	Specifies whether to show the cell borders. Data type: Boolean
Crosstab Property	
Boundary Value	Specifies the number of columns in one aggregate cell when the crosstab is displayed horizontally, or rows when displayed vertically. Data type: Integer
Column Totals on Left	Specifies whether to display the total columns on the left of the aggregates. Data type: Boolean
Horizontal Gap	Specifies the space between the left/right edge of a crosstab cell and the contents in it. Data type: Float
Row Totals on Top	Specifies whether to display the total rows on the top of the aggregates. Data type: Boolean
Suppress Column Header	Specifies whether to suppress the column header in view mode. Data type: Boolean
Suppress Row Header	Specifies whether to suppress the row header in view mode. Data type: Boolean
Use Table Style	Specifies whether to add headers to the Total rows and columns. Data type: Boolean
Vertical Gap	Specifies the space between the top/bottom edge of a crosstab cell and the contents in it. Data type: Float
Vertical Layout	Indicates whether the aggregate fields are displayed vertically. The property is read only.
TOC	
Anchor Display Value	Specifies a string or formula to display for the TOC entry for this object in the TOC Browser. By default it is the value of the specified object. The TOC Anchor property must be set to true for this property to take effect. Data type: String
TOC Anchor	Specifies whether to include the object in the TOC Browser for the report. Data type: Boolean

A crosstab has some child objects, the properties of which will be shown in the following topics:

- DBField
- Formula field
- Label

DBField

The properties of a DBField in a crosstab are:

Property Name	Description
General	
DBField Name	Shows the mapping name of the field in the catalog. This property is read only.
Geometry (only for aggregate field)	
Height	Specifies the height of the object, in inches or centimeters. Enter a numeric value to change the height. Data type: Float
Width	Specifies the width of the object, in inches or centimeters. Enter a numeric value to change the width. Data type: Float
Color	
Background	Specifies the background color of the object. Choose a color from the drop-down list, enter a hexadecimal RGB value (for example, 0xff0000), or select Custom from the drop-down list to specify the color. Data type: String
Foreground	Specifies the foreground color of the object. Choose a color from the drop-down list, enter a hexadecimal RGB value (for example, 0xff0000), or select Custom from the drop-down list to specify the color. Data type: String
CSS	
Class	Specifies a CSS class to be applied to the object which is a valid class in the CSS file. Data type: String
ID	Specifies the identifier of the object, which must be unique in the report. The ID property value can be a style sheet selector. Data type: String
Style	The property can be used in two ways. • Specifies a style to be applied to the object. Choose a style from the drop-down list which is available when a Style Group has been selected at the report tab level and when there are styles in the Style Group that can be applied to the object, or type in the style name. • Specifies a CSS selector to be applied to the object. Enter a valid class name from the CSS file. Data type: String
Padding	
Bottom Padding	Specifies the space in inches or centimeters between the text and the bottom border of the object. Enter a numeric value to change the padding. Data type: Float
Left Padding	Specifies the space in inches or centimeters between the text and the left border of the object. Enter a numeric value to change the padding. Data type: Float
Right Padding	Specifies the space in inches or centimeters between the text and the right border of the object. Enter a numeric value to change the padding. Data type: Float
Top Padding	Specifies the space in inches or centimeters between the text and the top border of the object. Enter a numeric value to change the padding. Data type: Float
Border	
Border Color	Specifies the color of the border of the object. Choose a color from the drop-down list, enter a hexadecimal RGB value (for example, 0xCCAA00), or select Custom from the drop-down list to specify the color. Data type: String
Border Thickness	Specifies the width of the border in inches or centimeters. Enter a numeric value to change the thickness. Data type: Float
Bottom Line	Specifies the line style of the bottom border of the object. Choose a style from the drop-down list. Data type: Enumeration
Left Line	Specifies the line style of the left border of the object. Choose a style from the drop-down list. Data type: Enumeration
Right Line	Specifies the line style of the right border of the object. Choose a style from the drop-down list. Data type: Enumeration
Shadow	Specifies whether to draw two shadowed borders, beneath and to the right of the object. Data type: Boolean
Shadow Color	Specifies the color of the shadow. Choose a color from the drop-down list, enter a hexadecimal RGB value (for example, 0xff0000), or select Custom from the drop-down list to specify the color and fill effect. Data type: String
Top Line	Specifies the line style of the top border of the object. Choose a style from the drop-down list. Data type: Enumeration
Pattern	
Pattern Color	Specifies the color in which to draw a pattern to fill the object. Choose a color from the drop-down list, enter a hexadecimal RGB value (for example, 0xff0000), or select Custom from the drop-down list to specify the color and fill effect. Data type: String
Pattern Style	Specifies the style of the pattern. Choose an option from the drop-down list. • **none** - No pattern will be applied to the object. • **50%** - The object will be filled in the Pattern Color of 50%-transparency. • **horizontal** - The pattern will be made of horizontal lines of the Pattern Color. • **vertical** - The pattern will be made of vertical lines of the Pattern Color. • **grid** - The pattern will be an overlapping of horizontal and vertical lines of the Pattern Color. • **diagonal** - The pattern will be made of diagonal lines of the Pattern Color. Data type: Enumeration
Text Format	
Auto Fit	Specifies whether to adjust the width of the object according to the contents. Data type: Boolean
Bold	Specifies whether to make the text bold. Data type: Boolean
Font Face	Specifies the font of the text. Choose an option from the drop-down list. Data type: Enumeration
Font Size	Specifies the font size of the text. Enter an integer value to change the size. Data type: Integer
Format	Specifies the display format of the text in the report result. It varies with data type and can be manually defined. Data type: String **Note:** For the BigDecimal type, to avoid precision loss, you should enter a prefix JRD when setting the format property value.
Horizontal Alignment	Specifies the horizontal justification of the text in the object. Choose an option from the drop-down list. Data type: Enumeration
Italic	Specifies whether to make the text italic. Data type: Boolean
Maximum Width	Specifies the maximum width of the text displayed. Enter a numeric value to change the width. Data type: Float
Strikethrough	Specifies whether to add a strikethrough line to the text. Data type: Boolean
Underline	Specifies whether to underline the text. Data type: Boolean
Vertical Alignment	Specifies the vertical justification of the text in the object. Choose an option from the drop-down list. Data type: Enumeration
Word Wrap	Specifies whether to wrap the text to the width. Data type: Boolean
Others	
Data Mapping File	Specifies the data mapping file to the object for NLS use. For example, if the data mapping file name is Category_de_DE.properties, set the value as Category (without the language and locale part). Data type: String
Link	Specifies to link the object to another report, a website, an e-mail address or a Blob data type field. Click in the value cell to set the link target. For details, see Binding a link to a label. Data type: String
Record Location	Specifies the calculation point for the properties which use formulas. Choose an option from the drop-down list. • **Default** - The values of the properties are calculated in the default location where they are located. • **Page Header** - The values of the properties are calculated in the page header. • **Page Footer** - The values of the properties are calculated in the page footer. Data type: Enumeration
Transfer Style	Specifies whether the style group of the primary report will be applied to the linked report if the object is linked to a report. Data type: Boolean
Value Delimiter	Specifies the separator for the data whose type is DBArray. The default value space means that the elements will be displayed in a horizontal line and separated by a space. Data type: String
Accessibility	
Abbr	It is mapped to the HTML attribute *abbr*. This attribute specifies an abbreviated form of the content of the object. The abbreviated text may be rendered by user agents when appropriate in place of the object's content. Data type: String
Axis	It is mapped to the HTML attribute *axis*. This attribute is used to place a cell into conceptual categories that can be considered to form axes in an n-dimensional space. The value of this attribute is a comma-separated list of category names. Data type: String
Headers	It is mapped to the HTML attribute *headers*. This attribute specifies the list of header cells that provide header information for the current data cell. The value of this attribute is a space-separated list of cell names; those cells must be named by setting their *id* attribute. Data type: String
Scope	It is mapped to the HTML attribute *scope*. This attribute specifies the set of data cells for which the current header cell provides header information. Available values are: • **Row** - The current cell provides header information for the rest of the row that contains it. • **Column** - The current cell provides header information for the rest of the column that contains it. • **none** - The scope attribute will not be generated when exporting to HTML. Data type: String

Formula field

The properties of a formula field in a crosstab are:

Property Name	Description
General	
Formula Name	Shows the name of the formula. This property is read only.
Geometry (only for aggregate field)	
Height	Specifies the height of the object, in inches or centimeters. Enter a numeric value to change the height. Data type: Float
Width	Specifies the width of the object, in inches or centimeters. Enter a numeric value to change the width. Data type: Float
Color	
Background	Specifies the background color of the object. Choose a color from the drop-down list, enter a hexadecimal RGB value (for example, 0xff0000), or select Custom from the drop-down list to specify the color. Data type: String
Foreground	Specifies the foreground color of the object. Choose a color from the drop-down list, enter a hexadecimal RGB value (for example, 0xff0000), or select Custom from the drop-down list to specify the color. Data type: String
CSS	
Class	Specifies a CSS class to be applied to the object which is a valid class in the CSS file. Data type: String
ID	Specifies the identifier of the object, which must be unique in the report. The ID property can be a style sheet selector. Data type: String
Style	The property can be used in two ways. • Specifies a style to be applied to the object. Choose a style from the drop-down list which is available when a Style Group has been selected at the report tab level and when there are styles in the Style Group that can be applied to the object, or type in the style name. • Specifies a CSS selector to be applied to the object. Enter a valid class name from the CSS file. Data type: String
Padding	
Bottom Padding	Specifies the space in inches or centimeters between the text and the bottom border of the object. Enter a numeric value to change the padding. Data type: Float
Left Padding	Specifies the space in inches or centimeters between the text and the left border of the object. Enter a numeric value to change the padding. Data type: Float
Right Padding	Specifies the space in inches or centimeters between the text and the right border of the object. Enter a numeric value to change the padding. Data type: Float
Top Padding	Specifies the space in inches or centimeters between the text and the top border of the object. Enter a numeric value to change the padding. Data type: Float
Border	
Border Color	Specifies the color of the border of the object. Choose a color from the drop-down list, enter a hexadecimal RGB value (for example, 0xCCAA00), or select Custom from the drop-down list to specify the color. Data type: String
Border Thickness	Specifies the width of the border in inches or centimeters. Enter a numeric value to change the thickness. Data type: Float
Bottom Line	Specifies the line style of the bottom border of the object. Choose a style from the drop-down list. Data type: Enumeration
Left Line	Specifies the line style of the left border of the object. Choose a style from the drop-down list. Data type: Enumeration
Right Line	Specifies the line style of the right border of the object. Choose a style from the drop-down list. Data type: Enumeration
Shadow	Specifies whether to draw two shadowed borders, beneath and to the right of the object. Data type: Boolean
Shadow Color	Specifies the color of the shadow. Choose a color from the drop-down list, enter a hexadecimal RGB value (for example, 0xff0000), or select Custom from the drop-down list to specify the color and fill effect. Data type: String
Top Line	Specifies the line style of the top border of the object. Choose a style from the drop-down list. Data type: Enumeration
Pattern	
Pattern Color	Specifies the color in which to draw a pattern to fill the object. Choose a color from the drop-down list, enter a hexadecimal RGB value (for example, 0xff0000), or select Custom from the drop-down list to specify the color and fill effect. Data type: String
Pattern Style	Specifies the style of the pattern. Choose an option from the drop-down list. • **none** - No pattern will be applied to the object. • **50%** - The object will be filled in the Pattern Color of 50%-transparency. • **horizontal** - The pattern will be made of horizontal lines of the Pattern Color. • **vertical** - The pattern will be made of vertical lines of the Pattern Color. • **grid** - The pattern will be an overlapping of horizontal and vertical lines of the Pattern Color. • **diagonal** - The pattern will be made of diagonal lines of the Pattern Color. Data type: Enumeration
Text Format	
Auto Fit	Specifies whether to adjust the width of the object according to the contents. Data type: Boolean
Bold	Specifies whether to make the text bold. Data type: Boolean
Font Face	Specifies the font of the text. Choose an option from the drop-down list. Data type: Enumeration
Font Size	Specifies the font size of the text. Enter an integer value to change the size. Data type: Integer
Format	Specifies the display format of the text in the report result. It varies with data type and can be manually defined. Data type: String **Note:** For the BigDecimal type, to avoid precision loss, you should enter a prefix JRD when setting the format property value.
Horizontal Alignment	Specifies the horizontal justification of the text in the object. Choose an option from the drop-down list. Data type: Enumeration
Italic	Specifies whether to make the text italic. Data type: Boolean
Maximum Width	Specifies the maximum width of the text displayed. Enter a numeric value to change the width. Data type: Float
Strikethrough	Specifies whether to add a strikethrough line to the text. Data type: Boolean
Underline	Specifies whether to underline the text. Data type: Boolean
Vertical Alignment	Specifies the vertical justification of the text in the object. Choose an option from the drop-down list. Data type: Enumeration
Word Wrap	Specifies whether to wrap the text to the width. Data type: Boolean
Others	
Data Mapping File	Specifies the data mapping file to the object for NLS use. For example, if the data mapping file name is Category_de_DE.properties, set the value as Category (without the language and locale part). Data type: String
Link	Specifies to link the object to another report, a website, an e-mail address or a Blob data type field. Click in the value cell to set the link target. For details, see Binding a link to a label. Data type: String
Record Location	Specifies the calculation point for the properties which use formulas. Choose an option from the drop-down list. • **Default** - The values of the properties are calculated in the default location where they are located. • **Page Header** - The values of the properties are calculated in the page header. • **Page Footer** - The values of the properties are calculated in the page footer. Data type: Enumeration
Transfer Style	Specifies whether the style group of the primary report will be applied to the linked report if the object is linked to a report. Data type: Boolean
Value Delimiter	Specifies the separator for the data whose type is DBArray. The default value space means that the elements will be displayed in a horizontal line and separated by a space. Data type: String
Accessibility	
Abbr	It is mapped to the HTML attribute *abbr*. This attribute specifies an abbreviated form of the content of the object. The abbreviated text may be rendered by user agents when appropriate in place of the object's content. Data type: String
Axis	It is mapped to the HTML attribute *axis*. This attribute is used to place a cell into conceptual categories that can be considered to form axes in an n-dimensional space. The value of this attribute is a comma-separated list of category names. Data type: String
Headers	It is mapped to the HTML attribute *headers*. This attribute specifies the list of header cells that provide header information for the current data cell. The value of this attribute is a space-separated list of cell names; those cells must be named by setting their *id* attribute. Data type: String
Scope	It is mapped to the HTML attribute *scope*. This attribute specifies the set of data cells for which the current header cell provides header information. Available values are: • **Row** - The current cell provides header information for the rest of the row that contains it. • **Column** - The current cell provides header information for the rest of the column that contains it. • **none** - The scope attribute will not be generated when exporting to HTML. Data type: String

Label

The header of a total row or column, the display name of a column or row field, and the display name of an aggregate field are represented as labels in the Report Structure panel of the Report Inspector. Their class types are CTTextField, CTDBTitleField, and CTLabelField respectively.

The properties of a label in a crosstab are:

Property Name	Description
Text Format	
Bold	Specifies whether to make the text bold. Data type: Boolean
Font Face	Specifies the font of the text. Choose an option from the drop-down list. Data type: Enumeration
Font Size	Specifies the font size of the text. Enter an integer value to change the size. Data type: Integer
Horizontal Alignment	Specifies the horizontal justification of the text in the object. Choose an option from the drop-down list. Data type: Enumeration
Italic	Specifies whether to make the text italic. Data type: Boolean
Strikethrough	Specifies whether to add a strikethrough line to the text. Data type: Boolean
Text	Specifies the text in the label. Enter a string to display as the label text. Data type: String
Underline	Specifies whether to underline the text. Data type: Boolean
Vertical Alignment	Specifies the vertical justification of the text in the object. Choose an option from the drop-down list. Data type: Enumeration
Word Wrap	Specifies whether to wrap the text to the width. Data type: Boolean
Color	
Background	Specifies the background color of the object. Choose a color from the drop-down list, enter a hexadecimal RGB value (for example, 0xff0000), or select Custom from the drop-down list to specify the color. Data type: String
Foreground	Specifies the foreground color of the object. Choose a color from the drop-down list, enter a hexadecimal RGB value (for example, 0xff0000), or select Custom from the drop-down list to specify the color. Data type: String
CSS	
Class	Specifies a CSS class to be applied to the object which is a valid class in the CSS file. Data type: String
ID	Specifies the identifier of the object, which must be unique in the report. The ID property can be a style sheet selector. Data type: String
Style	The property can be used in two ways. • Specifies a style to be applied to the object. Choose a style from the drop-down list which is available when a Style Group has been selected at the report tab level and when there are styles in the Style Group that can be applied to the object, or type in the style name. • Specifies a CSS selector to be applied to the object. Enter a valid class name from the CSS file. Data type: String
Padding	
Bottom Padding	Specifies the space in inches or centimeters between the text and the bottom border of the object. Enter a numeric value to change the padding. Data type: Float
Left Padding	Specifies the space in inches or centimeters between the text and the left border of the object. Enter a numeric value to change the padding. Data type: Float
Right Padding	Specifies the space in inches or centimeters between the text and the right border of the object. Enter a numeric value to change the padding. Data type: Float
Top Padding	Specifies the space in inches or centimeters between the text and the top border of the object. Enter a numeric value to change the padding. Data type: Float
Border	
Border Color	Specifies the color of the border of the object. Choose a color from the drop-down list, enter a hexadecimal RGB value (for example, 0xCCAA00), or select Custom from the drop-down list to specify the color. Data type: String
Border Thickness	Specifies the width of the border in inches or centimeters. Enter a numeric value to change the thickness. Data type: Float
Bottom Line	Specifies the line style of the bottom border of the object. Choose a style from the drop-down list. Data type: Enumeration
Left Line	Specifies the line style of the left border of the object. Choose a style from the drop-down list. Data type: Enumeration
Right Line	Specifies the line style of the right border of the object. Choose a style from the drop-down list. Data type: Enumeration
Shadow	Specifies whether to draw two shadowed borders, beneath and to the right of the object. Data type: Boolean
Shadow Color	Specifies the color of the shadow. Choose a color from the drop-down list, enter a hexadecimal RGB value (for example, 0xff0000), or select Custom from the drop-down list to specify the color and fill effect. Data type: String
Top Line	Specifies the line style of the top border of the object. Choose a style from the drop-down list. Data type: Enumeration
Pattern	
Pattern Color	Specifies the color in which to draw a pattern to fill the object. Choose a color from the drop-down list, enter a hexadecimal RGB value (for example, 0xff0000), or select Custom from the drop-down list to specify the color and fill effect. Data type: String
Pattern Style	Specifies the style of the pattern. Choose an option from the drop-down list. • **none** - No pattern will be applied to the object. • **50%** - The object will be filled in the Pattern Color of 50%-transparency. • **horizontal** - The pattern will be made of horizontal lines of the Pattern Color. • **vertical** - The pattern will be made of vertical lines of the Pattern Color. • **grid** - The pattern will be an overlapping of horizontal and vertical lines of the Pattern Color. • **diagonal** - The pattern will be made of diagonal lines of the Pattern Color. Data type: Enumeration
Others	
Invisible	Specifies whether to show the object in the design area and in the report results. All formulas and calculations will still be performed if the property is set to true. Data type: Boolean
Link	Specifies to link the object to another report, a website, an e-mail address or a Blob data type field. Click ... in the value cell to set the link target. For details, see Binding a link to a label. Data type: String
Record Location	Specifies the calculation point for the properties which use formulas. Choose an option from the drop-down list. • **Default** - The values of the properties are calculated in the default location where they are located. • **Page Header** - The values of the properties are calculated in the page header. • **Page Footer** - The values of the properties are calculated in the page footer. Data type: Enumeration
Suppress	Specifies whether to show the object in the design area and in the report results. All formulas and calculations will be skipped if the property is set to true. Data type: Boolean **Note:** When both the Invisible and Suppress properties of an object are set to true, Suppress has the higher priority.
Suppress Aggregate	Specifies whether to hide the Total row or column. Not applicable for CTLabelField. Data type: Boolean
Suppress When No Records	Specifies whether to display the object in the report results when no record is returned to its parent data container. Data type: Boolean
Transfer Style	Specifies whether the style group of the primary report will be applied to the linked report if the object is linked to a report. Data type: Boolean
Accessibility	
Abbr	It is mapped to the HTML attribute *abbr*. This attribute specifies an abbreviated form of the content of the object. The abbreviated text may be rendered by user agents when appropriate in place of the object's content. Data type: String
Axis	It is mapped to the HTML attribute *axis*. This attribute is used to place a cell into conceptual categories that can be considered to form axes in an n-dimensional space. The value of this attribute is a comma-separated list of category names. Data type: String
Headers	It is mapped to the HTML attribute *headers*. This attribute specifies the list of header cells that provide header information for the current data cell. The value of this attribute is a space-separated list of cell names; those cells must be named by setting their *id* attribute. Data type: String
Scope	It is mapped to the HTML attribute *scope*. This attribute specifies the set of data cells for which the current header cell provides header information. Available values are: • **Row** - The current cell provides header information for the rest of the row that contains it. • **Column** - The current cell provides header information for the rest of the column that contains it. • **none** - The scope attribute will not be generated when exporting to HTML. Data type: String

DBField

The properties of a DBField in a web report are:

Property Name	Description
General	
DBField Name	Shows the mapping name of the field in the catalog. This property is read only.
Geometry	
Height	Specifies the height of the object, in inches or centimeters. Enter a numeric value to change the height. Data type: Float
Width	Specifies the width of the object, in inches or centimeters. Enter a numeric value to change the width. Data type: Float
Color	
Background	Specifies the background color of the object. Choose a color from the drop-down list, enter a hexadecimal RGB value (for example, 0xff0000), or select Custom from the drop-down list to specify the color. Data type: String
Foreground	Specifies the foreground color of the object. Choose a color from the drop-down list, enter a hexadecimal RGB value (for example, 0xff0000), or select Custom from the drop-down list to specify the color. Data type: String
CSS	
Class	Specifies a CSS class to be applied to the object which is a valid class in the CSS file. Data type: String
ID	Specifies the identifier of the object, which must be unique in the report. The ID property can be a style sheet selector. Data type: String
Style	The property can be used in two ways. • Specifies a style to be applied to the object. Choose a style from the drop-down list which is available when a Style Group has been selected at the report tab level and when there are styles in the Style Group that can be applied to the object, or type in the style name. • Specifies a CSS selector to be applied to the object. Enter a valid class name from the CSS file. Data type: String
Excel	
Column Index	Specifies the X coordinate of the object relative to its parent container when exported to Excel or CSV, measured in cells. The Columned property at the report level must be set to true for this property to take effect. Data type: Integer
Row Index	Specifies the Y coordinate of the object relative to its parent container when exported to Excel or CSV, measured in cells. The Columned property at the report level must be set to true for this property to take effect. Data type: Integer
Padding	
Bottom Padding	Specifies the space in inches or centimeters between the text and the bottom border of the object. Enter a numeric value to change the padding. Data type: Float
Left Padding	Specifies the space in inches or centimeters between the text and the left border of the object. Enter a numeric value to change the padding. Data type: Float
Right Padding	Specifies the space in inches or centimeters between the text and the right border of the object. Enter a numeric value to change the padding. Data type: Float
Top Padding	Specifies the space in inches or centimeters between the text and the top border of the object. Enter a numeric value to change the padding. Data type: Float
Border	
Border Color	Specifies the color of the border of the object. Choose a color from the drop-down list, enter a hexadecimal RGB value (for example, 0xCCAA00), or select Custom from the drop-down list to specify the color. Data type: String
Border Thickness	Specifies the width of the border in inches or centimeters. Enter a numeric value to change the thickness. Data type: Float
Bottom Line	Specifies the line style of the bottom border of the object. Choose a style from the drop-down list. Data type: Enumeration
Left Line	Specifies the line style of the left border of the object. Choose a style from the drop-down list. Data type: Enumeration
Right Line	Specifies the line style of the right border of the object. Choose a style from the drop-down list. Data type: Enumeration
Shadow	Specifies whether to draw two shadowed borders, beneath and to the right of the object. Data type: Boolean
Shadow Color	Specifies the color of the shadow. Choose a color from the drop-down list, enter a hexadecimal RGB value (for example, 0xff0000), or select Custom from the drop-down list to specify the color and fill effect. Data type: String
Top Line	Specifies the line style of the top border of the object. Choose a style from the drop-down list. Data type: Enumeration
Pattern	
Pattern Color	Specifies the color in which to draw a pattern to fill the object. Choose a color from the drop-down list, enter a hexadecimal RGB value (for example, 0xff0000), or select Custom from the drop-down list to specify the color and fill effect. Data type: String
Pattern Style	Specifies the style of the pattern. Choose an option from the drop-down list. • **none** - No pattern will be applied to the object. • **50%** - The object will be filled in the Pattern Color of 50%-transparency. • **horizontal** - The pattern will be made of horizontal lines of the Pattern Color. • **vertical** - The pattern will be made of vertical lines of the Pattern Color. • **grid** - The pattern will be an overlapping of horizontal and vertical lines of the Pattern Color. • **diagonal** - The pattern will be made of diagonal lines of the Pattern Color. Data type: Enumeration
Text Format	
Auto Fit	Specifies whether to adjust the width and height of the object according to the contents. Data type: Boolean
Bold	Specifies whether to make the text bold. Data type: Boolean
Font Face	Specifies the font of the text. Choose an option from the drop-down list. Data type: Enumeration
Font Size	Specifies the font size of the text. Enter an integer value to change the size. Data type: Integer
Format	Specifies the display format of the text in the report result. It varies with data type and can be manually defined. Data type: String **Note:** For the BigDecimal type, to avoid precision loss, you should enter a prefix JRD when setting the format property value.
Horizontal Alignment	Specifies the horizontal justification of the text in the object. Choose an option from the drop-down list. Data type: Enumeration
Italic	Specifies whether to make the text italic. Data type: Boolean
Maximum Width	Specifies the maximum width of the text displayed. Enter a numeric value to change the width. This property often works with the Auto Fit property. If Auto Fit = true and Maximum Width is not equal to 0, the text will extend until the width is this value. Data type: Float
Strikethrough	Specifies whether to add a strikethrough line to the text. Data type: Boolean
Underline	Specifies whether to underline the text. Data type: Boolean
Vertical Alignment	Specifies the vertical justification of the text in the object. Choose an option from the drop-down list. Data type: Enumeration
Word Wrap	Specifies whether to wrap the text to the width. Data type: Boolean
Others	
Data Mapping File	Specifies the data mapping file to the object for NLS use. For example, if the data mapping file name is Category_de_DE.properties, set the value as Category (without the language and locale part). Data type: String
Enable Hyperlink in Excel	Specifies whether to enable the link defined on the object when exporting the report to an Excel file. Data type: Boolean
Enable Hyperlink in HTML	Specifies whether to enable the link defined on the object when exporting the report to an HTML file. Data type: Boolean
Enable Hyperlink in PDF	Specifies whether to enable the link defined on the object when exporting the report to a PDF file. Data type: Boolean
Export to CSV	Specifies whether to include the object when exporting the report to CSV. Data type: Boolean
Export to Excel	Specifies whether to include the object when exporting the report to Excel. Data type: Boolean
Export to Report Result	Specifies whether to include the object when exporting the report to Report Result. Data type: Boolean
Invisible	Specifies whether to show the object in the design area and in the report results. All formulas and calculations will still be performed if the property is set to true. Data type: Boolean
Link	Specifies to link the object to another report, a website, an e-mail address or a Blob data type field. Click ... in the value cell to set the link target. For details, see Binding a link to a label. Data type: String
Logic Column	Specifies whether to show the object in the next visible table cell in the same row when the column which holds the object is hidden. Choose an option from the drop-down list. Data type: Enumeration **Notes:** • The next visible table cell should be completely empty, that is, the cell should not hold any content including blank space. • When the property for several objects in the same row is set to next visible column, and the columns holding these objects are all hidden, then only the object in the rightmost column will be shown in the next visible cell.
Position	Specifies the position of the object. Choose an option from the drop-down list. • **static** - The object is positioned at the location in which it is inserted. If selected, the X, Y and other position-related properties will be hidden or disabled. The position of the object is affected by preceding objects in the same container. • **absolute** - The object will be located at the position specified by dragging and dropping or by setting its X and Y property values. The position of the object is not affected by preceding objects in the same container. Data type: Enumeration
Record Location	Specifies the calculation point for the properties which use formulas. Choose an option from the drop-down list. • **Default** - The values of the properties are calculated in the default location where they are located. • **Page Header** - The values of the properties are calculated in the page header. • **Page Footer** - The values of the properties are calculated in the page footer. Data type: Enumeration
Transfer Style	Specifies whether the style group of the primary report will be applied to the linked report if the object is linked to a report. Data type: Boolean
Value Delimiter	Specifies the separator for the data whose type is DBArray. The default value space means that the elements will be displayed in a horizontal line and separated by a space. Data type: String

Note: For tables, only these DBField properties can be rendered in Web Report Studio:

- Data Mapping File
- Link
- Properties of the DBField object in the Web Report Studio Inspector panel.

Formula field

The properties of a formula field in a web report are:

Property Name	Description
General	
Formula Name	Shows the name of the formula. This property is read only.
Geometry	
Height	Specifies the height of the object, in inches or centimeters. Enter a numeric value to change the height. Data type: Float
Width	Specifies the width of the object, in inches or centimeters. Enter a numeric value to change the width. Data type: Float
Color	
Background	Specifies the background color of the object. Choose a color from the drop-down list, enter a hexadecimal RGB value (for example, 0xffd0000), or select Custom from the drop-down list to specify the color. Data type: String
Foreground	Specifies the foreground color of the object. Choose a color from the drop-down list, enter a hexadecimal RGB value (for example, 0xff0000), or select Custom from the drop-down list to specify the color. Data type: String
CSS	
Class	Specifies a CSS class to be applied to the object which is a valid class in the CSS file. Data type: String
ID	Specifies the identifier of the object, which must be unique in the report. The ID property can be a style sheet selector. Data type: String
Style	The property can be used in two ways. • Specifies a style to be applied to the object. Choose a style from the drop-down list which is available when a Style Group has been selected at the report tab level and when there are styles in the Style Group that can be applied to the object, or type in the style name. • Specifies a CSS selector to be applied to the object. Enter a valid class name from the CSS file. Data type: String
Excel	
Column Index	Specifies the X coordinate of the object relative to its parent container when exported to Excel or CSV, measured in cells. The Columned property at the report level must be set to true for this property to take effect. Data type: Integer
Row Index	Specifies the Y coordinate of the object relative to its parent container when exported to Excel or CSV, measured in cells. The Columned property at the report level must be set to true for this property to take effect. Data type: Integer
Padding	
Bottom Padding	Specifies the space in inches or centimeters between the text and the bottom border of the object. Enter a numeric value to change the padding. Data type: Float
Left Padding	Specifies the space in inches or centimeters between the text and the left border of the object. Enter a numeric value to change the padding. Data type: Float
Right Padding	Specifies the space in inches or centimeters between the text and the right border of the object. Enter a numeric value to change the padding. Data type: Float
Top Padding	Specifies the space in inches or centimeters between the text and the top border of the object. Enter a numeric value to change the padding. Data type: Float
Border	
Border Color	Specifies the color of the border of the object. Choose a color from the drop-down list, enter a hexadecimal RGB value (for example, 0xCCAA00), or select Custom from the drop-down list to specify the color. Data type: String
Border Thickness	Specifies the width of the border in inches or centimeters. Enter a numeric value to change the thickness. Data type: Float
Bottom Line	Specifies the line style of the bottom border of the object. Choose a style from the drop-down list. Data type: Enumeration
Left Line	Specifies the line style of the left border of the object. Choose a style from the drop-down list. Data type: Enumeration
Right Line	Specifies the line style of the right border of the object. Choose a style from the drop-down list. Data type: Enumeration
Shadow	Specifies whether to draw two shadowed borders, beneath and to the right of the object. Data type: Boolean
Shadow Color	Specifies the color of the shadow. Choose a color from the drop-down list, enter a hexadecimal RGB value (for example, 0xff0000), or select Custom from the drop-down list to specify the color and fill effect. Data type: String
Top Line	Specifies the line style of the top border of the object. Choose a style from the drop-down list. Data type: Enumeration
Pattern	
Pattern Color	Specifies the color in which to draw a pattern to fill the object. Choose a color from the drop-down list, enter a hexadecimal RGB value (for example, 0xff0000), or select Custom from the drop-down list to specify the color and fill effect. Data type: String
Pattern Style	Specifies the style of the pattern. Choose an option from the drop-down list. • **none** - No pattern will be applied to the object. • **50%** - The object will be filled in the Pattern Color of 50%-transparency. • **horizontal** - The pattern will be made of horizontal lines of the Pattern Color. • **vertical** - The pattern will be made of vertical lines of the Pattern Color. • **grid** - The pattern will be an overlapping of horizontal and vertical lines of the Pattern Color. • **diagonal** - The pattern will be made of diagonal lines of the Pattern Color. Data type: Enumeration
Text Format	
Auto Fit	Specifies whether to adjust the width and height of the object according to the contents. Data type: Boolean
Bold	Specifies whether to make the text bold. Data type: Boolean
Font Face	Specifies the font of the text. Choose an option from the drop-down list. Data type: Enumeration
Font Size	Specifies the font size of the text. Enter an integer value to change the size. Data type: Integer
Format	Specifies the display format of the text in the report result. It varies with data type and can be manually defined. Data type: String **Note:** For the BigDecimal type, to avoid precision loss, you should enter a prefix JRD when setting the format property value.
Horizontal Alignment	Specifies the horizontal justification of the text in the object. Choose an option from the drop-down list. Data type: Enumeration
Italic	Specifies whether to make the text italic. Data type: Boolean
Maximum Width	Specifies the maximum width of the text displayed. Enter a numeric value to change the width. This property often works with the Auto Fit property. If Auto Fit = true and Maximum Width is not equal to 0, the text will extend until the width is this value. Data type: Float
Strikethrough	Specifies whether to add a strikethrough line to the text. Data type: Boolean
Underline	Specifies whether to underline the text. Data type: Boolean
Vertical Alignment	Specifies the vertical justification of the text in the object. Choose an option from the drop-down list. Data type: Enumeration
Word Wrap	Specifies whether to wrap the text to the width. Data type: Boolean
Others	
Data Mapping File	Specifies the data mapping file to the object for NLS use. For example, if the data mapping file name is Category_de_DE.properties, set the value as Category (without the language and locale part). Data type: String
Enable Hyperlink in Excel	Specifies whether to enable the link defined on the object when exporting the report to an Excel file. Data type: Boolean
Enable Hyperlink in HTML	Specifies whether to enable the link defined on the object when exporting the report to an HTML file. Data type: Boolean
Enable Hyperlink in PDF	Specifies whether to enable the link defined on the object when exporting the report to a PDF file. Data type: Boolean
Export to CSV	Specifies whether to include the object when exporting the report to CSV. Data type: Boolean
Export to Excel	Specifies whether to include the object when exporting the report to Excel. Data type: Boolean
Invisible	Specifies whether to show the object in the design area and in the report results. All formulas and calculations will still be performed if the property is set to true. Data type: Boolean
Link	Specifies to link the object to another report, a website, an e-mail address or a Blob data type field. Click ... in the value cell to set the link target. For details, see Binding a link to a label. Data type: String
Logic Column	Specifies whether to show the object in the next visible table cell in the same row when the column which holds the object is hidden. Choose an option from the drop-down list. Data type: Enumeration **Notes:** • The next visible table cell should be completely empty, that is, the cell should not hold any content including blank space. • When the property for several objects in the same row is set to next visible column, and the columns holding these objects are all hidden, then only the object in the rightmost column will be shown in the next visible cell.
Position	Specifies the position of the object. Choose an option from the drop-down list. • **static** - The object is positioned at the location in which it is inserted. If selected, the X, Y and other position-related properties will be hidden or disabled. The position of the object is affected by preceding objects in the same container. • **relative** - The object is positioned at an offset to the position at which it is inserted. The offset is determined by the X and Y property values. • **absolute** - The object will be located at the position specified by dragging and dropping or by setting its X and Y property values. The position of the object is not affected by preceding objects in the same container. Data type: Enumeration
Record Location	Specifies the calculation point for the properties which use formulas. Choose an option from the drop-down list. • **Default** - The values of the properties are calculated in the default location where they are located. • **Page Header** - The values of the properties are calculated in the page header. • **Page Footer** - The values of the properties are calculated in the page footer. Data type: Enumeration
Transfer Style	Specifies whether the style group of the primary report will be applied to the linked report if the object is linked to a report. Data type: Boolean
Value Delimiter	Specifies the separator for the data whose type is DBArray. The default value space means that the elements will be displayed in a horizontal line and separated by a space. Data type: String

Geographic map

The properties of a geographic map in a web report are:

Property Name	Description
Geography	
Display As	Specifies the display type of the map location. Choose an option from the drop-down list. • **Marker** - Displays the map location as marker. • **Area** - Displays the map location as area. • **Marker and Area** - Displays the map location as marker and area. Data type: Enumeration
Geometry	
Height	Specifies the height of the object, in inches or centimeters. Enter a numeric value to change the height. Data type: Float
Width	Specifies the width of the object, in inches or centimeters. Enter a numeric value to change the width. Data type: Float
Others	
Default for Filter	Specifies whether to show the data component as the default component in the Apply to drop-down list of the Filter dialog in Web Report Studio. Data type: Boolean **Note:** In a web report, only one data component's Default for Filter property can be set to true.
Export to Applet	Specifies whether to include the object when exporting the report to Applet. Data type: Boolean
Export to Excel	Specifies whether to include the object when exporting the report to Excel. Data type: Boolean
Export to HTML	Specifies whether to include the object when exporting the report to HTML. Data type: Boolean
Export to PDF	Specifies whether to include the object when exporting the report to PDF. Data type: Boolean
Export to PostScript	Specifies whether to include the object when exporting the report to PostScript. Data type: Boolean
Export to Report Result	Specifies whether to include the object when exporting the report to Report Result. Data type: Boolean
Export to RTF	Specifies whether to include the object when exporting the report to RTF. Data type: Boolean
Invisible	Specifies whether to show the object in the design area and in the report results. All formulas and calculations will still be performed if the property is set to true. Data type: Boolean
Invisible for Filter Dialogs	Specifies whether to show the data component in the Apply to drop-down list of the Filter dialog in Web Report Studio. This property cannot be edited when Default for Filter is set to true. Data type: Boolean
Suppress	Specifies whether to show the object in the design area and in the report results. All formulas and calculations will be skipped if the property is set to true. Data type: Boolean **Note:** When both the Invisible and Suppress properties of an object are set to true, Suppress has the higher priority.
Suppress When No Records	Specifies whether to display the object in the report results when no record is returned to its parent data container. Data type: Boolean
TOC	
Anchor Display Value	Specifies a string or formula to display for the TOC entry for this object in the TOC Browser. By default it is the value of the specified object. The TOC Anchor property must be set to true for this property to take effect. Data type: String
TOC Anchor	Specifies whether to include the object in the TOC Browser for the report. Data type: Boolean

A geographic map has two child objects, marker and area, the properties of which will be shown in the following topic:

- Geographic map marker
- Geographic map area

Geographic map marker

The properties of a geographic map marker are:

Property Name	Description
Others	
Invisible	Specifies whether to show the object in the design area and in the report results. All formulas and calculations will still be performed if the property is set to true. Data type: Boolean

Geographic map area

The properties of a geographic map area are:

Property Name	Description
Boundary	
Boundary Color	Specifies the color of the area border. Choose a color from the drop-down list, enter a hexadecimal RGB value (for example, 0xff0000), or select Custom from the drop-down list to specify the color and fill effect. Data type: String
Boundary Style	Specifies the line style of the area border. Choose an option from the drop-down list. Data type: Enumeration
Boundary Width	Specifies the width of the area border. Enter a numeric value to change the width. Data type: Float
Others	
Invisible	Specifies whether to show the object in the design area and in the report results. All formulas and calculations will still be performed if the property is set to true. Data type: Boolean
Suppress	Specifies whether to show the object in the design area and in the report results. All formulas and calculations will be skipped if the property is set to true. Data type: Boolean **Note:** When both the Invisible and Suppress properties of an object are set to true, Suppress has the higher priority.
Suppress When No Records	Specifies whether to display the object in the report results when no record is returned to its parent data container. Data type: Boolean

Image

The properties of an image in a web report are:

Property Name	Description
Geometry	
Height	Specifies the height of the object, in inches or centimeters. Enter a numeric value to change the height. Data type: Float
Width	Specifies the width of the object, in inches or centimeters. Enter a numeric value to change the width. Data type: Float
CSS	
Class	Specifies a CSS class to be applied to the object which is a valid class in the CSS file. Data type: String
ID	Specifies the identifier of the object, which must be unique in the report. The ID property can be a style sheet selector. Data type: String
Style	The property can be used in two ways. • Specifies a style to be applied to the object. Choose a style from the drop-down list which is available when a Style Group has been selected at the report tab level and when there are styles in the Style Group that can be applied to the object, or type in the style name. • Specifies a CSS selector to be applied to the object. Enter a valid class name from the CSS file. Data type: String
Others	
Export to Applet	Specifies whether to include the object when exporting the report to Applet. Data type: Boolean
Export to Excel	Specifies whether to include the object when exporting the report to Excel. Data type: Boolean
Export to HTML	Specifies whether to include the object when exporting the report to HTML. Data type: Boolean
Export to PDF	Specifies whether to include the object when exporting the report to PDF. Data type: Boolean
Export to PostScript	Specifies whether to include the object when exporting the report to PostScript. Data type: Boolean
Export to Report Result	Specifies whether to include the object when exporting the report to Report Result. Data type: Boolean
Export to RTF	Specifies whether to include the object when exporting the report to RTF. Data type: Boolean
Horizontal Alignment	Specifies the horizontal alignment of the image in the container. Choose an option from the drop-down list. Data type: Enumeration
Invisible	Specifies whether to show the object in the design area and in the report results. All formulas and calculations will still be performed if the property is set to true. Data type: Boolean
Link	Specifies to link the object to another report, a website, an e-mail address or a Blob data type field. Click ... in the value cell to set the link target. For details, see Binding a link to a label. Data type: String
Logic Column	Specifies whether to show the object in the next visible table cell in the same row when the column which holds the object is hidden. Choose an option from the drop-down list. Data type: Enumeration **Notes:** • The next visible table cell should be completely empty, that is, the cell should not hold any content including blank space. • When the property for several objects in the same row is set to next visible column, and the columns holding these objects are all hidden, then only the object in the rightmost column will be shown in the next visible cell.
Record Location	Specifies the calculation point for the properties which use formulas. Choose an option from the drop-down list. • **Default** - The values of the properties are calculated in the default location where they are located. • **Page Header** - The values of the properties are calculated in the page header. • **Page Footer** - The values of the properties are calculated in the page footer. Data type: Enumeration
Transfer Style	Specifies whether the style group of the primary report will be applied to the linked report if the object is linked to a report. Data type: Boolean
Vertical Alignment	Specifies the vertical alignment of the image in the container. Choose an option from the drop-down list. Data type: Enumeration
Excel	
Column Index	Specifies the X coordinate of the object relative to its parent container when exported to Excel or CSV, measured in cells. The Columned property at the report level must be set to true for this property to take effect. Data type: Integer
Column Number	Specifies the number of columns which will be the object's width in the exported Excel or CSV file. The Columned property at the report level must be set to true for this property to take effect. Data type: String
Row Index	Specifies the Y coordinate of the object relative to its parent container when exported to Excel or CSV, measured in cells. The Columned property at the report level must be set to true for this property to take effect. Data type: Integer
Row Number	Specifies the number of rows which will be the object's height in the exported Excel or CSV file. The Columned property at the report level must be set to true for this property to take effect. Data type: String
Image Property	
Alternate Text	Specifies the text that will be displayed instead if the image cannot be displayed. Data type: String
Disabled	Specifies whether to disable the actions on the image. Data type: Boolean
Image Name	Specifies the file name of the image. The image can be one on your local file system or retrieved by using a URL. Data type: String
Maximum Scaling Ratio	Specifies the maximum ratio up to which the image can be scaled. Data type: Float
Name	Specifies the name of the image. Data type: String
Rotation	Specifies the angle at which to rotate the image, in degrees. The following is the meaning of different values: • 0 - No rotation. • positive value - Rotates the image clockwise. • negative value - Rotates the image anticlockwise. Data type: Float
Scaling Mode	Specifies the scaling mode of the image, which controls the image behavior when you try to adjust the image field size. Choose an option from the drop-down list. • **actual size** - The image will remain its original size. If the image field size is smaller than the image size, part of the image will be cut off. • **fit image** - The image size will be automatically adjusted to fill up the image field, with its original perspective remained but under the limitation of Maximum Scaling Ratio. • **fit width** - The image will be automatically adjusted to fit the width of the image field but under the limitation of Maximum Scaling Ratio. • **fit height** - The image will be automatically adjusted to fit the height of the image field but under the limitation of Maximum Scaling Ratio. • **customize** - The image size will be automatically adjusted to fit the image field, regardless of the Maximum Scaling Ratio's value. Data type: Enumeration
Tab Index	Specifies the index that defines the tab order for the image. Data type: Integer
Title	Specifies the tip information about the image, which will be displayed when you hover the mouse pointer over the image at runtime. Data type: String
Value	Specifies the initial value of the image. Data type: String
Event	
On Blur	Specifies the action when the object loses input focus. Data type: String
On Click	Specifies the action when the user clicks the left mouse button on the object. Data type: String
On Double Click	Specifies the action when the user double-clicks the object. Data type: String
On Focus	Specifies the action when the object receives focus. Data type: String
On Key Down	Specifies the action when the user presses a key. Data type: String
On Key Press	Specifies the action when the user presses an alphanumeric key. Data type: String
On Key Up	Specifies the action when the user releases a key. Data type: String
On Mouse Down	Specifies the action when the user clicks the object with either mouse button. Data type: String
On Mouse Move	Specifies the action when the user moves the mouse over the object. Data type: String
On Mouse Out	Specifies the action when the user moves the mouse pointer outside the boundaries of the object. Data type: String
On Mouse Over	Specifies the action when the user moves the mouse pointer into the object. Data type: String
On Mouse Up	Specifies the action when the user releases a mouse button while the mouse is over the object. Data type: String

Label

The properties of a label in a web report are:

Property Name	Description
Text Format	
Auto Fit	Specifies whether to adjust the width and height of the object according to the contents. Data type: Boolean
Bold	Specifies whether to make the text bold. Data type: Boolean
Font Face	Specifies the font of the text. Choose an option from the drop-down list. Data type: Enumeration
Font Size	Specifies the font size of the text. Enter an integer value to change the size. Data type: Integer
Horizontal Alignment	Specifies the horizontal justification of the text in the object. Choose an option from the drop-down list. Data type: Enumeration
Italic	Specifies whether to make the text italic. Data type: Boolean
Strikethrough	Specifies whether to add a strikethrough line to the text. Data type: Boolean
Text	Specifies the text in the label. Enter a string to display as the label text. Data type: String
Underline	Specifies whether to underline the text. Data type: Boolean
Vertical Alignment	Specifies the vertical justification of the text in the object. Choose an option from the drop-down list. Data type: Enumeration
Word Wrap	Specifies whether to wrap the text to the width. Data type: Boolean
Geometry	
Height	Specifies the height of the object, in inches or centimeters. Enter a numeric value to change the height. Data type: Float
Width	Specifies the width of the object, in inches or centimeters. Enter a numeric value to change the width. Data type: Float
Color	
Background	Specifies the background color of the object. Choose a color from the drop-down list, enter a hexadecimal RGB value (for example, 0xff0000), or select Custom from the drop-down list to specify the color. Data type: String
Foreground	Specifies the foreground color of the object. Choose a color from the drop-down list, enter a hexadecimal RGB value (for example, 0xff0000), or select Custom from the drop-down list to specify the color. Data type: String
CSS	
Class	Specifies a CSS class to be applied to the object which is a valid class in the CSS file. Data type: String
ID	Specifies the identifier of the object, which must be unique in the report. The ID property can be a style sheet selector. Data type: String
Style	The property can be used in two ways. • Specifies a style to be applied to the object. Choose a style from the drop-down list which is available when a Style Group has been selected at the report tab level and when there are styles in the Style Group that can be applied to the object, or type in the style name. • Specifies a CSS selector to be applied to the object. Enter a valid class name from the CSS file. Data type: String
Excel	
Column Index	Specifies the X coordinate of the object relative to its parent container when exported to Excel or CSV, measured in cells. The Columned property at the report tab level must be set to true for this property to take effect. Data type: Integer
Row Index	Specifies the Y coordinate of the object relative to its parent container when exported to Excel or CSV, measured in cells. The Columned property at the report tab level must be set to true for this property to take effect. Data type: Integer
Padding	
Bottom Padding	Specifies the space in inches or centimeters between the text and the bottom border of the object. Enter a numeric value to change the padding. Data type: Float
Left Padding	Specifies the space in inches or centimeters between the text and the left border of the object. Enter a numeric value to change the padding. Data type: Float
Right Padding	Specifies the space in inches or centimeters between the text and the right border of the object. Enter a numeric value to change the padding. Data type: Float
Top Padding	Specifies the space in inches or centimeters between the text and the top border of the object. Enter a numeric value to change the padding. Data type: Float
Border	
Border Color	Specifies the color of the border of the object. Choose a color from the drop-down list, enter a hexadecimal RGB value (for example, 0xCCAA00), or select Custom from the drop-down list to specify the color. Data type: String
Border Thickness	Specifies the width of the border in inches or centimeters. Enter a numeric value to change the thickness. Data type: Float
Bottom Line	Specifies the line style of the bottom border of the object. Choose a style from the drop-down list. Data type: Enumeration
Left Line	Specifies the line style of the left border of the object. Choose a style from the drop-down list. Data type: Enumeration
Right Line	Specifies the line style of the right border of the object. Choose a style from the drop-down list. Data type: Enumeration
Shadow	Specifies whether to draw two shadowed borders, beneath and to the right of the object. Data type: Boolean
Shadow Color	Specifies the color of the shadow. Choose a color from the drop-down list, enter a hexadecimal RGB value (for example, 0xff0000), or select Custom from the drop-down list to specify the color and fill effect. Data type: String
Top Line	Specifies the line style of the top border of the object. Choose a style from the drop-down list. Data type: Enumeration
Pattern	
Pattern Color	Specifies the color in which to draw a pattern to fill the object. Choose a color from the drop-down list, enter a hexadecimal RGB value (for example, 0xff0000), or select Custom from the drop-down list to specify the color and fill effect. Data type: String
Pattern Style	Specifies the style of the pattern. Choose an option from the drop-down list. • **none** - No pattern will be applied to the object. • **50%** - The object will be filled in the Pattern Color of 50%-transparency. • **horizontal** - The pattern will be made of horizontal lines of the Pattern Color. • **vertical** - The pattern will be made of vertical lines of the Pattern Color. • **grid** - The pattern will be an overlapping of horizontal and vertical lines of the Pattern Color. • **diagonal** - The pattern will be made of diagonal lines of the Pattern Color. Data type: Enumeration
Others	
Bind Column	When the label's Sortable or Filterable property is set to true, you need to specify a field as the value so that the record of the field can be sorted or filtered via the proper button beside the label when viewing the report in Web Report Studio. Data type: String
Enable Hyperlink in Excel	Specifies whether to enable the link defined on the object when exporting the report to an Excel file. Data type: Boolean
Enable Hyperlink in HTML	Specifies whether to enable the link defined on the object when exporting the report to an HTML file. Data type: Boolean
Enable Hyperlink in PDF	Specifies whether to enable the link defined on the object when exporting the report to a PDF file. Data type: Boolean
Export to Applet	Specifies whether to include the object when exporting the report to Applet. Data type: Boolean
Export to CSV	Specifies whether to include the object when exporting the report to CSV. Data type: Boolean
Export to Excel	Specifies whether to include the object when exporting the report to Excel. Data type: Boolean
Export to HTML	Specifies whether to include the object when exporting the report to HTML. Data type: Boolean
Export to PDF	Specifies whether to include the object when exporting the report to PDF. Data type: Boolean
Export to PostScript	Specifies whether to include the object when exporting the report to PostScript. Data type: Boolean
Export to Report Result	Specifies whether to include the object when exporting the report to Report Result. Data type: Boolean
Export to RTF	Specifies whether to include the object when exporting the report to RTF. Data type: Boolean
Export to Text	Specifies whether to include the object when exporting the report to Text. Data type: Boolean
Export to XML	Specifies whether to include the object when exporting the report to XML. Data type: Boolean
Filterable	Specifies whether the records can be filtered. If true, a filter button appears beside the label when viewing the report in Web Report Studio. You can use the button to filter the records based on a specific field's value. The field is specified via the Bind Column property. Data type: Boolean
Invisible	Specifies whether to show the object in the design area and in the report results. All formulas and calculations will still be performed if the property is set to true. Data type: Boolean
Link	Specifies to link the object to another report, a website, an e-mail address or a Blob data type field. Click [...] in the value cell to set the link target. For details, see Binding a link to a label. Data type: String
Logic Column	Specifies whether to show the object in the next visible table cell in the same row when the column which holds the object is hidden. Available to labels in table only. Choose an option from the drop-down list. Data type: Enumeration **Notes:** • The next visible table cell should be completely empty, that is, the cell should not hold any content including blank space. • When the property for several objects in the same row is set to next visible column, and the columns holding these objects are all hidden, then only the object in the rightmost column will be shown in the next visible cell.
Position	Specifies the position of the object. Choose an option from the drop-down list. • **static** - The object is positioned at the location in which it is inserted. If selected, the X, Y and other position-related properties will be hidden or disabled. The position of the object is affected by preceding objects in the same container. • **absolute** - The object will be located at the position specified by dragging and dropping or by setting its X and Y property values. The position of the object is not affected by preceding objects in the same container. Data type: Enumeration
Record Location	Specifies the calculation point for the properties which use formulas. Choose an option from the drop-down list. • **Default** - The values of the properties are calculated in the default location where they are located. • **Page Header** - The values of the properties are calculated in the page header. • **Page Footer** - The values of the properties are calculated in the page footer. Data type: Enumeration
Sortable	Specifies whether the records can be sorted. If true, a sort button appears beside the label when viewing the report in Web Report Studio. You can use the button to arrange the records of a specific field in ascending or descending order. The field is specified via the Bind Column property. Data type: Boolean
Suppress	Specifies whether to show the object in the design area and in the report results. All formulas and calculations will be skipped if the property is set to true. Data type: Boolean **Note:** When both the Invisible and Suppress properties of an object are set to true, Suppress has the higher priority.
Suppress When No Records	Specifies whether to display the object in the report results when no record is returned to its parent data container. Data type: Boolean
Transfer Style	Specifies whether the style group of the primary report will be applied to the linked report if the object is linked to a report. Data type: Boolean

Note: For tables, only these label properties can be rendered in Web Report Studio:

- Link
- Properties of the label object in the Web Report Studio Inspector panel.

Multimedia object (OLE object)

The properties of a multimedia object which is also called OLE object in a web report are:

Property Name	Description
Geometry	
Height	Specifies the height of the object, in inches or centimeters. Enter a numeric value to change the height. Data type: Float
Width	Specifies the width of the object, in inches or centimeters. Enter a numeric value to change the width. Data type: Float
Color	
Background	Specifies the background color of the object. Choose a color from the drop-down list, enter a hexadecimal RGB value (for example, 0xff0000), or select Custom from the drop-down list to specify the color. Data type: String
CSS	
Class	Specifies a CSS class to be applied to the object which is a valid class in the CSS file. Data type: String
Style	The property can be used in two ways. • Specifies a style to be applied to the object. Choose a style from the drop-down list which is available when a Style Group has been selected at the report tab level and when there are styles in the Style Group that can be applied to the object, or type in the style name. • Specifies a CSS selector to be applied to the object. Enter a valid class name from the CSS file. Data type: String
Others	
Export to Applet	Specifies whether to include the object when exporting the report to Applet. Data type: Boolean
Export to CSV	Specifies whether to include the object when exporting the report to CSV. Data type: Boolean
Export to Excel	Specifies whether to include the object when exporting the report to Excel. Data type: Boolean
Export to HTML	Specifies whether to include the object when exporting the report to HTML. Data type: Boolean
Export to PDF	Specifies whether to include the object when exporting the report to PDF. Data type: Boolean
Export to PostScript	Specifies whether to include the object when exporting the report to PostScript. Data type: Boolean
Export to Report Result	Specifies whether to include the object when exporting the report to Report Result. Data type: Boolean
Export to RTF	Specifies whether to include the object when exporting the report to RTF. Data type: Boolean
Export to Text	Specifies whether to include the object when exporting the report to Text. Data type: Boolean
Export to XML	Specifies whether to include the object when exporting the report to XML. Data type: Boolean
Invisible	Specifies whether to show the object in the design area and in the report results. All formulas and calculations will still be performed if the property is set to true. Data type: Boolean
Record Location	Specifies the calculation point for the properties which use formulas. Choose an option from the drop-down list. • **Default** - The values of the properties are calculated in the default location where they are located. • **Page Header** - The values of the properties are calculated in the page header. • **Page Footer** - The values of the properties are calculated in the page footer. Data type: Enumeration
Excel	
Column Index	Specifies the X coordinate of the object relative to its parent container when exported to Excel or CSV, measured in cells. The Columned property at the report level must be set to true for this property to take effect. Data type: Integer
Row Index	Specifies the Y coordinate of the object relative to its parent container when exported to Excel or CSV, measured in cells. The Columned property at the report level must be set to true for this property to take effect. Data type: Integer
Border	
Border Color	Specifies the color of the border of the object. Choose a color from the drop-down list, enter a hexadecimal RGB value (for example, 0xCCAA00), or select Custom from the drop-down list to specify the color. Data type: String
Border Thickness	Specifies the width of the border in inches or centimeters. Enter a numeric value to change the thickness. Data type: Float
Bottom Line	Specifies the line style of the bottom border of the object. Choose a style from the drop-down list. Data type: Enumeration
Left Line	Specifies the line style of the left border of the object. Choose a style from the drop-down list. Data type: Enumeration
Right Line	Specifies the line style of the right border of the object. Choose a style from the drop-down list. Data type: Enumeration
Top Line	Specifies the line style of the top border of the object. Choose a style from the drop-down list. Data type: Enumeration
Object Settings	
Access Key	Specifies the accelerator key for the object. Data type: String
Alternate Text	Specifies the text will be displayed instead if the object cannot be displayed. Data type: String
Archive	Specifies the character string that can be used to implement your own archive functionality for the object. Data type: String
Class ID	Specifies the class identifier of the object. Data type: String
Code Base	Specifies the base path used to resolve relative URIs specified by the Class ID and Archive properties. When null, its default value is the base URI of the current report. Data type: String
Declare	Specifies whether to make the current object definition a declaration only. The object must be instantiated by a subsequent object definition referring to this declaration. Data type: Boolean
Horizontal Space	Specifies the horizontal margin for the object. Data type: Float
Name	Specifies the name of the object. Data type: String
Stand By	Specifies the message to be displayed while the object is being loaded. Data type: String
Tab Index	Specifies the index that defines the tab order for the object. Data type: Integer
Title	Specifies the tip information of the object, which will be displayed when you hover the mouse pointer over the object after the report is exported to HTML or Page Report Result. Data type: String
Use Map	Specifies the URL, often with a bookmark extension (#name), to use as a client-side image map. Data type: String
Value Type	Specifies the MIME (Multipurpose Internet Mail Extensions) type of the object. Data type: String
Vertical Space	Specifies the vertical margin for the object. Data type: Float

Page panel

The properties of a page panel in a web report are:

Property Name	Description
Paper	
Height	Specifies the height of the page in inches or centimeters. The Width and Height values change when Orientation changes. The Page Type setting determines the Width and Height values but you can change the latter as desired. Enter a numeric value to change the height. Data type: Float
Orientation	Specifies how to position the report page. The Orientation setting affects the Width and Height values. Choose an option from the drop-down list. • **portrait** - The page is positioned vertically. • **landscape** - The page is positioned horizontally. Data type: Enumeration
Page Type	Specifies the report page dimensions. Choose a size from the drop-down list. Data type: Enumeration
Width	Specifies the width of the page in inches or centimeters. The Width and Height values change when Orientation changes. The Page Type setting determines the Width and Height values but you can change the latter as desired. Enter a numeric value to change the width. Data type: Float
Margin	
Bottom Margin	Specifies the distance in inches or centimeters between the bottom page edge and the data area. Enter a numeric value to change the margin. Data type: Float
Left Margin	Specifies the distance in inches or centimeters between the left page edge and the data area. Enter a numeric value to change the margin. Data type: Float
Right Margin	Specifies the distance in inches or centimeters between the right page edge and the data area. Enter a numeric value to change the margin. Data type: Float
Top Margin	Specifies the distance in inches or centimeters between the top page edge and the data area. Enter a numeric value to change the thickness. Data type: Float

The following topics list properties of the children of a page panel:

- Page header panel
- Page footer panel

Page header panel

The properties of a page header panel in a web report are:

Property Name	Description
General	
Class Type	Indicates the class type of the object. This property is read only.
Geometry	
Height	Specifies the height of the object, in inches or centimeters. Enter a numeric value to change the height. Data type: Float
Width	Displays the width of the object, in inches or centimeters. This property is read only.
Color	
Background	Specifies the background color and fill effect of the object. Choose a color from the drop-down list, enter a hexadecimal RGB value (for example, 0xff0000), or select Custom from the drop-down list to specify the color and fill effect. Data type: String
CSS	
Class	Specifies a CSS class to be applied to the object which is a valid class in the CSS file. Data type: String
Style	The property can be used in two ways. • Specifies a style to be applied to the object. Choose a style from the drop-down list which is available when a Style Group has been selected at the report tab level and when there are styles in the Style Group that can be applied to the object, or type in the style name. • Specifies a CSS selector to be applied to the object. Enter a valid class name from the CSS file. Data type: String
Others	
Export to CSV	Specifies whether to include the object when exporting the report to CSV. Data type: Boolean
Export to Excel	Specifies whether to include the object when exporting the report to Excel. Data type: Boolean
Invisible	Specifies whether to show the object in the design area and in the report results. All formulas and calculations will still be performed if the property is set to true. Data type: Boolean
Record Location	Specifies the calculation point for the properties which use formulas. Choose an option from the drop-down list. • **Default** - The values of the properties are calculated in the default location where they are located. • **Page Header** - The values of the properties are calculated in the page header. • **Page Footer** - The values of the properties are calculated in the page footer. Data type: Enumeration
Show Bottom Line	Specifies whether to include a horizontal line at the bottom of the panel. Applies to panels in table components only. The default is false, which means that no horizontal line appears at the bottom of the panel. Data type: Boolean
Suppress	Specifies whether to show the object in the design area and in the report results. All formulas and calculations will be skipped if the property is set to true. Not supported in Web Report Studio. Data type: Boolean **Note:** When both the Invisible and Suppress properties of an object are set to true, Suppress has the higher priority.
Suppress Blank Panel	Specifies whether a panel with no data is included in the report results. The default is false which means an empty panel is shown. Data type: Boolean
Suppress When No Records	Specifies whether to display the object in the report results when no record is returned to its parent data container. Not supported in Web Report Studio. Data type: Boolean
Underlay	Specifies whether the panel will be layered beneath the next panel in the banded object. A panel layered beneath a subsequent panel can be used to contain a background picture or watermark. Banded page header, page footer, or group footer panels cannot be layered. The default is false which means that the panel is not layered beneath the next panel. Data type: Boolean
Border	
Border Color	Specifies the color of the border of the object. Choose a color from the drop-down list, enter a hexadecimal RGB value (for example, 0xCCAA00), or select Custom from the drop-down list to specify the color. Data type: String
Border Thickness	Specifies the width of the border in inches or centimeters. Enter a numeric value to change the thickness. Data type: Float
Bottom Line	Specifies the line style of the bottom border of the object. Choose a style from the drop-down list. Data type: Enumeration
Left Line	Specifies the line style of the left border of the object. Choose a style from the drop-down list. Data type: Enumeration
Right Line	Specifies the line style of the right border of the object. Choose a style from the drop-down list. Data type: Enumeration
Top Line	Specifies the line style of the top border of the object. Choose a style from the drop-down list. Data type: Enumeration

Page footer panel

The properties of a page footer panel in a web report are:

Property Name	Description
General	
Class Type	Indicates the class type of the object. This property is read only.
Geometry	
Height	Specifies the height of the object, in inches or centimeters. Enter a numeric value to change the height. Data type: Float
Width	Displays the width of the object, in inches or centimeters. This property is read only.
Color	
Background	Specifies the background color and fill effect of the object. Choose a color from the drop-down list, enter a hexadecimal RGB value (for example, 0xff0000), or select Custom from the drop-down list to specify the color and fill effect. Data type: String
CSS	
Class	Specifies a CSS class to be applied to the object which is a valid class in the CSS file. Data type: String
Style	The property can be used in two ways. • Specifies a style to be applied to the object. Choose a style from the drop-down list which is available when a Style Group has been selected at the report tab level and when there are styles in the Style Group that can be applied to the object, or type in the style name. • Specifies a CSS selector to be applied to the object. Enter a valid class name from the CSS file. Data type: String
Others	
Export to CSV	Specifies whether to include the object when exporting the report to CSV. Data type: Boolean
Export to Excel	Specifies whether to include the object when exporting the report to Excel. Data type: Boolean
Invisible	Specifies whether to show the object in the design area and in the report results. All formulas and calculations will still be performed if the property is set to true. Data type: Boolean
Record Location	Specifies the calculation point for the properties which use formulas. Choose an option from the drop-down list. • **Default** - The values of the properties are calculated in the default location where they are located. • **Page Header** - The values of the properties are calculated in the page header. • **Page Footer** - The values of the properties are calculated in the page footer. Data type: Enumeration
Show Bottom Line	Specifies whether to include a horizontal line at the bottom of the panel. Applies to panels in table components only. The default is false, which means that no horizontal line appears at the bottom of the panel. Data type: Boolean
Suppress	Specifies whether to show the object in the design area and in the report results. All formulas and calculations will be skipped if the property is set to true. Not supported in Web Report Studio. Data type: Boolean **Note:** When both the Invisible and Suppress properties of an object are set to true, Suppress has the higher priority.
Suppress Blank Panel	Specifies whether a panel with no data is included in the report results. The default is false which means an empty panel is shown. Data type: Boolean
Suppress on Last Page	If true, the footer panel on the last page will not be displayed. Data type: Boolean
Suppress When No Records	Specifies whether to display the object in the report results when no record is returned to its parent data container. Not supported in Web Report Studio. Data type: Boolean
Underlay	Specifies whether the panel will be layered beneath the next panel in the banded object. A panel layered beneath a subsequent panel can be used to contain a background picture or watermark. Banded page header, page footer, or group footer panels cannot be layered. The default is false which means that the panel is not layered beneath the next panel. Data type: Boolean
Border	
Border Color	Specifies the color of the border of the object. Choose a color from the drop-down list, enter a hexadecimal RGB value (for example, 0xff0000), or select Custom from the drop-down list to specify the color. Data type: String
Border Thickness	Specifies the width of the border in inches or centimeters. Enter a numeric value to change the thickness. Data type: Float
Bottom Line	Specifies the line style of the bottom border of the object. Choose a style from the drop-down list. Data type: Enumeration
Left Line	Specifies the line style of the left border of the object. Choose a style from the drop-down list. Data type: Enumeration
Right Line	Specifies the line style of the right border of the object. Choose a style from the drop-down list. Data type: Enumeration
Top Line	Specifies the line style of the top border of the object. Choose a style from the drop-down list. Data type: Enumeration

Parameter field

The properties of a parameter field in a web report are:

Property Name	Description
General	
Parameter Name	Shows the name of the parameter. This property is read only.
Geometry	
Height	Specifies the height of the object, in inches or centimeters. Enter a numeric value to change the height. Data type: Float
Width	Specifies the width of the object, in inches or centimeters. Enter a numeric value to change the width. Data type: Float
Color	
Background	Specifies the background color of the object. Choose a color from the drop-down list, enter a hexadecimal RGB value (for example, 0xff0000), or select Custom from the drop-down list to specify the color. Data type: String
Foreground	Specifies the foreground color of the object. Choose a color from the drop-down list, enter a hexadecimal RGB value (for example, 0xff0000), or select Custom from the drop-down list to specify the color. Data type: String
CSS	
ID	Specifies the identifier of the object, which must be unique in the report. The ID property can be a style sheet selector. A String type value.
Class	Specifies a CSS class to be applied to the object which is a valid class in the CSS file. Data type: String
Style	The property can be used in two ways. • Specifies a style to be applied to the object. Choose a style from the drop-down list which is available when a Style Group has been selected at the report tab level and when there are styles in the Style Group that can be applied to the object, or type in the style name. • Specifies a CSS selector to be applied to the object. Enter a valid class name from the CSS file. Data type: String
Excel	
Column Index	Specifies the X coordinate of the object relative to its parent container when exported to Excel or CSV, measured in cells. The Columned property at the report level must be set to true for this property to take effect. Data type: Integer
Row Index	Specifies the Y coordinate of the object relative to its parent container when exported to Excel or CSV, measured in cells. The Columned property at the report level must be set to true for this property to take effect. Data type: Integer
Padding	
Bottom Padding	Specifies the space in inches or centimeters between the text and the bottom border of the object. Enter a numeric value to change the padding. Data type: Float
Left Padding	Specifies the space in inches or centimeters between the text and the left border of the object. Enter a numeric value to change the padding. Data type: Float
Right Padding	Specifies the space in inches or centimeters between the text and the right border of the object. Enter a numeric value to change the padding. Data type: Float
Top Padding	Specifies the space in inches or centimeters between the text and the top border of the object. Enter a numeric value to change the padding. Data type: Float
Border	
Border Color	Specifies the color of the border of the object. Choose a color from the drop-down list, enter a hexadecimal RGB value (for example, 0xCCAA00), or select Custom from the drop-down list to specify the color. Data type: String
Border Thickness	Specifies the width of the border in inches or centimeters. Enter a numeric value to change the thickness. Data type: Float
Bottom Line	Specifies the line style of the bottom border of the object. Choose a style from the drop-down list. Data type: Enumeration
Left Line	Specifies the line style of the left border of the object. Choose a style from the drop-down list. Data type: Enumeration
Right Line	Specifies the line style of the right border of the object. Choose a style from the drop-down list. Data type: Enumeration
Shadow	Specifies whether to draw two shadowed borders, beneath and to the right of the object. Data type: Boolean
Shadow Color	Specifies the color of the shadow. Choose a color from the drop-down list, enter a hexadecimal RGB value (for example, 0xff0000), or select Custom from the drop-down list to specify the color and fill effect. Data type: String
Top Line	Specifies the line style of the top border of the object. Choose a style from the drop-down list. Data type: Enumeration
Pattern	
Pattern Color	Specifies the color in which to draw a pattern to fill the object. Choose a color from the drop-down list, enter a hexadecimal RGB value (for example, 0xff0000), or select Custom from the drop-down list to specify the color and fill effect. Data type: String
Pattern Style	Specifies the style of the pattern. Choose an option from the drop-down list. • **none** - No pattern will be applied to the object. • **50%** - The object will be filled in the Pattern Color of 50%-transparency. • **horizontal** - The pattern will be made of horizontal lines of the Pattern Color. • **vertical** - The pattern will be made of vertical lines of the Pattern Color. • **grid** - The pattern will be an overlapping of horizontal and vertical lines of the Pattern Color. • **diagonal** - The pattern will be made of diagonal lines of the Pattern Color. Data type: Enumeration
Text Format	
Auto Fit	Specifies whether to adjust the width and height of the object according to the contents. Data type: Boolean
Bold	Specifies whether to make the text bold. Data type: Boolean
Font Face	Specifies the font of the text. Choose an option from the drop-down list. Data type: Enumeration
Font Size	Specifies the font size of the text. Enter an integer value to change the size. Data type: Integer
Format	Specifies the display format of the text in the report result. It varies with data type and can be manually defined. Data type: String **Note:** For the BigDecimal type, to avoid precision loss, you should enter a prefix JRD when setting the format property value.
Horizontal Alignment	Specifies the horizontal justification of the text in the object. Choose an option from the drop-down list. Data type: Enumeration
Italic	Specifies whether to make the text italic. Data type: Boolean
Maximum Width	Specifies the maximum width of the text displayed. Enter a numeric value to change the width. This property often works with the Auto Fit property. If Auto Fit = true and Maximum Width is not equal to 0, the text will extend until the width is this value. Data type: Float
Strikethrough	Specifies whether to add a strikethrough line to the text. Data type: Boolean
Underline	Specifies whether to underline the text. Data type: Boolean
Vertical Alignment	Specifies the vertical justification of the text in the object. Choose an option from the drop-down list. Data type: Enumeration
Word Wrap	Specifies whether to wrap the text to the width. Data type: Boolean
Others	
Data Mapping File	Specifies the data mapping file to the object for NLS use. For example, if the data mapping file name is Category_de_DE.properties, set the value as Category (without the language and locale part). Data type: String
Enable Hyperlink in Excel	Specifies whether to enable the link defined on the object when exporting the report to an Excel file. Data type: Boolean
Enable Hyperlink in HTML	Specifies whether to enable the link defined on the object when exporting the report to an HTML file. Data type: Boolean
Enable Hyperlink in PDF	Specifies whether to enable the link defined on the object when exporting the report to a PDF file. Data type: Boolean
Export to CSV	Specifies whether to include the object when exporting the report to CSV. Data type: Boolean
Export to Excel	Specifies whether to include the object when exporting the report to Excel. Data type: Boolean
Invisible	Specifies whether to show the object in the design area and in the report results. All formulas and calculations will still be performed if the property is set to true. Data type: Boolean
Link	Specifies to link the object to another report, a website, an e-mail address or a Blob data type field. Click ... in the value cell to set the link target. For details, see Binding a link to a label. Data type: String
Logic Column	Specifies whether to show the object in the next visible table cell in the same row when the column which holds the object is hidden. Choose an option from the drop-down list. Data type: Enumeration **Notes:** • The next visible table cell should be completely empty, that is, the cell should not hold any content including blank space. • When the property for several objects in the same row is set to next visible column, and the columns holding these objects are all hidden, then only the object in the rightmost column will be shown in the next visible cell.
Parameter	Substitutes the current parameter with another one by selecting from the drop-down list. To make the property editable, uncheck Forbid changing parameter in the Options dialog (**File** > **Options** > **Panel** > **Forbid changing parameter**). Data type: String
Position	Specifies the position of the object. Choose an option from the drop-down list. • **static** - The object is positioned at the location in which it is inserted. If selected, the X, Y and other position-related properties will be hidden or disabled. The position of the object is affected by preceding objects in the same container. • **absolute** - The object will be located at the position specified by dragging and dropping or by setting its X and Y property values. The position of the object is not affected by preceding objects in the same container. Data type: Enumeration
Record Location	Specifies the calculation point for the properties which use formulas. Choose an option from the drop-down list. • **Default** - The values of the properties are calculated in the default location where they are located. • **Page Header** - The values of the properties are calculated in the page header. • **Page Footer** - The values of the properties are calculated in the page footer. Data type: Enumeration
Suppress	Specifies whether to show the object in the design area and in the report results. All formulas and calculations will be skipped if the property is set to true. Not supported in Web Report Studio. Data type: Boolean **Note:** When both the Invisible and Suppress properties of an object are set to true, Suppress has the higher priority.
Suppress When No Records	Specifies whether to display the object in the report results when no record is returned to its parent data container. Not supported in Web Report Studio. Data type: Boolean
Suppress When Null	If true and its value is null, the field will not be displayed. Data type: Boolean
Transfer Style	Specifies whether the style group of the primary report will be applied to the linked report if the object is linked to a report. Data type: Boolean
Value Delimiter	Specifies the separator for the data whose type is DBArray. The default value space means that the elements will be displayed in a horizontal line and separated by a space. Data type: String

Report body

The properties of the report body in a web report are:

Property Name	Description
Color	
Background	Specifies the background color and fill effect of the object. Choose a color from the drop-down list, enter a hexadecimal RGB value (for example, 0xff0000), or select Custom from the drop-down list to specify the color and fill effect. Data type: String
CSS	
Class	Specifies a CSS class to be applied to the object which is a valid class in the CSS file. Data type: String
Style	The property can be used in two ways. • Specifies a style to be applied to the object. Choose a style from the drop-down list which is available when a Style Group has been selected at the report tab level and when there are styles in the Style Group that can be applied to the object, or type in the style name. • Specifies a CSS selector to be applied to the object. Enter a valid class name from the CSS file. Data type: String
Others	
Record Location	Specifies the calculation point for the properties which use formulas. Choose an option from the drop-down list. • **Default** - The values of the properties are calculated in the default location where they are located. • **Page Header** - The values of the properties are calculated in the page header. • **Page Footer** - The values of the properties are calculated in the page footer. Data type: Enumeration
Suppress When Empty	Specifies whether to display the object in the report results when no record is returned to it. Data type: Boolean
Border	
Border Color	Specifies the color of the border of the object. Choose a color from the drop-down list, enter a hexadecimal RGB value (for example, 0xCCAA00), or select Custom from the drop-down list to specify the color. Data type: String
Border Thickness	Specifies the width of the border in inches or centimeters. Enter a numeric value to change the thickness. Data type: Float
Bottom Line	Specifies the line style of the bottom border of the object. Choose a style from the drop-down list. Data type: Enumeration
Left Line	Specifies the line style of the left border of the object. Choose a style from the drop-down list. Data type: Enumeration
Right Line	Specifies the line style of the right border of the object. Choose a style from the drop-down list. Data type: Enumeration
Top Line	Specifies the line style of the top border of the object. Choose a style from the drop-down list. Data type: Enumeration
TOC	
TOC Anchor	Specifies whether to include the object in the TOC Browser for the report. Data type: Boolean

Special field

The properties of a special field in a web report are:

Property Name	Description
Geometry	
Height	Specifies the height of the object, in inches or centimeters. Enter a numeric value to change the height. Data type: Float
Width	Specifies the width of the object, in inches or centimeters. Enter a numeric value to change the width. Data type: Float
Color	
Background	Specifies the background color of the object. Choose a color from the drop-down list, enter a hexadecimal RGB value (for example, 0xff0000), or select Custom from the drop-down list to specify the color. Data type: String
Foreground	Specifies the foreground color of the object. Choose a color from the drop-down list, enter a hexadecimal RGB value (for example, 0xff0000), or select Custom from the drop-down list to specify the color. Data type: String
CSS	
Class	Specifies a CSS class to be applied to the object which is a valid class in the CSS file. Data type: String
ID	Specifies the identifier of the object, which must be unique in the report. The ID property can be a style sheet selector. Data type: String
Style	The property can be used in two ways. • Specifies a style to be applied to the object. Choose a style from the drop-down list which is available when a Style Group has been selected at the report tab level and when there are styles in the Style Group that can be applied to the object, or type in the style name. • Specifies a CSS selector to be applied to the object. Enter a valid class name from the CSS file. Data type: String
Excel	
Column Index	Specifies the X coordinate of the object relative to its parent container when exported to Excel or CSV, measured in cells. The Columned property at the report level must be set to true for this property to take effect. Data type: Integer
Row Index	Specifies the Y coordinate of the object relative to its parent container when exported to Excel or CSV, measured in cells. The Columned property at the report level must be set to true for this property to take effect. Data type: Integer
Padding	
Bottom Padding	Specifies the space in inches or centimeters between the text and the bottom border of the object. Enter a numeric value to change the padding. Data type: Float
Left Padding	Specifies the space in inches or centimeters between the text and the left border of the object. Enter a numeric value to change the padding. Data type: Float
Right Padding	Specifies the space in inches or centimeters between the text and the right border of the object. Enter a numeric value to change the padding. Data type: Float
Top Padding	Specifies the space in inches or centimeters between the text and the top border of the object. Enter a numeric value to change the padding. Data type: Float
Border	
Border Color	Specifies the color of the border of the object. Choose a color from the drop-down list, enter a hexadecimal RGB value (for example, 0xCCAA00), or select Custom from the drop-down list to specify the color. Data type: String
Border Thickness	Specifies the width of the border in inches or centimeters. Enter a numeric value to change the thickness. Data type: Float
Bottom Line	Specifies the line style of the bottom border of the object. Choose a style from the drop-down list. Data type: Enumeration
Left Line	Specifies the line style of the left border of the object. Choose a style from the drop-down list. Data type: Enumeration
Right Line	Specifies the line style of the right border of the object. Choose a style from the drop-down list. Data type: Enumeration
Shadow	Specifies whether to draw two shadowed borders, beneath and to the right of the object. Data type: Boolean
Shadow Color	Specifies the color of the shadow. Choose a color from the drop-down list, enter a hexadecimal RGB value (for example, 0xff0000), or select Custom from the drop-down list to specify the color and fill effect. Data type: String
Top Line	Specifies the line style of the top border of the object. Choose a style from the drop-down list. Data type: Enumeration
Pattern	
Pattern Color	Specifies the color in which to draw a pattern to fill the object. Choose a color from the drop-down list, enter a hexadecimal RGB value (for example, 0xff0000), or select Custom from the drop-down list to specify the color and fill effect. Data type: String
Pattern Style	Specifies the style of the pattern. Choose an option from the drop-down list. • **none** - No pattern will be applied to the object. • **50%** - The object will be filled in the Pattern Color of 50%-transparency. • **horizontal** - The pattern will be made of horizontal lines of the Pattern Color. • **vertical** - The pattern will be made of vertical lines of the Pattern Color. • **grid** - The pattern will be an overlapping of horizontal and vertical lines of the Pattern Color. • **diagonal** - The pattern will be made of diagonal lines of the Pattern Color. Data type: Enumeration
Text Format	
Auto Fit	Specifies whether to adjust the width and height of the object according to the contents. Data type: Boolean
Bold	Specifies whether to make the text bold. Data type: Boolean
Font Face	Specifies the font of the text. Choose an option from the drop-down list. Data type: Enumeration
Font Size	Specifies the font size of the text. Enter an integer value to change the size. Data type: Integer
Format	Specifies the display format of the text in the report result. It varies with data type and can be manually defined. Data type: String **Note:** For the BigDecimal type, to avoid precision loss, you should enter a prefix JRD when setting the format property value.
Horizontal Alignment	Specifies the horizontal justification of the text in the object. Choose an option from the drop-down list. Data type: Enumeration
Italic	Specifies whether to make the text italic. Data type: Boolean
Maximum Width	Specifies the maximum width of the text displayed. Enter a numeric value to change the width. This property often works with the Auto Fit property. If Auto Fit = true and Maximum Width is not equal to 0, the text will extend until the width is this value. Data type: Float
Strikethrough	Specifies whether to add a strikethrough line to the text. Data type: Boolean
Underline	Specifies whether to underline the text. Data type: Boolean
Vertical Alignment	Specifies the vertical justification of the text in the object. Choose an option from the drop-down list. Data type: Enumeration
Word Wrap	Specifies whether to wrap the text to the width. Data type: Boolean
Others	
Data Evaluation Setting	Specifies the group information for the object. Available only for the group by fields in table group columns. Choose an option from the drop-down list. • **current column** The object will take value of the group by field in the current column. • **current row** The object will take value of the group by field in the current row. Data type: Enumeration
Data Mapping File	Specifies the data mapping file to the object for NLS use. For example, if the data mapping file name is Category_de_DE.properties, set the value as Category (without the language and locale part). Data type: String
Enable Hyperlink in Excel	Specifies whether to enable the link defined on the object when exporting the report to an Excel file. Data type: Boolean
Enable Hyperlink in HTML	Specifies whether to enable the link defined on the object when exporting the report to an HTML file. Data type: Boolean
Enable Hyperlink in PDF	Specifies whether to enable the link defined on the object when exporting the report to a PDF file. Data type: Boolean
Export to CSV	Specifies whether to include the object when exporting the report to CSV. Data type: Boolean
Export to Excel	Specifies whether to include the object when exporting the report to Excel. Data type: Boolean
Field Type	Specifies another special field format to substitute the current special field. Choose an option from the drop-down list. Data type: Enumeration
Invisible	Specifies whether to show the object in the design area and in the report results. All formulas and calculations will still be performed if the property is set to true. Data type: Boolean
Link	Specifies to link the object to another report, a website, an e-mail address or a Blob data type field. Click [...] in the value cell to set the link target. For details, see Binding a link to a label. Data type: String
Position	Specifies the position of the object. Choose an option from the drop-down list. • **static** - The object is positioned at the location in which it is inserted. If selected, the X, Y and other position-related properties will be hidden or disabled. The position of the object is affected by preceding objects in the same container. • **absolute** - The object will be located at the position specified by dragging and dropping or by setting its X and Y property values. The position of the object is not affected by preceding objects in the same container. Data type: Enumeration
Logic Column	Specifies whether to show the object in the next visible table cell in the same row when the column which holds the object is hidden. Choose an option from the drop-down list. Data type: Enumeration **Notes:** • The next visible table cell should be completely empty, that is, the cell should not hold any content including blank space. • When the property for several objects in the same row is set to next visible column, and the columns holding these objects are all hidden, then only the object in the rightmost column will be shown in the next visible cell.
Record Location	Specifies the calculation point for the properties which use formulas. Choose an option from the drop-down list. • **Default** - The values of the properties are calculated in the default location where they are located. • **Page Header** - The values of the properties are calculated in the page header. • **Page Footer** - The values of the properties are calculated in the page footer. Data type: Enumeration
Suppress	Specifies whether to show the object in the design area and in the report results. All formulas and calculations will be skipped if the property is set to true. Not supported in Web Report Studio. Data type: Boolean **Note:** When both the Invisible and Suppress properties of an object are set to true, Suppress has the higher priority.
Suppress When No Records	Specifies whether to display the object in the report results when no record is returned to its parent data container. Not supported in Web Report Studio. Data type: Boolean
Transfer Style	Specifies whether the style group of the primary report will be applied to the linked report if the object is linked to a report. Data type: Boolean

Table

The properties of a table in a web report are:

Property Name	Description
Color	
Background	Specifies the background color of the object. Choose a color from the drop-down list, enter a hexadecimal RGB value (for example, 0xff0000), or select Custom from the drop-down list to specify the color. Data type: String
CSS	
Class	Specifies a CSS class to be applied to the object which is a valid class in the CSS file. Data type: String
Style	The property can be used in two ways. • Specifies a style to be applied to the object. Choose a style from the drop-down list which is available when a Style Group has been selected at the report tab level and when there are styles in the Style Group that can be applied to the object, or type in the style name. • Specifies a CSS selector to be applied to the object. Enter a valid class name from the CSS file. Data type: String
Others	
Auto Expands	Specifies whether the objects in cells of the table will expand to their empty neighbouring cells horizontally when the actual size of the objects is bigger than the cells due to no enough column or page width for auto fit. This property requires that the property Auto Fit for the objects in the table cells is set to true. Data type: Boolean **Note:** The property does not apply to horizontal tables.
Default for Filter	Specifies whether to show the data component as the default component in the Apply to drop-down list of the Filter dialog in Web Report Studio. Data type: Boolean **Note:** In a web report, only one data component's Default for Filter property can be set to true.
Enable Navigation	Specifies whether to enable page navigation on the table. If it is set to true, when the web report is opened in Web Report Studio, if the table contains more than one page, a page navigation bar specific for the table will be available right below it. You can use the bar to view the desired pages. When this property is set to true, you can further specify the height and width of the page for the table to display multiple pages, by setting the two properties Navigation Height and Navigation Width. If later Web Report Studio users resize the table, the two properties will be reset to the resized values. Data type: Boolean
Export to Applet	Specifies whether to include the object when exporting the report to Applet. Data type: Boolean
Export to CSV	Specifies whether to include the object when exporting the report to CSV. Data type: Boolean
Export to Excel	Specifies whether to include the object when exporting the report to Excel. Data type: Boolean
Export to HTML	Specifies whether to include the object when exporting the report to HTML. Data type: Boolean
Export to PDF	Specifies whether to include the object when exporting the report to PDF. Data type: Boolean
Export to PostScript	Specifies whether to include the object when exporting the report to PostScript. Data type: Boolean
Export to Report Result	Specifies whether to include the object when exporting the report to Report Result. Data type: Boolean
Export to RTF	Specifies whether to include the object when exporting the report to RTF. Data type: Boolean
Export to Text	Specifies whether to include the object when exporting the report to Text. Data type: Boolean
Export to XML	Specifies whether to include the object when exporting the report to XML. Data type: Boolean
Invisible	Specifies whether to show the object in the design area and in the report results. All formulas and calculations will still be performed if the property is set to true. Data type: Boolean
Invisible for Filter Dialogs	Specifies whether to show the data component in the Apply to drop-down list of the Filter dialog in Web Report Studio. This property cannot be edited when Default for Filter is set to true. Data type: Boolean
Navigation Height	Specifies the height of the page for the table to display multiple pages. Enter a numeric value to specify the height. If no value is given here, JReport will calculate the table page height at runtime. This property takes effect when Enable Navigation is set to true. Data type: Float
Navigation Width	Specifies the width of the page for the table to display multiple pages. Enter a numeric value to specify the width. If no value is given here, JReport will calculate the table page width at runtime. This property takes effect when Enable Navigation is set to true. Data type: Float
Record Location	Specifies the calculation point for the properties which use formulas. Choose an option from the drop-down list. • **Default** - The values of the properties are calculated in the default location where they are located. • **Page Header** - The values of the properties are calculated in the page header. • **Page Footer** - The values of the properties are calculated in the page footer. Data type: Enumeration
Suppress	Specifies whether to show the object in the design area and in the report results. All formulas and calculations will be skipped if the property is set to true. Data type: Boolean **Note:** When both the Invisible and Suppress properties of an object are set to true, Suppress has the higher priority.
Suppress When No Records	Specifies whether to display the object in the report results when no record is returned to its parent data container. Data type: Boolean
Border	
Border Color	Specifies the color of the border of the object. Choose a color from the drop-down list, enter a hexadecimal RGB value (for example, 0xCCAA00), or select Custom from the drop-down list to specify the color. Data type: String
Border Thickness	Specifies the width of the border in inches or centimeters. Enter a numeric value to change the thickness. Data type: Float
Bottom Line	Specifies the line style of the bottom border of the object. Choose a style from the drop-down list. Data type: Enumeration
Left Line	Specifies the line style of the left border of the object. Choose a style from the drop-down list. Data type: Enumeration
Right Line	Specifies the line style of the right border of the object. Choose a style from the drop-down list. Data type: Enumeration
Top Line	Specifies the line style of the top border of the object. Choose a style from the drop-down list. Data type: Enumeration
TOC	
Anchor Display Value	Specifies a string or formula to display for the TOC entry for this object in the TOC Browser. By default it is the value of the specified object. The TOC Anchor property must be set to true for this property to take effect. Data type: String
TOC Anchor	Specifies whether to include the object in the TOC Browser for the report. Data type: Boolean

Note: Only these table properties can be rendered in Web Report Studio:

- Enable Navigation
- Invisible
- Navigation Height
- Navigation Width
- Properties of the table object in the Web Report Studio Inspector panel.

The following topics list properties of the children of a table:

- Table header row
- Table group
- Table footer row
- Table column
- Table cell

Table header row

The properties of a table header row are:

Property Name	Description
Color	
Background	Specifies the background color and fill effect of the object. Choose a color from the drop-down list, enter a hexadecimal RGB value (for example, 0xff0000), or select Custom from the drop-down list to specify the color and fill effect. Data type: String
CSS	
Class	Specifies a CSS class to be applied to the object which is a valid class in the CSS file. Data type: String
Style	The property can be used in two ways. • Specifies a style to be applied to the object. Choose a style from the drop-down list which is available when a Style Group has been selected at the report tab level and when there are styles in the Style Group that can be applied to the object, or type in the style name. • Specifies a CSS selector to be applied to the object. Enter a valid class name from the CSS file. Data type: String
Others	
Export to CSV	Specifies whether to include the object when exporting the report to CSV. Data type: Boolean
Export to Excel	Specifies whether to include the object when exporting the report to Excel. Data type: Boolean
Fill Whole Page	Specifies whether to make the row extended to the bottom of the page, so that the next row starts on a new page. Data type: Boolean
Invisible	Specifies whether to show the object in the design area and in the report results. All formulas and calculations will still be performed if the property is set to true. Data type: Boolean
On New Page	Specifies whether the row starts on a new page. This property is applied only when the report has page layout enabled. The default is false which means the row starts on a new page only if the previous page is filled. Data type: Boolean
Record Location	Specifies the calculation point for the properties which use formulas. Choose an option from the drop-down list. • **Default** - The values of the properties are calculated in the default location where they are located. • **Page Header** - The values of the properties are calculated in the page header. • **Page Footer** - The values of the properties are calculated in the page footer. Data type: Enumeration
Repeat Header for Excel	Specifies whether to make the object appear in every page of the report results when exporting the report to Excel. Data type: Boolean
Repeat Header for TXT	Specifies whether to make the object appear in every page of the report results when exporting the report to a .txt file. Data type: Boolean
Repeat Header for XML	Specifies whether to make the object appear in every page of the report results when exporting the report to XML. Data type: Boolean
Repeat on Pages	Specifies whether to make the object appear in every page of the report results. Data type: Boolean
Suppress	Specifies whether to show the object in the design area and in the report results. All formulas and calculations will be skipped if the property is set to true. Data type: Boolean **Note:** When both the Invisible and Suppress properties of an object are set to true, Suppress has the higher priority.
Suppress When No Records	Specifies whether to display the object in the report results when no record is returned to its parent data container. Data type: Boolean
Border	
Border Color	Specifies the color of the border of the object. Choose a color from the drop-down list, enter a hexadecimal RGB value (for example, 0xCCAA00), or select Custom from the drop-down list to specify the color. Data type: String
Border Thickness	Specifies the width of the border in inches or centimeters. Enter a numeric value to change the thickness. Data type: Float
Bottom Line	Specifies the line style of the bottom border of the object. Choose a style from the drop-down list. Data type: Enumeration
Left Line	Specifies the line style of the left border of the object. Choose a style from the drop-down list. Data type: Enumeration
Right Line	Specifies the line style of the right border of the object. Choose a style from the drop-down list. Data type: Enumeration
Top Line	Specifies the line style of the top border of the object. Choose a style from the drop-down list. Data type: Enumeration

Note: Only these table header properties can be rendered in Web Report Studio:

- Invisible
- Properties of the table header object in the Web Report Studio Inspector panel.

Table group

The properties of a table group are:

Property Name	Description
Others	
Current Block Index	The two properties, Current Block Index and Items per Block, work together to control the data of the object to be displayed in continuous page mode. Current Block Index specifies the index of the data block that will be displayed. 0 means the first block index, and 1 the second, and so on. Data type: Integer
Expand Detail Data	Specifies whether to expand the groups to show the details when the report is opened in Web Report Studio. Only works in continuous page mode. Data type: Boolean
Items per Block	The two properties, Current Block Index and Items per Block, work together to control the data of the object to be displayed in continuous page mode. Item per Block specifies the number of records in each data block. Data type: Integer
Group Layout	
Keep Group Together	Specifies whether to keep the whole group together. Data type: Boolean
TOC	
Anchor Display Value	Specifies a string or formula to display for the TOC entry for this object in the TOC Browser. By default it is the value of the specified object. The TOC Anchor property must be set to true for this property to take effect. Data type: String
TOC Anchor	Specifies whether to include the object in the TOC Browser for the report. A Boolean type value.
Security	
Cascade	Specifies whether to allow the security identifiers specified by the following three properties to view the group's detail or child groups. • **true** - The detail row, the group header row, and the group footer row will be displayed. • **false** - Only the group header and footer rows will be displayed. Data type: Boolean
Grant	Enables to set cached report bursting. The property accepts user names delimited by "|" as the value. You can specify which group of data can be viewed by which user. Data type: String
Groups	Enables to set cached report bursting. The property accepts group names delimited by "|" as the value. You can specify which group of data can be viewed by which group of users. Data type: String
Roles	Enables to set cached report bursting. The property accepts role names delimited by "|" as the value. You can specify which group of data can be viewed by which role. Data type: String

Note: All these table group properties are not supported in Web Report Studio.

A table group may contain the following objects:

- Table group header row
- Table detail row
- Table group footer row

Table group header row

The properties of a table group header row are:

Property Name	Description
Color	
Background	Specifies the background color and fill effect of the object. Choose a color from the drop-down list, enter a hexadecimal RGB value (for example, 0xff0000), or select Custom from the drop-down list to specify the color and fill effect. Data type: String
CSS	
Class	Specifies a CSS class to be applied to the object which is a valid class in the CSS file. Data type: String
Style	The property can be used in two ways. • Specifies a style to be applied to the object. Choose a style from the drop-down list which is available when a Style Group has been selected at the report tab level and when there are styles in the Style Group that can be applied to the object, or type in the style name. • Specifies a CSS selector to be applied to the object. Enter a valid class name from the CSS file. Data type: String
Others	
Export to CSV	Specifies whether to include the object when exporting the report to CSV. Data type: Boolean
Export to Excel	Specifies whether to include the object when exporting the report to Excel. Data type: Boolean
Fill Whole Page	Specifies whether to make the panel extended to the bottom of the page, so that the next panel starts on a new page. Data type: Boolean
Invisible	Specifies whether to show the object in the design area and in the report results. All formulas and calculations will still be performed if the property is set to true. Data type: Boolean
On New Page	Specifies whether the row starts on a new page. This property is applied only when the report has page layout enabled. The default is false which means the row starts on a new page only if the previous page is filled. Data type: Boolean
Record Location	Specifies the calculation point for the properties which use formulas. Choose an option from the drop-down list. • **Default** - The values of the properties are calculated in the default location where they are located. • **Page Header** - The values of the properties are calculated in the page header. • **Page Footer** - The values of the properties are calculated in the page footer. Data type: Enumeration
Repeat	Specifies whether to make the group header appear in every page. Data type: Boolean
Suppress	Specifies whether to show the object in the design area and in the report results. All formulas and calculations will be skipped if the property is set to true. Data type: Boolean **Note:** When both the Invisible and Suppress properties of an object are set to true, Suppress has the higher priority.
Suppress When No Records	Specifies whether to display the object in the report results when no record is returned to its parent data container. Data type: Boolean
Border	
Border Color	Specifies the color of the border of the object. Choose a color from the drop-down list, enter a hexadecimal RGB value (for example, 0xCCAA00), or select Custom from the drop-down list to specify the color. Data type: String
Border Thickness	Specifies the width of the border in inches or centimeters. Enter a numeric value to change the thickness. Data type: Float
Bottom Line	Specifies the line style of the bottom border of the object. Choose a style from the drop-down list. Data type: Enumeration
Left Line	Specifies the line style of the left border of the object. Choose a style from the drop-down list. Data type: Enumeration
Right Line	Specifies the line style of the right border of the object. Choose a style from the drop-down list. Data type: Enumeration
Top Line	Specifies the line style of the top border of the object. Choose a style from the drop-down list. Data type: Enumeration

Table detail row

The properties of a table detail row are:

Property Name	Description
Color	
Background	Specifies the background color and fill effect of the object. Choose a color from the drop-down list, enter a hexadecimal RGB value (for example, 0xff0000), or select Custom from the drop-down list to specify the color and fill effect. Data type: String
CSS	
Class	Specifies a CSS class to be applied to the object which is a valid class in the CSS file. Data type: String
Style	The property can be used in two ways. • Specifies a style to be applied to the object. Choose a style from the drop-down list which is available when a Style Group has been selected at the report tab level and when there are styles in the Style Group that can be applied to the object, or type in the style name. • Specifies a CSS selector to be applied to the object. Enter a valid class name from the CSS file. Data type: String
Others	
Export to CSV	Specifies whether to include the object when exporting the report to CSV. Data type: Boolean
Export to Excel	Specifies whether to include the object when exporting the report to Excel. Data type: Boolean
Invisible	Specifies whether to show the object in the design area and in the report results. All formulas and calculations will still be performed if the property is set to true. Data type: Boolean
Record Location	Specifies the calculation point for the properties which use formulas. Choose an option from the drop-down list. • **Default** - The values of the properties are calculated in the default location where they are located. • **Page Header** - The values of the properties are calculated in the page header. • **Page Footer** - The values of the properties are calculated in the page footer. Data type: Enumeration
Suppress	Specifies whether to show the object in the design area and in the report results. All formulas and calculations will be skipped if the property is set to true. Data type: Boolean **Note:** When both the Invisible and Suppress properties of an object are set to true, Suppress has the higher priority.
Suppress When No Records	Specifies whether to display the object in the report results when no record is returned to its parent data container. Data type: Boolean
Border	
Border Color	Specifies the color of the border of the object. Choose a color from the drop-down list, enter a hexadecimal RGB value (for example, 0xCCAA00), or select Custom from the drop-down list to specify the color. Data type: String
Border Thickness	Specifies the width of the border in inches or centimeters. Enter a numeric value to change the thickness. Data type: Float
Bottom Line	Specifies the line style of the bottom border of the object. Choose a style from the drop-down list Data type: Enumeration
Left Line	Specifies the line style of the left border of the object. Choose a style from the drop-down list. Data type: Enumeration
Right Line	Specifies the line style of the right border of the object. Choose a style from the drop-down list. Data type: Enumeration
Top Line	Specifies the line style of the top border of the object. Choose a style from the drop-down list. Data type: Enumeration

Table group footer row

The properties of a table group footer row are:

Property Name	Description
Color	
Background	Specifies the background color and fill effect of the object. Choose a color from the drop-down list, enter a hexadecimal RGB value (for example, 0xff0000), or select Custom from the drop-down list to specify the color and fill effect. Data type: String
CSS	
Class	Specifies a CSS class to be applied to the object which is a valid class in the CSS file. Data type: String
Style	The property can be used in two ways. • Specifies a style to be applied to the object. Choose a style from the drop-down list which is available when a Style Group has been selected at the report tab level and when there are styles in the Style Group that can be applied to the object, or type in the style name. • Specifies a CSS selector to be applied to the object. Enter a valid class name from the CSS file. Data type: String
Others	
Export to CSV	Specifies whether to include the object when exporting the report to CSV. Data type: Boolean
Export to Excel	Specifies whether to include the object when exporting the report to Excel. Data type: Boolean
Fill Whole Page	Specifies whether to make the row extended to the bottom of the page, so that the next row starts on a new page. Data type: Boolean
Invisible	Specifies whether to show the object in the design area and in the report results. All formulas and calculations will still be performed if the property is set to true. Data type: Boolean
On New Page	Specifies whether the row starts on a new page. This property is applied only when the report has page layout enabled. The default is false which means the row starts on a new page only if the previous page is filled. Data type: Boolean
Record Location	Specifies the calculation point for the properties which use formulas. Choose an option from the drop-down list. • **Default** - The values of the properties are calculated in the default location where they are located. • **Page Header** - The values of the properties are calculated in the page header. • **Page Footer** - The values of the properties are calculated in the page footer. Data type: Enumeration
Suppress	Specifies whether to show the object in the design area and in the report results. All formulas and calculations will be skipped if the property is set to true. Data type: Boolean **Note:** When both the Invisible and Suppress properties of an object are set to true, Suppress has the higher priority.
Suppress When No Records	Specifies whether to display the object in the report results when no record is returned to its parent data container. Data type: Boolean
Border	
Border Color	Specifies the color of the border of the object. Choose a color from the drop-down list, enter a hexadecimal RGB value (for example, 0xCCAA00), or select Custom from the drop-down list to specify the color. Data type: String
Border Thickness	Specifies the width of the border in inches or centimeters. Enter a numeric value to change the thickness. Data type: Float
Bottom Line	Specifies the line style of the bottom border of the object. Choose a style from the drop-down list. Data type: Enumeration
Left Line	Specifies the line style of the left border of the object. Choose a style from the drop-down list. Data type: Enumeration
Right Line	Specifies the line style of the right border of the object. Choose a style from the drop-down list. Data type: Enumeration
Top Line	Specifies the line style of the top border of the object. Choose a style from the drop-down list. Data type: Enumeration

Table footer row

The properties of a table footer row are:

Property Name	Description
Color	
Background	Specifies the background color and fill effect of the object. Choose a color from the drop-down list, enter a hexadecimal RGB value (for example, 0xff0000), or select Custom from the drop-down list to specify the color and fill effect. Data type: String
CSS	
Class	Specifies a CSS class to be applied to the object which is a valid class in the CSS file. Data type: String
Style	The property can be used in two ways. • Specifies a style to be applied to the object. Choose a style from the drop-down list which is available when a Style Group has been selected at the report tab level and when there are styles in the Style Group that can be applied to the object, or type in the style name. • Specifies a CSS selector to be applied to the object. Enter a valid class name from the CSS file. Data type: String
Others	
Export to CSV	Specifies whether to include the object when exporting the report to CSV. Data type: Boolean
Export to Excel	Specifies whether to include the object when exporting the report to Excel. Data type: Boolean
Fill Whole Page	Specifies whether to make the row extended to the bottom of the page, so that the next row starts on a new page. Data type: Boolean
Invisible	Specifies whether to show the object in the design area and in the report results. All formulas and calculations will still be performed if the property is set to true. Data type: Boolean
On New Page	Specifies whether the row starts on a new page. This property is applied only when the report has page layout enabled. The default is false which means the row starts on a new page only if the previous page is filled. Data type: Boolean
Record Location	Specifies the calculation point for the properties which use formulas. Choose an option from the drop-down list. • **Default** - The values of the properties are calculated in the default location where they are located. • **Page Header** - The values of the properties are calculated in the page header. • **Page Footer** - The values of the properties are calculated in the page footer. Data type: Enumeration
Suppress	Specifies whether to show the object in the design area and in the report results. All formulas and calculations will be skipped if the property is set to true. Data type: Boolean **Note:** When both the Invisible and Suppress properties of an object are set to true, Suppress has the higher priority.
Suppress When No Records	Specifies whether to display the object in the report results when no record is returned to its parent data container. Data type: Boolean
Border	
Border Color	Specifies the color of the border of the object. Choose a color from the drop-down list, enter a hexadecimal RGB value (0xCCAA00), or select Custom from the drop-down list to specify the color. Data type: String
Border Thickness	Specifies the width of the border in inches or centimeters. Enter a numeric value to change the thickness. Data type: Float
Bottom Line	Specifies the line style of the bottom border of the object. Choose a style from the drop-down list. Data type: Enumeration
Left Line	Specifies the line style of the left border of the object. Choose a style from the drop-down list. Data type: Enumeration
Right Line	Specifies the line style of the right border of the object. Choose a style from the drop-down list. Data type: Enumeration
Top Line	Specifies the line style of the top border of the object. Choose a style from the drop-down list. Data type: Enumeration

Note: Only these table footer properties can be rendered in Web Report Studio:

- Invisible
- Properties of the table footer object in the Web Report Studio Inspector panel.

Table column

There are four types of columns in a table, which are common column, detail column, group column and summary column. These table column objects have the same properties as follows:

Property Name	Description
Auto Fit	Specifies whether to adjust the width of the object according to the contents. If the property is set to true, the width of the object will be adjusted automatically only when: • The Position property for the field/label in this object is static. • If the Position property for the field/label in this object is absolute, then the Auto Fit property for the field/label should be set to true. Data type: Boolean
Invisible	Specifies whether to show the object in the design area and in the report results. All formulas and calculations will still be performed if the property is set to true. Data type: Boolean

Table cell

The properties of a table cell are:

Property Name	Description
Geometry	
Height	Specifies the height of the object, in inches or centimeters. Enter a numeric value to change the height. Data type: Float
Width	Specifies the width of the object, in inches or centimeters. Enter a numeric value to change the width.
Color	
Background	Specifies the background color of the object. Choose a color from the drop-down list, enter a hexadecimal RGB value (for example, 0xff0000), or select Custom from the drop-down list to specify the color. Data type: String
CSS	
Class	Specifies a CSS class to be applied to the object which is a valid class in the CSS file. Data type: String
Style	The property can be used in two ways. • Specifies a style to be applied to the object. Choose a style from the drop-down list which is available when a Style Group has been selected at the report tab level and when there are styles in the Style Group that can be applied to the object, or type in the style name. • Specifies a CSS selector to be applied to the object. Enter a valid class name from the CSS file. Data type: String
Others	
Export to CSV	Specifies whether to include the object when exporting the report to CSV. Data type: Boolean
Export to Excel	Specifies whether to include the object when exporting the report to Excel. Data type: Boolean
Horizontal Alignment	Specifies the horizontal justification of the content in the table cell. Choose an option from the drop-down list. Data type: Enumeration **Note:** When the Position property of the object in the table cell is set to absolute, this property does not take effect.
Joining Merge	Specifies whether to make all the sequential rows included in the cell vertically merged in the report results. It is meaningful to set the property only when the cell is merged with other cells in the same column. Data type: Boolean
Record Location	Specifies the calculation point for the properties which use formulas. Choose an option from the drop-down list. • **Default** - The values of the properties are calculated in the default location where they are located. • **Page Header** - The values of the properties are calculated in the page header. • **Page Footer** - The values of the properties are calculated in the page footer. Data type: Enumeration
Repeat	Specifies whether to repeat the content of the cell in every page of the report results when the cell is split into pages. Available only for vertically merged cell. Data type: Boolean
Suppress When No Records	Specifies whether to display the object in the report results when no record is returned to its parent data container. Data type: Boolean
Vertical Alignment	Specifies the vertical justification of the content in the table cell. Choose an option from the drop-down list. Data type: Enumeration **Note:** When the Position property of the object in the table cell is set to absolute, this property does not take effect.
Border	
Border Color	Specifies the color of the border of the object. Choose a color from the drop-down list, enter a hexadecimal RGB value (for example, 0xCCAA00), or select Custom from the drop-down list to specify the color. Data type: String
Border Thickness	Specifies the width of the border in inches or centimeters. Enter a numeric value to change the thickness. Data type: Float
Bottom Line	Specifies the line style of the bottom border of the object. Choose a style from the drop-down list. Data type: Enumeration
Left Line	Specifies the line style of the left border of the object. Choose a style from the drop-down list. Data type: Enumeration
Right Line	Specifies the line style of the right border of the object. Choose a style from the drop-down list. Data type: Enumeration
Top Line	Specifies the line style of the top border of the object. Choose a style from the drop-down list. Data type: Enumeration

Note: Only properties of the table cell object in the Web Report Studio Inspector panel can be rendered.

Tabular

The properties of a tabular in a web report are:

Property Name	Description
Color	
Background	Specifies the background color of the object. Choose a color from the drop-down list, enter a hexadecimal RGB value (for example, 0xff0000), or select Custom from the drop-down list to specify the color. Data type: String
CSS	
Class	Specifies a CSS class to be applied to the object which is a valid class in the CSS file. Data type: String
Style	The property can be used in two ways. • Specifies a style to be applied to the object. Choose a style from the drop-down list which is available when a Style Group has been selected at the report tab level and when there are styles in the Style Group that can be applied to the object, or type in the style name. • Specifies a CSS selector to be applied to the object. Enter a valid class name from the CSS file. Data type: String
Others	
Export to Applet	Specifies whether to include the object when exporting the report to Applet. Data type: Boolean
Export to CSV	Specifies whether to include the object when exporting the report to CSV. Data type: Boolean
Export to Excel	Specifies whether to include the object when exporting the report to Excel. Data type: Boolean
Export to HTML	Specifies whether to include the object when exporting the report to HTML. Data type: Boolean
Export to PDF	Specifies whether to include the object when exporting the report to PDF. Data type: Boolean
Export to PostScript	Specifies whether to include the object when exporting the report to PostScript. Data type: Boolean
Export to Report Result	Specifies whether to include the object when exporting the report to Report Result. Data type: Boolean
Export to RTF	Specifies whether to include the object when exporting the report to RTF. Data type: Boolean
Export to Text	Specifies whether to include the object when exporting the report to Text. Data type: Boolean
Export to XML	Specifies whether to include the object when exporting the report to XML. Data type: Boolean
Horizontal Auto Size	Specifies whether to make the width of the object automatically expanded according to the size of the components inserted. Data type: Boolean
Invisible	Specifies whether to show the object in the design area and in the report results. All formulas and calculations will still be performed if the property is set to true. Data type: Boolean
Position	Specifies the position of the object. Choose an option from the drop-down list. • **static** - The object is positioned at the location in which it is inserted. If selected, the X, Y and other position-related properties will be hidden or disabled. The position of the object is affected by preceding objects in the same container. • **absolute** - The object will be located at the position specified by dragging and dropping or by setting its X and Y property values. The position of the object is not affected by preceding objects in the same container. Data type: Enumeration
Suppress	Specifies whether to show the object in the design area and in the report results. All formulas and calculations will be skipped if the property is set to true. Data type: Boolean **Note:** When both the Invisible and Suppress properties of an object are set to true, Suppress has the higher priority.
Vertical Auto Size	Specifies whether to make the height of the object automatically expanded according to the size of the components inserted. Data type: Boolean
Excel	
Column Index	Specifies the X coordinate of the object relative to its parent container when exported to Excel or CSV, measured in cells. The Columned property at the report level must be set to true for this property to take effect. Data type: Integer
Row Index	Specifies the Y coordinate of the object relative to its parent container when exported to Excel or CSV, measured in cells. The Columned property at the report level must be set to true for this property to take effect. Data type: Integer
TOC	
Anchor Display Value	Specifies a string or formula to display for the TOC entry for this object in the TOC Browser. By default it is the value of the specified object. The TOC Anchor property must be set to true for this property to take effect. Data type: String
TOC Anchor	Specifies whether to include the object in the TOC Browser for the report. Data type: Boolean

A tabular may contain these objects:

- Tabular row group
- Tabular cell

Tabular row group

The properties of a tabular row group in a web report are:

Property Name	Description
Cross Page	Specifies whether the row can be split across a page break. This property is applied only when the report has page layout enabled. Data type: Boolean
Fill Whole Page	Specifies whether to make the row extended to the bottom of the page, so that the next panel starts on a new page. Data type: Boolean
On New Page	Specifies whether the row starts on a new page. This property is only applied if the report has page layout enabled. The default is false which means the panel starts on a new page only if the previous page is filled. Data type: Boolean

Tabular cell

The properties of a tabular cell in a web report are:

Property Name	Description
Geometry	
Height	Specifies the height of the object, in inches or centimeters. Enter a numeric value to change the height. Data type: Float
Width	Specifies the width of the object, in inches or centimeters. Enter a numeric value to change the width. Data type: Float
Color	
Background	Specifies the background color of the object. Choose a color from the drop-down list, enter a hexadecimal RGB value (for example, 0xff0000), or select Custom from the drop-down list to specify the color. Data type: String
CSS	
Class	Specifies a CSS class to be applied to the object which is a valid class in the CSS file. Data type: String
Style	The property can be used in two ways. • Specifies a style to be applied to the object. Choose a style from the drop-down list which is available when a Style Group has been selected at the report tab level and when there are styles in the Style Group that can be applied to the object, or type in the style name. • Specifies a CSS selector to be applied to the object. Enter a valid class name from the CSS file. Data type: String
Others	
Horizontal Alignment	Specifies the horizontal justification of the contents in the tabular cell. Choose an option from the drop-down list. Data type: Enumeration
Repeat Content	A cell may be extended to new pages for being affected by other cells even though it can be wholly put in the current page. In this case, if the property value is true, then its contents will appear on the new pages again. Data type: Boolean **Note:** If there are two or more cells that will be affected in the same row group and the values of the property for these cells are true, then only the contents of the last one will be repeated.
Vertical Alignment	Specifies the vertical justification of the contents in the tabular cell. Applied only when the web report runs in Web Report Studio. Choose an option from the drop-down list. Data type: Enumeration
TOC	
Anchor Display Value	Specifies a string or formula to display for the TOC entry for this object in the TOC Browser. By default it is the value of the specified object. The TOC Anchor property must be set to true for this property to take effect. Data type: String
TOC Anchor	Specifies whether to include the object in the TOC Browser for the report. Data type: Boolean

TOC

The properties of a TOC object in a web report are:

Property Name	Description
General	
Root Label	Specifies the text of the label which will be displayed on the root of the TOC tree. Data type: String
Color	
Background	Specifies the background color of the object. Choose a color from the drop-down list, enter a hexadecimal RGB value (for example, 0xff0000), or select Custom from the drop-down list to specify the color. Data type: String
Foreground	Specifies the foreground color of the object. Choose a color from the drop-down list, enter a hexadecimal RGB value (for example, 0xff0000), or select Custom from the drop-down list to specify the color. Data type: String
CSS	
Class	Specifies a CSS class to be applied to the object which is a valid class in the CSS file. Data type: String
Style	The property can be used in two ways. • Specifies a style to be applied to the object. Choose a style from the drop-down list which is available when a Style Group has been selected at the report tab level and when there are styles in the Style Group that can be applied to the object, or type in the style name. • Specifies a CSS selector to be applied to the object. Enter a valid class name from the CSS file. Data type: String
Font	
Bold	Specifies whether to make the text bold. Data type: Boolean
Font Face	Specifies the font of the text. Choose an option from the drop-down list. Data type: Enumeration
Font Size	Specifies the font size of the text. Enter an integer value to change the size. Data type: Integer
Italic	Specifies whether to make the text italic. Data type: Boolean
Strikethrough	Specifies whether to add a strikethrough line to the text. Data type: Boolean
Underline	Specifies whether to underline the text. Data type: Boolean

Web control

The properties of web controls in a web report are listed as follows:

- Filter control
- Navigation control
- Parameter control
- Parameter form control

Filter control

The properties of a filter control are:

Property Name	Description
Geometry	
Height	Specifies the height of the object, in inches or centimeters. Enter a numeric value to change the height. Data type: Float
Width	Specifies the width of the object, in inches or centimeters. Enter a numeric value to change the width. Data type: Float
Color	
Background	Specifies the background color of the object. Choose a color from the drop-down list, enter a hexadecimal RGB value (for example, 0xff0000), or select Custom from the drop-down list to specify the color. Data type: String
Foreground	Specifies the foreground color of the object. Choose a color from the drop-down list, enter a hexadecimal RGB value (for example, 0xff0000), or select Custom from the drop-down list to specify the color. Data type: String
CSS	
Class	Specifies a CSS class to be applied to the object which is a valid class in the CSS file. Data type: String
ID	Specifies the identifier of the object, which must be unique in the report. The ID property can be a style sheet selector. Data type: String
Style	The property can be used in two ways. • Specifies a style to be applied to the object. Choose a style from the drop-down list which is available when a Style Group has been selected at the report tab level and when there are styles in the Style Group that can be applied to the object, or type in the style name. • Specifies a CSS selector to be applied to the object. Enter a valid class name from the CSS file. Data type: String
Others	
Export to Applet	Specifies whether to include the object when exporting the report to Applet. Data type: Boolean
Export to CSV	Specifies whether to include the object when exporting the report to CSV. Data type: Boolean
Export to Excel	Specifies whether to include the object when exporting the report to Excel. Data type: Boolean
Export to HTML	Specifies whether to include the object when exporting the report to HTML. Data type: Boolean
Export to PDF	Specifies whether to include the object when exporting the report to PDF. Data type: Boolean
Export to PostScript	Specifies whether to include the object when exporting the report to PostScript. Data type: Boolean
Export to Report Result	Specifies whether to include the object when exporting the report to Report Result. Data type: Boolean
Export to RTF	Specifies whether to include the object when exporting the report to RTF. Data type: Boolean
Export to Text	Specifies whether to include the object when exporting the report to Text. Data type: Boolean
Export to XML	Specifies whether to include the object when exporting the report to XML. Data type: Boolean
Invisible	Specifies whether to show the object in the design area and in the report results. All formulas and calculations will still be performed if the property is set to true. Data type: Boolean
Excel	
Column Index	Specifies the X coordinate of the object relative to its parent container when exported to Excel or CSV, measured in cells. The Columned property at the report level must be set to true for this property to take effect. Data type: Integer
Row Index	Specifies the Y coordinate of the object relative to its parent container when exported to Excel or CSV, measured in cells. The Columned property at the report level must be set to true for this property to take effect. Data type: Integer
Border	
Border Color	Specifies the color of the border of the object. Choose a color from the drop-down list, enter a hexadecimal RGB value (for example, 0xCCAA00), or select Custom from the drop-down list to specify the color. Data type: String
Border Thickness	Specifies the width of the border in inches or centimeters. Enter a numeric value to change the thickness. Data type: Float
Bottom Line	Specifies the line style of the bottom border of the object. Choose a style from the drop-down list. Data type: Enumeration
Left Line	Specifies the line style of the left border of the object. Choose a style from the drop-down list. Data type: Enumeration
Right Line	Specifies the line style of the right border of the object. Choose a style from the drop-down list. Data type: Enumeration
Top Line	Specifies the line style of the top border of the object. Choose a style from the drop-down list. Data type: Enumeration
Title	
Background	Specifies the background color of the title bar. Choose a color from the drop-down list, enter a hexadecimal RGB value (for example, 0xff0000), or select Custom from the drop-down list to specify the color. Data type: String
Bold	Specifies whether to make the text bold in the title bar. Data type: Boolean
Font Face	Specifies the font of the text in the title bar. Choose an option from the drop-down list. Data type: Enumeration
Font Size	Specifies the font size of the text in the title bar. Enter an integer value to change the size. Data type: Integer
Foreground	Specifies the foreground color of the title bar. Choose a color from the drop-down list, enter a hexadecimal RGB value (for example, 0xff0000), or select Custom from the drop-down list to specify the color. Data type: String
Horizontal Alignment	Specifies the horizontal justification of the text in the title bar. Choose an option from the drop-down list. Data type: Enumeration
Italic	Specifies whether to make the text italic in the title bar. Data type: Boolean
Map Name	Specifies whether to use the name of the DBField added to the filter control as the text displayed in the title bar. Data type: Boolean
Show Title	Specifies whether to show the title bar. It is meaningful to set all the other properties in the Title category when this property is set to true. Data type: Boolean
Text	Specifies the text displayed in the title bar. This property can be edited only when Map Name is set to false. Enter a string to change the text. Data type: String
Underline	Specifies whether to underline the text in the title bar. Data type: Boolean
Text Format	
Bold	Specifies whether to make the text bold. Data type: Boolean
Font Face	Specifies the font of the text. Choose an option from the drop-down list. Data type: Enumeration
Font Size	Specifies the font size of the text. Enter an integer value to change the size. Data type: Integer
Format	Specifies the display format of the text in the report result. It varies with data type and can be manually defined. Data type: String **Note:** For the BigDecimal type, to avoid precision loss, you should enter a prefix JRD when setting the format property value.
Italic	Specifies whether to make the text italic. Data type: Boolean
Underline	Specifies whether to underline the text. Data type: Boolean

Navigation control

The properties of a navigation control are:

Property Name	Description
Geometry	
Height	Specifies the height of the object, in inches or centimeters. Enter a numeric value to change the height. Data type: Float
Width	Specifies the width of the object, in inches or centimeters. Enter a numeric value to change the width. Data type: Float
Color	
Background	Specifies the background color of the object. Choose a color from the drop-down list, enter a hexadecimal RGB value (for example, 0xff0000), or select Custom from the drop-down list to specify the color. Data type: String
CSS	
Class	Specifies a CSS class to be applied to the object which is a valid class in the CSS file. Data type: String
Style	The property can be used in two ways. • Specifies a style to be applied to the object. Choose a style from the drop-down list which is available when a Style Group has been selected at the report tab level and when there are styles in the Style Group that can be applied to the object, or type in the style name. • Specifies a CSS selector to be applied to the object. Enter a valid class name from the CSS file. Data type: String
Others	
Export to Applet	Specifies whether to include the object when exporting the report to Applet. Data type: Boolean
Export to CSV	Specifies whether to include the object when exporting the report to CSV. Data type: Boolean
Export to Excel	Specifies whether to include the object when exporting the report to Excel. Data type: Boolean
Export to HTML	Specifies whether to include the object when exporting the report to HTML. Data type: Boolean
Export to PDF	Specifies whether to include the object when exporting the report to PDF. Data type: Boolean
Export to PostScript	Specifies whether to include the object when exporting the report to PostScript. Data type: Boolean
Export to Report Result	Specifies whether to include the object when exporting the report to Report Result. Data type: Boolean
Export to RTF	Specifies whether to include the object when exporting the report to RTF. Data type: Boolean
Export to Text	Specifies whether to include the object when exporting the report to Text. Data type: Boolean
Export to XML	Specifies whether to include the object when exporting the report to XML. Data type: Boolean
Invisible	Specifies whether to show the object in the design area and in the report results. All formulas and calculations will still be performed if the property is set to true. Data type: Boolean
Excel	
Column Index	Specifies the X coordinate of the object relative to its parent container when exported to Excel or CSV, measured in cells. The Columned property at the report level must be set to true for this property to take effect. Data type: Integer
Row Index	Specifies the Y coordinate of the object relative to its parent container when exported to Excel or CSV, measured in cells. The Columned property at the report level must be set to true for this property to take effect. Data type: Integer
Border	
Border Color	Specifies the color of the border of the object. Choose a color from the drop-down list, enter a hexadecimal RGB value (for example, 0xCCAA00), or select Custom from the drop-down list to specify the color. Data type: String
Border Thickness	Specifies the width of the border in inches or centimeters. Enter a numeric value to change the thickness. Data type: Float
Bottom Line	Specifies the line style of the bottom border of the object. Choose a style from the drop-down list. Data type: Enumeration
Left Line	Specifies the line style of the left border of the object. Choose a style from the drop-down list. Data type: Enumeration
Right Line	Specifies the line style of the right border of the object. Choose a style from the drop-down list. Data type: Enumeration
Top Line	Specifies the line style of the top border of the object. Choose a style from the drop-down list. Data type: Enumeration

A navigation control has a child object:

- Navigation item

Navigation item

The properties of a navigation item are:

Property Name	Description
Text Format	
Auto Fit	Specifies whether to adjust the width and height of the object according to the contents. Data type: Boolean
Bold	Specifies whether to make the text bold. Data type: Boolean
Font Face	Specifies the font of the text. Choose an option from the drop-down list. Data type: Enumeration
Font Size	Specifies the font size of the text. Enter an integer value to change the size. Data type: Integer
Horizontal Alignment	Specifies the horizontal justification of the text in the object. Choose an option from the drop-down list. Data type: Enumeration
Italic	Specifies whether to make the text italic. Data type: Boolean
Strikethrough	Specifies whether to add a strikethrough line to the text. Data type: Boolean
Text	Specifies the text in the object. Enter a string to display as the object text. Data type: String
Underline	Specifies whether to underline the text. Data type: Boolean
Vertical Alignment	Specifies the vertical justification of the text in the object. Choose an option from the drop-down list. Data type: Enumeration
Word Wrap	Specifies whether to wrap the text to the width. Data type: Boolean
Color	
Background	Specifies the background color of the object. Choose a color from the drop-down list, enter a hexadecimal RGB value (for example, 0xff0000), or select Custom from the drop-down list to specify the color. Data type: String
Foreground	Specifies the foreground color of the object. Choose a color from the drop-down list, enter a hexadecimal RGB value (for example, 0xff0000), or select Custom from the drop-down list to specify the color. Data type: String
Others	
Export to Applet	Specifies whether to include the object when exporting the report to Applet. Data type: Boolean
Export to CSV	Specifies whether to include the object when exporting the report to CSV. Data type: Boolean
Export to Excel	Specifies whether to include the object when exporting the report to Excel. Data type: Boolean
Export to HTML	Specifies whether to include the object when exporting the report to HTML. Data type: Boolean
Export to PDF	Specifies whether to include the object when exporting the report to PDF. Data type: Boolean
Export to PostScript	Specifies whether to include the object when exporting the report to PostScript. Data type: Boolean
Export to Report Result	Specifies whether to include the object when exporting the report to Report Result. Data type: Boolean
Export to RTF	Specifies whether to include the object when exporting the report to RTF. Data type: Boolean
Export to Text	Specifies whether to include the object when exporting the report to Text. Data type: Boolean
Export to XML	Specifies whether to include the object when exporting the report to XML. Data type: Boolean
Padding	
Bottom Padding	Specifies the space in inches or centimeters between the text and the bottom border of the object. Enter a numeric value to change the padding. Data type: Float
Left Padding	Specifies the space in inches or centimeters between the text and the left border of the object. Enter a numeric value to change the padding. Data type: Float
Right Padding	Specifies the space in inches or centimeters between the text and the right border of the object. Enter a numeric value to change the padding. Data type: Float
Top Padding	Specifies the space in inches or centimeters between the text and the top border of the object. Enter a numeric value to change the padding. Data type: Float
Border	
Border Color	Specifies the color of the border of the object. Choose a color from the drop-down list, enter a hexadecimal RGB value (for example, 0xCCAA00), or select Custom from the drop-down list to specify the color. Data type: String
Border Thickness	Specifies the width of the border in inches or centimeters. Enter a numeric value to change the thickness. Data type: Float
Bottom Line	Specifies the line style of the bottom border of the object. Choose a style from the drop-down list. Data type: Enumeration
Left Line	Specifies the line style of the left border of the object. Choose a style from the drop-down list. Data type: Enumeration
Right Line	Specifies the line style of the right border of the object. Choose a style from the drop-down list. Data type: Enumeration
Shadow	Specifies whether to draw two shadowed borders, beneath and to the right of the object. Data type: Boolean
Shadow Color	Specifies the color of the shadow. Choose a color from the drop-down list, enter a hexadecimal RGB value (for example, 0xff0000), or select Custom from the drop-down list to specify the color and fill effect. Data type: String
Top Line	Specifies the line style of the top border of the object. Choose a style from the drop-down list. Data type: Enumeration
Pattern	
Pattern Color	Specifies the color in which to draw a pattern to fill the object. Choose a color from the drop-down list, enter a hexadecimal RGB value (for example, 0xff0000), or select Custom from the drop-down list to specify the color and fill effect. Data type: String
Pattern Style	Specifies the style of the pattern. Choose an option from the drop-down list. • **none** - No pattern will be applied to the object. • **50**% - The object will be filled in the Pattern Color of 50%-transparency. • **horizontal** - The pattern will be made of horizontal lines of the Pattern Color. • **vertical** - The pattern will be made of vertical lines of the Pattern Color. • **grid** - The pattern will be an overlapping of horizontal and vertical lines of the Pattern Color. • **diagonal** - The pattern will be made of diagonal lines of the Pattern Color. Data type: Enumeration

Parameter control

The properties of a parameter control are:

Property Name	Description
Geometry	
Height	Specifies the height of the object, in inches or centimeters. Enter a numeric value to change the height. Data type: Float
Width	Specifies the width of the object, in inches or centimeters. Enter a numeric value to change the width. Data type: Float
Color	
Background	Specifies the background color of the object. Choose a color from the drop-down list, enter a hexadecimal RGB value (for example, 0xff0000), or select Custom from the drop-down list to specify the color. Data type: String
Foreground	Specifies the foreground color of the object. Choose a color from the drop-down list, enter a hexadecimal RGB value (for example, 0xff0000), or select Custom from the drop-down list to specify the color. Data type: String
CSS	
Class	Specifies a CSS class to be applied to the object which is a valid class in the CSS file. Data type: String
ID	Specifies the identifier of the object, which must be unique in the report. The ID property can be a style sheet selector. Data type: String
Style	The property can be used in two ways. • Specifies a style to be applied to the object. Choose a style from the drop-down list which is available when a Style Group has been selected at the report tab level and when there are styles in the Style Group that can be applied to the object, or type in the style name. • Specifies a CSS selector to be applied to the object. Enter a valid class name from the CSS file. Data type: String
Excel	
Column Index	Specifies the X coordinate of the object relative to its parent container when exported to Excel or CSV, measured in cells. The Columned property at the report level must be set to true for this property to take effect. Data type: Integer
Row Index	Specifies the Y coordinate of the object relative to its parent container when exported to Excel or CSV, measured in cells. The Columned property at the report level must be set to true for this property to take effect. Data type: Integer
Border	
Border Color	Specifies the color of the border of the object. Choose a color from the drop-down list, enter a hexadecimal RGB value (for example, 0xCCAA00), or select Custom from the drop-down list to specify the color. Data type: String
Border Thickness	Specifies the width of the border in inches or centimeters. Enter a numeric value to change the thickness. Data type: Float
Bottom Line	Specifies the line style of the bottom border of the object. Choose a style from the drop-down list. Data type: Enumeration
Left Line	Specifies the line style of the left border of the object. Choose a style from the drop-down list. Data type: Enumeration
Right Line	Specifies the line style of the right border of the object. Choose a style from the drop-down list. Data type: Enumeration
Shadow	Specifies whether to draw two shadowed borders, beneath and to the right of the object. Data type: Boolean
Shadow Color	Specifies the color of the shadow. Choose a color from the drop-down list, enter a hexadecimal RGB value (for example, 0xff0000), or select Custom from the drop-down list to specify the color and fill effect. Data type: String
Top Line	Specifies the line style of the top border of the object. Choose a style from the drop-down list. Data type: Enumeration
Pattern	
Pattern Color	Specifies the color in which to draw a pattern to fill the object. Choose a color from the drop-down list, enter a hexadecimal RGB value (for example, 0xff0000), or select Custom from the drop-down list to specify the color and fill effect. Data type: String
Pattern Style	Specifies the style of the pattern. Choose an option from the drop-down list. • **none** - No pattern will be applied to the object. • **50**% - The object will be filled in the Pattern Color of 50%-transparency. • **horizontal** - The pattern will be made of horizontal lines of the Pattern Color. • **vertical** - The pattern will be made of vertical lines of the Pattern Color. • **grid** - The pattern will be an overlapping of horizontal and vertical lines of the Pattern Color. • **diagonal** - The pattern will be made of diagonal lines of the Pattern Color. Data type: Enumeration
Text Format	
Bold	Specifies whether to make the text bold. Data type: Boolean
Font Face	Specifies the font of the text. Choose an option from the drop-down list. Data type: Enumeration
Font Size	Specifies the font size of the text. Enter an integer value to change the size. Data type: Integer
Format	Specifies the display format of the text in the report result. It varies with data type and can be manually defined. Data type: String **Note:** For the BigDecimal type, to avoid precision loss, you should enter a prefix JRD when setting the format property value.
Horizontal Alignment	Specifies the horizontal justification of the text in the object. Choose an option from the drop-down list. Data type: Enumeration
Italic	Specifies whether to make the text italic. Data type: Boolean
Underline	Specifies whether to underline the text. Data type: Boolean
Vertical Alignment	Specifies the vertical justification of the text in the object. Choose an option from the drop-down list. Data type: Enumeration
Others	
Export to Applet	Specifies whether to include the object when exporting the report to Applet. Data type: Boolean
Export to CSV	Specifies whether to include the object when exporting the report to CSV. Data type: Boolean
Export to Excel	Specifies whether to include the object when exporting the report to Excel. Data type: Boolean
Export to HTML	Specifies whether to include the object when exporting the report to HTML. Data type: Boolean
Export to PDF	Specifies whether to include the object when exporting the report to PDF. Data type: Boolean
Export to PostScript	Specifies whether to include the object when exporting the report to PostScript. Data type: Boolean
Export to Report Result	Specifies whether to include the object when exporting the report to Report Result. Data type: Boolean
Export to RTF	Specifies whether to include the object when exporting the report to RTF. Data type: Boolean
Export to Text	Specifies whether to include the object when exporting the report to Text. Data type: Boolean
Export to XML	Specifies whether to include the object when exporting the report to XML. Data type: Boolean
Invisible	Specifies whether to show the object in the design area and in the report results. All formulas and calculations will still be performed if the property is set to true. Data type: Boolean
Parameter	Specifies the value of the parameter. Data type: String

Parameter form control

The properties of a parameter form control are:

Property Name	Description
Geometry	
Height	Specifies the height of the object, in inches or centimeters. Enter a numeric value to change the height. Data type: Float
Width	Specifies the width of the object, in inches or centimeters. Enter a numeric value to change the width. Data type: Float
Color	
Background	Specifies the background color of the object. Choose a color from the drop-down list, enter a hexadecimal RGB value (for example, 0xff0000), or select Custom from the drop-down list to specify the color. Data type: String
CSS	
Class	Specifies a CSS class to be applied to the object which is a valid class in the CSS file. Data type: String
ID	Specifies the identifier of the object, which must be unique in the report. The ID property can be a style sheet selector. Data type: String
Style	The property can be used in two ways. • Specifies a style to be applied to the object. Choose a style from the drop-down list which is available when a Style Group has been selected at the report tab level and when there are styles in the Style Group that can be applied to the object, or type in the style name. • Specifies a CSS selector to be applied to the object. Enter a valid class name from the CSS file. Data type: String
Others	
Export to Applet	Specifies whether to include the object when exporting the report to Applet. Data type: Boolean
Export to CSV	Specifies whether to include the object when exporting the report to CSV. Data type: Boolean
Export to Excel	Specifies whether to include the object when exporting the report to Excel. Data type: Boolean
Export to HTML	Specifies whether to include the object when exporting the report to HTML. Data type: Boolean
Export to PDF	Specifies whether to include the object when exporting the report to PDF. Data type: Boolean
Export to PostScript	Specifies whether to include the object when exporting the report to PostScript. Data type: Boolean
Export to Report Result	Specifies whether to include the object when exporting the report to Report Result. Data type: Boolean
Export to RTF	Specifies whether to include the object when exporting the report to RTF. Data type: Boolean
Export to Text	Specifies whether to include the object when exporting the report to Text. Data type: Boolean
Export to XML	Specifies whether to include the object when exporting the report to XML. Data type: Boolean
Invisible	Specifies whether to show the object in the design area and in the report results. All formulas and calculations will still be performed if the property is set to true. Data type: Boolean
Position	Specifies the position of the object. Choose an option from the drop-down list. • **static** - The object is positioned at the location in which it is inserted. If selected, the X, Y and other position-related properties will be hidden or disabled. The position of the object is affected by preceding objects in the same container. • **absolute** - The object will be located at the position specified by dragging and dropping or by setting its X and Y property values. The position of the object is not affected by preceding objects in the same container. Data type: Enumeration
Title	
Background	Specifies the background color of the title bar. Choose a color from the drop-down list, enter a hexadecimal RGB value (for example, 0xff0000), or select Custom from the drop-down list to specify the color. Data type: String
Bold	Specifies whether to make the text bold in the title bar. Data type: Boolean
Font Face	Specifies the font of the text in the title bar. Choose an option from the drop-down list. Data type: Enumeration
Font Size	Specifies the font size of the text in the title bar. Enter an integer value to change the size. Data type: Integer
Foreground	Specifies the foreground color of the title bar. Choose a color from the drop-down list, enter a hexadecimal RGB value (for example, 0xff0000), or select Custom from the drop-down list to specify the color. Data type: String
Horizontal Alignment	Specifies the horizontal justification of the text in the title bar. Choose an option from the drop-down list. Data type: Enumeration
Italic	Specifies whether to make the text italic in the title bar. Data type: Boolean
Show Title	Specifies whether to show the title bar. It is meaningful to set all the other properties in the Title category when this property is set to true. Data type: Boolean
Text	Specifies the text displayed in the title bar. Enter a string to change the text. Data type: String
Underline	Specifies whether to underline the text in the title bar. Data type: Boolean
Excel	
Column Index	Specifies the X coordinate of the object relative to its parent container when exported to Excel or CSV, measured in cells. The Columned property at the report level must be set to true for this property to take effect. Data type: Integer
Row Index	Specifies the Y coordinate of the object relative to its parent container when exported to Excel or CSV, measured in cells. The Columned property at the report level must be set to true for this property to take effect. Data type: Integer
Border	
Border Color	Specifies the color of the border of the object. Choose a color from the drop-down list, enter a hexadecimal RGB value (for example, 0xCCAA00), or select Custom from the drop-down list to specify the color. Data type: String
Border Thickness	Specifies the width of the border in inches or centimeters. Enter a numeric value to change the thickness. Data type: Float
Bottom Line	Specifies the line style of the bottom border of the object. Choose a style from the drop-down list. Data type: Enumeration
Left Line	Specifies the line style of the left border of the object. Choose a style from the drop-down list. Data type: Enumeration
Right Line	Specifies the line style of the right border of the object. Choose a style from the drop-down list. Data type: Enumeration
Top Line	Specifies the line style of the top border of the object. Choose a style from the drop-down list. Data type: Enumeration
Pattern	
Pattern Color	Specifies the color in which to draw a pattern to fill the object. Choose a color from the drop-down list, enter a hexadecimal RGB value (for example, 0xff0000), or select Custom from the drop-down list to specify the color and fill effect. Data type: String
Pattern Style	Specifies the style of the pattern. Choose an option from the drop-down list. • **none** - No pattern will be applied to the object. • **50**% - The object will be filled in the Pattern Color of 50%-transparency. • **horizontal** - The pattern will be made of horizontal lines of the Pattern Color. • **vertical** - The pattern will be made of vertical lines of the Pattern Color. • **grid** - The pattern will be an overlapping of horizontal and vertical lines of the Pattern Color. • **diagonal** - The pattern will be made of diagonal lines of the Pattern Color. Data type: Enumeration

Web report

The properties of a web report are:

Property Name	Description
General	
Default Format for Viewing Report	Specifies the default format with which to view the report on JReport Server. The default value is to adopt the JReport Server setting. If you select a specific format, you can click the button beside to further configure the format settings. After this property is configured, it will only control the directly-running results of the report on JReport Server, but has no effect on the results of advanced running and scheduled running. • **<Server Setting>** Adopts the format setting on JReport Server. • **HTML** Specifies to display the report in HTML format. Click in the value cell to configure the format settings in the HTML Option dialog. • **PDF** Specifies to display the report in PDF format. Click in the value cell to configure the format settings in the PDF Option dialog. • **Excel** Specifies to display the report in Excel format. Click in the value cell to configure the format settings in the Excel Option dialog. • **Text** Specifies to display the report in Text format. Click in the value cell to configure the format settings in the Text Option dialog. • **RTF** Specifies to display the report in RTF format. Click in the value cell to configure the format settings in the RTF Option dialog. • **XML** Specifies to display the report in XML format. Click in the value cell to configure the format settings in the XML Option dialog. • **PostScript** Specifies to display the report in PostScript format. Click in the value cell to configure the format settings in the PostScript Option dialog. • **Applet** Specifies to display the report in Applet format. Click in the value cell to configure the format settings in the Applet Option dialog. Web reports cannot run in Applet. • **Web Report** Specifies to display the report in Web Report Studio. Click in the value cell to configure the format settings in the Web Report Option dialog. Data type: Enumeration
Result Buffer	
Result Buffer Size	Specifies the number of report results pages which will be stored in the buffer. Data type: Integer
Others	
Click Priority	Specifies the priority of the actions that will be triggered at runtime when you click on certain objects in the report which are bound with some actions. Click in the value cell to set the priority in the Click Priority dialog. Data type: String
Embedded Fonts	Specifies the True Type Fonts that have been used in the report so as to embed them if you want to export the report results to a PDF file. You can select multiple fonts by pressing the Ctrl or Shift key. For more information, refer to Delivering the result in PDF. Data type: String
No Temp File	Specifies whether or not to create temporary files. • **true** - Not to create temporary files. Gains faster engine performance. • **false** - To create temporary files. Gains better accuracy in calculating data. Data type: Boolean

On Parameter Value Change	Specifies to use formulas to validate values of the parameters used by the report, then when the parameter values are changed at runtime, the values will be passed to the formulas first for validation. If the values are valid, they will be applied to the parameters; if not, you can make messages returned. Note that when you define the formula, you need to make it return a blank string for valid parameter values. Choose the view elements in the business view the library component uses which are mapped to the required formulas from the drop-down list (to select multiple view elements, make user of the Ctrl or Shift keys on the keyboard, after selecting the elements, click outside of the value cell to confirm).
	
	For example, for a string type parameter which requires a value that is of 4-7 characters, you can define a formula like this:
	
	`if(length(@P_String) > 8 ) "The value is too long."`
	`else if (length(@P_String) < 3 ) "The value is too short."`
	`else ""`
	
	Then, when a string value of 9 characters is specified to the parameter, you will be prompted with the "The value is too long." message.
	
	Data type: Object Array

Page Background	Specifies the background color of the report page. Choose a color from the drop-down list, enter a hexadecimal RGB value (for example, 0xff0000), or select Custom from the drop-down list to specify the color. Data type: String
Style Group	Specifies the style group for the report. If the report is to be used as a linked report, if the option "Pass style group information down to linked report" is check when defining the link condition, the linked report will inherit the style group of the primary report, if not checked, the linked report will take the style group specifed by the property. Data type: String
Excel	
Columned	If true, the engine performance will be improved when exporting the report to CSV or Excel format. Only after the Columned property is set to true, it is meaningful to specify values to the following Excel properties of objects in the report tab, otherwise the specified values for these Excel properties cannot take effect. • Column Index/Row Index • Column Number/Row Number. The properties are for chart platforms and images. Data type: Boolean
No Page Break	If true, there will be no page breaks on each individual sheet when exporting a report to Excel with the option Column Format selected in the Export to Excel dialog. This means the column headings will appear just at the top of the Excel spreadsheet and will not be repeated throughout the report. If the report breaks to a new sheet then once again the page header will display. Data type: Boolean **Note:** Exporting a report to an Excel file without page breaks means the split pages are technically merged within the Excel sheet, which could bring the following results and limitations: • In the exported Excel format file, only one page header panel and one page footer panel in the banded object will be displayed in each sheet. Only the first banded page header and the last banded page footer will be displayed in the exported result. As for tables, only the first table header and the last table footer will be exported. • You cannot get a continuous table or banded object if the report has other objects in horizontal locations. All split parts of the horizontal table will be exported vertically. • Merging tables or crosstabs together will extend the Excel sheet, which may cover other objects if they happen to be in the extended direction. When merging crosstabs together, the column/row index might be extended over the Excel sheet range, and the report part which is out of this range will get lost in the result file. • If a whole merged crosstab gets quite large, it might cause an out of memory condition during the exporting process.

Properties for library component

The following are object properties for library component included in the Report Inspector, which are listed in alphabetical order:

- Chart
- Configuration panel
- Contents
- Crosstab
- DBField
- Formula field
- Geographic map
- Image
- Label
- Table
- Web control
- Wrapper

Configuration panel

The properties of the configuration panel of a library component are:

Property Name	Description
General	
Class Type	Indicates the class type of the object. This property is read only.
Data Inherit	Indicates whether the dataset for this object is inherited from another object. This property is read only.
Dataset	Indicates the dataset used by the object. This property is read only.
Instance Name	Shows the instance name of the object. This property is read only.
Geometry	
Height	Specifies the height of the object, in inches or centimeters. Enter a numeric value to change the height. Data type: Float
CSS	
Class	Specifies a CSS class to be applied to the object which is a valid class in the CSS file. Data type: String
Others	
Default Show	Specifies whether to show the configuration panel by default when the library component is inserted into a dashboard. Data type: Boolean

Contents

The properties of the library component contents are:

Property Name	Description
General	
Click Priority	Specifies the priority of the actions that will be triggered at runtime when you click on certain objects in the library component which are bound with some actions. Click ... in the value cell to set the priority in the Click Priority dialog. Data type: String
Height	Specifies the normal height of the contents in inches. Enter a numeric value to change the height. Data type: Float
Horizontal Auto Fit	Specifies whether to change the width of the component in the contents automatically to fit the content width when the component is resized in JDashboard. Data type: Boolean
Minimum Height	Specifies the minimum auto fit height of the contents in inches. The property works only when Vertical Auto Fit is set to true. Enter a numeric value to change the height. Data type: Float
Minimum Width	Specifies the minimum auto fit width of the contents in inches. The property works only when Horizontal Auto Fit is set to true. Enter a numeric value to change the width. Data type: Float
Scroll Bar	Specifies whether to show the scrollbar when the minimum size of the library component is larger than its actual size at runtime in JDashboard. Data type: Boolean
Vertical Auto Fit	Specifies whether to change the height of the component in the contents automatically to fit the content height when the component is resized in JDashboard. Data type: Boolean
Width	Specifies the normal width of the contents in inches. Enter a numeric value to change the width. Data type: Float
CSS	
Class	Indicates the CSS class that will be applied to the object. This property is read only. Data type: String
Other	

On Parameter Value Change	Specifies to use formulas to validate values of the parameters used by the library component, then when the parameter values are changed at runtime, the values will be passed to the formulas first for validation. If the values are valid, they will be applied to the parameters; if not, you can make messages returned. Note that when you define the formula, you need to make it return a blank string for valid parameter values. Choose the view elements in the business view the library component uses which are mapped to the required formulas from the drop-down list (to select multiple view elements, make use of the Ctrl or Shift keys on the keyboard, after selecting the elements, click outside of the value cell to confirm).
	
	For example, for a string type parameter which requires a value that is of 4-7 characters, you can define a formula like this:
	
	`if(length(@P_String) > 8 ) "The value is too long."`
	`else if (length(@P_String) < 3 ) "The value is too short."`
	`else ""`
	
	Then, when a string value of 9 characters is specified to the parameter, you will be prompted with the "The value is too long." message.
	
	Data type: Object Array

The following topic lists properties of the child objects of the library component contents:

- Body
- Data source
- Page panel

Body

The properties of the body in the library component contents are:

Property Name	Description
Border	
Bottom Border Color	Shows the color of the bottom border line of the object. This property is read only. Data type: String
Bottom Border Thickness	Shows the thickness of the bottom border line of the object. This property is read only. Data type: Float
Bottom Line	Shows the line style of the bottom border of the object. This property is read only. Data type: Enumeration
Left Border Color	Shows the color of the left border line of the object. This property is read only. Data type: String
Left Border Thickness	Shows the thickness of the left border line of the object. This property is read only. Data type: Float
Left Line	Shows the line style of the left border of the object. This property is read only. Data type: Enumeration
Right Border Color	Shows the color of the right border line of the object. This property is read only. Data type: String
Right Border Thickness	Shows the thickness of the right border line of the object. This property is read only. Data type: Float
Right Line	Shows the line style of the right border of the object. This property is read only. Data type: Enumeration
Top Border Color	Shows the color of the top border line of the object. This property is read only. Data type: String
Top Border Thickness	Shows the thickness of the top border line of the object. This property is read only. Data type: Float
Top Line	Shows the line style of the top border of the object. This property is read only. Data type: Enumeration
Color	
Background	Specifies the background color of the object. Choose a color from the drop-down list, enter a hexadecimal RGB value (for example, 0xff0000), or select Custom from the drop-down list to specify the color. Data type: String
CSS	
Class	Specifies a CSS class to be applied to the object which is a valid class in the CSS file. Data type: String
Padding	
Bottom Padding	Specifies the space between the text and the bottom border of the object. Enter a numeric value to change the padding. Data type: Float
Left Padding	Specifies the space between the text and the left border of the object. Enter a numeric value to change the padding. Data type: Float
Right Padding	Specifies the space between the text and the right border of the object. Enter a numeric value to change the padding. Data type: Float
Top Padding	Specifies the space between the text and the top border of the object. Enter a numeric value to change the padding. Data type: Float
Margin	
Bottom Margin	Specifies the distance in inches or centimeters between the body and the bottom border of the content. Enter a numeric value to change the margin. Data type: Float
Left Margin	Specifies the distance in inches or centimeters between the body and the left border of the content. Enter a numeric value to change the margin. Data type: Float
Right Margin	Specifies the distance in inches or centimeters between the body and the right border of the content. Enter a numeric value to change the margin. Data type: Float
Top Margin	Specifies the distance in inches or centimeters between the body and the top border of the content. Enter a numeric value to change the margin. Data type: Float

Data source

The properties of the data source in a library component are:

Property Name	Description
Type	Indicates the type of the data source used by the library component. This property is read only.

The data source has two child objects, dataset and refresh object, the properties of which will be shown in the following topic:

- Dataset
- Refresh object

Dataset

The properties of a dataset in a library component are:

Property Name	Description
General	
Class Type	Indicates the class type of the object. This property is read only.
Data Name	Shows the name of the query or business view in use. It can be null. When it is null, the available resources for the dataset are parameters and constant level formulas in the data source.
Data Source Name	Specifies the name of the data source. This property is read only.
Is Query	It is true when the dataset is based on a query or false when based on a business view.
Others	
Data Driver	Specifies the cached query result file as the data resource. Usually used for support purpose. The input format for cached query result file is as follows: `jrquery:/jet.universe.resultfile.UResultFileResultSet;Fullpath_of_cached_query_result` Data type: String
Maximum Records	Specifies the maximum number of records you want to display for all of the data containers using this dataset. The default is to display all the records. Data type: Integer
Security	
Security Policy Name	Specifies the name of the data source security imported from the catalog. Choose a security policy from the value drop-down list. Data type: String **Note:** Column-level security (CLS) is not supported on library component at present, so for a library component, only the data source security policies which have not been defined with CLS will be available in the value list.
Data Buffer	
Maximum Page Number	Specifies the maximum number of pages in the data buffer. Data type: Integer
Records per Page	Specifies the number of records in each page in the data buffer. Data type: Integer

Notes:

- The properties should be set according to different databases, namely, different databases may have different configurations of data buffer.
- If the library component runs in a different environment, the properties should be specified again.
- As tested, if enough physical memory is available, and you know the total number of records in the dataset, then you can set Records per Page approximate to the total number of records to achieve better performance. Maximum Page Number is related to the library component, and should not be very large.
- The size of data buffer should be less than the available physical memory.
- The Heap Size should be set when the data buffer is quite large.

Refresh object

The properties of a refresh object in a library component are:

Property Name	Description
General	
Business View Name	Displays the name of the business view for which the refresh object is created. This property is read only. Data type: String
Data Source Name	Displays the name of the data source in which the business view is created. This property is read only. Data type: String
Name	Displays the name of the refresh object, which by default is the business view name. This property is read only. Data type: String
Query Name	Displays the name of the query on which the business view is created. Data type: String
Refresh	
Enable Auto Refresh	Specifies whether to make the data components that are created on the business view automatically refresh themselves at runtime based on a defined interval. Data type: Boolean
Incremental Fetch	Specifies whether to fetch incremental data only when an auto refresh takes place. Enabled when Enable Auto Refresh is set to true. If set to true, you can click ... in the value cell to open the Incremental Condition dialog to specify conditions to filter the incremental data. This property does not work on crosstabs and real time charts. Data type: Boolean
Interval	Specifies the time interval between two auto refreshes. It ranges from 0 to 86400 seconds. Enabled when Enable Auto Refresh is set to true. Enter a numeric value to change the interval. Data type: Integer
Unique Key	Specifies the unique key used to delete duplicated data from the incremental data filtered by incremental condition. Click ... in the value cell to open the Unique Key dialog to specify the key. It is meaningful to specify the property when Incremental Fetch is set to true. Data type: Array
Value Number	Specifies the number of records that can be displayed in the data components. The earliest records will be removed when records exceed the specified number. Enter a numeric value to change the number. It is meaningful to specify the property when Incremental Fetch is set to true. Data type: Integer

Page panel

The properties of a page panel in a library component are the same as those in a web report, and the only difference is that a page panel in a library component has no child object.

Crosstab

The properties of a crosstab in a library component are:

Property Name	Description
Color	
Background	Specifies the background color of the object. Choose a color from the drop-down list, enter a hexadecimal RGB value (for example, 0xff0000), or select Custom from the drop-down list to specify the color. Data type: String
Alternating Line Color	
Direction	Specifies whether the color defined by the property Pattern List will apply to the fields in the same rows or columns in the crosstab. • **vertical** - The value of the property Pattern List will take effect on the fields in the same columns, except for the total columns. • **horizontal** - The value of the property Pattern List will take effect on the fields in the same rows, except for the total rows. Data type: Enumeration
Enable	Specifies whether to enable setting alternating line color for the crosstab. Only when it is set to true, the settings for the properties Direction and Pattern List can take effect. Data type: Boolean **Note:** The setting of the alternating line color function for crosstab can only take effect when all the aggregate fields in the crosstab have transparent background.
Pattern List	Specifies the color pattern for the fields in the same rows or columns in the crosstab. Click ... in the value cell and click the small squares in the color tray one by one to specify the patterns. The patterns can be one or more of the following: Automatic, Color, Texture, and Gradient. Data type: String
CSS	
Class	Specifies a CSS class to be applied to the object which is a valid class in the CSS file. Data type: String
Style	The property can be used in two ways. • Specifies a style to be applied to the object. Choose a style from the drop-down list which is available when a Style Group has been selected at the report tab level and when there are styles in the Style Group that can be applied to the object, or type in the style name. • Specifies a CSS selector to be applied to the object. Enter a valid class name from the CSS file. Data type: String
Others	
Export to Applet	Specifies whether to include the object when exporting the report to Applet. Data type: Boolean
Export to CSV	Specifies whether to include the object when exporting the report to CSV. Data type: Boolean
Export to Excel	Specifies whether to include the object when exporting the report to Excel. Data type: Boolean
Export to HTML	Specifies whether to include the object when exporting the report to HTML. Data type: Boolean
Export to PDF	Specifies whether to include the object when exporting the report to PDF. Data type: Boolean
Export to PostScript	Specifies whether to include the object when exporting the report to PostScript. Data type: Boolean
Export to Report Result	Specifies whether to include the object when exporting the report to Report Result. Data type: Boolean
Export to RTF	Specifies whether to include the object when exporting the report to RTF. Data type: Boolean
Export to Text	Specifies whether to include the object when exporting the report to Text. Data type: Boolean
Export to XML	Specifies whether to include the object when exporting the report to XML. Data type: Boolean
Export to CSV	Specifies whether to include the object when exporting the report to CSV. Data type: Boolean
Export to Excel	Specifies whether to include the object when exporting the report to Excel. Data type: Boolean
Invisible	Specifies whether to show the object in the design area and in the report results. All formulas and calculations will still be performed if the property is set to true. Data type: Boolean
Navigation Height	Specifies the height of the frame for holding the crosstab. Enter a numeric value to specify the height. If no value is given here, JReport will calculate the height at runtime. Data type: Float
Navigation Width	Specifies the width of the frame for holding the crosstab. Enter a numeric value to specify the width. If no value is given here, JReport will calculate the width at runtime. Data type: Float
Record Location	Specifies the calculation point for the properties which use formulas. Choose an option from the drop-down list. • **Default** - The values of the properties are calculated in the default location where they are located. • **Page Header** - The values of the properties are calculated in the page header. • **Page Footer** - The values of the properties are calculated in the page footer. Data type: Enumeration
Suppress	Specifies whether to show the object in the design area and in the report results. All formulas and calculations will be skipped if the property is set to true. Data type: Boolean **Note:** When both the Invisible and Suppress properties of an object are set to true, Suppress has the higher priority.
Suppress When No Records	Specifies whether to display the object in the report results when no record is returned to its parent data container. Data type: Boolean
Border	
Border Color	Specifies the color of the border of the object. Choose a color from the drop-down list, enter a hexadecimal RGB value (0xCCAA00), \or select Custom from the drop-down list to specify the color. Data type: String
Has Border	Specifies whether to show the cell borders. Data type: Boolean
Crosstab Property	
Boundary Value	Specifies the number of columns in one aggregate cell when the crosstab is displayed horizontally, or rows when displayed vertically. Data type: Integer
Column Totals on Left	Specifies whether to display the total columns on the left of the aggregates. Data type: Boolean
Horizontal Gap	Specifies the space between the left/right edge of a crosstab cell and the contents in it. Data type: Float
Row Totals on Top	Specifies whether to display the total rows on the top of the aggregates. Data type: Boolean
Suppress Column Header	Specifies whether to suppress the column header in view mode. Data type: Boolean
Suppress Row Header	Specifies whether to suppress the row header in view mode. Data type: Boolean
Use Table Style	Specifies whether to add headers to the Total rows and columns. Data type: Boolean
Vertical Gap	Specifies the space between the top/bottom edge of a crosstab cell and the contents in it. Data type: Float
Vertical Layout	Indicates whether the aggregate fields are displayed vertically. The property is read only.

A crosstab in a library component has some child objects: DBField, formula field and label, the properties of which are the same as those in a web report.

DBField

The properties of a DBField in a library component are:

Property Name	Description
General	
DBField Name	Shows the mapping name of the field in the catalog. This property is read only.
Geometry	
Height	Specifies the height of the object, in inches or centimeters. Enter a numeric value to change the height. Data type: Float
Width	Specifies the width of the object, in inches or centimeters. Enter a numeric value to change the width. Data type: Float
Color	
Background	Specifies the background color of the object. Choose a color from the drop-down list, enter a hexadecimal RGB value (for example, 0xff0000), or select Custom from the drop-down list to specify the color. Data type: String
Foreground	Specifies the foreground color of the object. Choose a color from the drop-down list, enter a hexadecimal RGB value (for example, 0xff0000), or select Custom from the drop-down list to specify the color. Data type: String
CSS	
Class	Specifies a CSS class to be applied to the object which is a valid class in the CSS file. Data type: String
ID	Specifies the identifier of the object, which must be unique in the report. The ID property can be a style sheet selector. Data type: String
Style	The property can be used in two ways. • Specifies a style to be applied to the object. Choose a style from the drop-down list which is available when a Style Group has been selected at the report tab level and when there are styles in the Style Group that can be applied to the object, or type in the style name. • Specifies a CSS selector to be applied to the object. Enter a valid class name from the CSS file. Data type: String
Padding	
Bottom Padding	Specifies the space in inches or centimeters between the text and the bottom border of the object. Enter a numeric value to change the padding. Data type: Float
Left Padding	Specifies the space in inches or centimeters between the text and the left border of the object. Enter a numeric value to change the padding. Data type: Float
Right Padding	Specifies the space in inches or centimeters between the text and the right border of the object. Enter a numeric value to change the padding. Data type: Float
Top Padding	Specifies the space in inches or centimeters between the text and the top border of the object. Enter a numeric value to change the padding. Data type: Float
Border	
Border Color	Specifies the color of the border of the object. Choose a color from the drop-down list, enter a hexadecimal RGB value (for example, 0xCCAA00), or select Custom from the drop-down list to specify the color. Data type: String
Border Thickness	Specifies the width of the border in inches or centimeters. Enter a numeric value to change the thickness. Data type: Float
Bottom Line	Specifies the line style of the bottom border of the object. Choose a style from the drop-down list. Data type: Enumeration
Left Line	Specifies the line style of the left border of the object. Choose a style from the drop-down list. Data type: Enumeration
Right Line	Specifies the line style of the right border of the object. Choose a style from the drop-down list. Data type: Enumeration
Shadow	Specifies whether to draw two shadowed borders, beneath and to the right of the object. Data type: Boolean
Shadow Color	Specifies the color of the shadow. Choose a color from the drop-down list, enter a hexadecimal RGB value (for example, 0xff0000), or select Custom from the drop-down list to specify the color and fill effect. Data type: String
Top Line	Specifies the line style of the top border of the object. Choose a style from the drop-down list. Data type: Enumeration
Pattern	
Pattern Color	Specifies the color in which to draw a pattern to fill the object. Choose a color from the drop-down list, enter a hexadecimal RGB value (for example, 0xff0000), or select Custom from the drop-down list to specify the color and fill effect. Data type: String
Pattern Style	Specifies the style of the pattern. Choose an option from the drop-down list. • **none** - No pattern will be applied to the object. • **50%** - The object will be filled in the Pattern Color of 50%-transparency. • **horizontal** - The pattern will be made of horizontal lines of the Pattern Color. • **vertical** - The pattern will be made of vertical lines of the Pattern Color. • **grid** - The pattern will be an overlapping of horizontal and vertical lines of the Pattern Color. • **diagonal** - The pattern will be made of diagonal lines of the Pattern Color. Data type: Enumeration
Text Format	
Auto Fit	Specifies whether to adjust the width and height of the object according to the contents. Data type: Boolean
Bold	Specifies whether to make the text bold. Data type: Boolean
Font Face	Specifies the font of the text. Choose an option from the drop-down list. Data type: Enumeration
Font Size	Specifies the font size of the text. Enter an integer value to change the size. Data type: Integer
Format	Specifies the display format of the text in the report result. It varies with data type and can be manually defined. Data type: String **Note:** For the BigDecimal type, to avoid precision loss, you should enter a prefix JRD when setting the format property value.
Horizontal Alignment	Specifies the horizontal justification of the text in the object. Choose an option from the drop-down list. Data type: Enumeration
Italic	Specifies whether to make the text italic. Data type: Boolean
Maximum Width	Specifies the maximum width of the text displayed. Enter a numeric value to change the width. This property often works with the Auto Fit property. If Auto Fit = true and Maximum Width is not equal to 0, the text will extend until the width is this value. Data type: Float
Strikethrough	Specifies whether to add a strikethrough line to the text. Data type: Boolean
Underline	Specifies whether to underline the text. Data type: Boolean
Vertical Alignment	Specifies the vertical justification of the text in the object. Choose an option from the drop-down list. Data type: Enumeration
Word Wrap	Specifies whether to wrap the text to the width. Data type: Boolean
Others	
Data Mapping File	Specifies the data mapping file to the object for NLS use. For example, if the data mapping file name is Category_de_DE.properties, set the value as Category (without the language and locale part). Data type: String
Enable Hyperlink in Excel	Specifies whether to enable the link defined on the object when exporting the report to an Excel file. Data type: Boolean
Enable Hyperlink in HTML	Specifies whether to enable the link defined on the object when exporting the report to an HTML file. Data type: Boolean
Enable Hyperlink in PDF	Specifies whether to enable the link defined on the object when exporting the report to a PDF file. Data type: Boolean
Export to CSV	Specifies whether to include the object when exporting the report to CSV. Data type: Boolean
Export to Excel	Specifies whether to include the object when exporting the report to Excel. Data type: Boolean
Export to Report Result	Specifies whether to show the object in the design area and in the report results. All formulas and calculations will still be performed. Data type: Boolean
Invisible	Specifies whether to show the object in the design area and in the report results. All formulas and calculations will still be performed if the property is set to true. Data type: Boolean
Link	Specifies to link the object to another report, a website, an e-mail address or a Blob data type field. Click ... in the value cell to set the link target. For details, see Binding a link to a label. Data type: String
Logic Column	Specifies whether to show the object in the next visible table cell in the same row when the column which holds the object is hidden. Choose an option from the drop-down list. Data type: Enumeration **Notes:** • The next visible table cell should be completely empty, that is, the cell should not hold any content including blank space. • When the property for several objects in the same row is set to next visible column, and the columns holding these objects are all hidden, then only the object in the rightmost column will be shown in the next visible cell.
Position	Specifies the position of the object. Choose an option from the drop-down list. • **static** - The object is positioned at the location in which it is inserted. If selected, the X, Y and other position-related properties will be hidden or disabled. The position of the object is affected by preceding objects in the same container. • **absolute** - The object will be located at the position specified by dragging and dropping or by setting its X and Y property values. The position of the object is not affected by preceding objects in the same container. Data type: Enumeration
Record Location	Specifies the calculation point for the properties which use formulas. Choose an option from the drop-down list. • **Default** - The values of the properties are calculated in the default location where they are located. • **Page Header** - The values of the properties are calculated in the page header. • **Page Footer** - The values of the properties are calculated in the page footer. Data type: Enumeration
Transfer Style	Specifies whether the style group of the primary report will be applied to the linked report if the object is linked to a report. Data type: Boolean
Value Delimiter	Specifies the separator for the data whose type is DBArray. The default value space means that the elements will be displayed in a horizontal line and separated by a space. Data type: String

Note: For tables, only these DBField properties can be rendered in JDashboard:

- Data Mapping File
- Link
- Properties of the DBField object in the Web Report Studio Inspector panel.

Formula field

The properties of a formula field in a library component are the same as those in a web report.

Geographic map

The properties of a geographic map and its child objects in a library component are the same as those in a web report.

Image

The properties of an image in a library component are:

Property Name	Description
Geometry	
Height	Specifies the height of the object, in inches or centimeters. Enter a numeric value to change the height. Data type: Float
Width	Specifies the width of the object, in inches or centimeters. Enter a numeric value to change the width. Data type: Float
CSS	
Class	Specifies a CSS class to be applied to the object which is a valid class in the CSS file. Data type: String
ID	Specifies the identifier of the object, which must be unique in the report. The ID property can be a style sheet selector. Data type: String
Style	The property can be used in two ways. • Specifies a style to be applied to the object. Choose a style from the drop-down list which is available when a Style Group has been selected at the report tab level and when there are styles in the Style Group that can be applied to the object, or type in the style name. • Specifies a CSS selector to be applied to the object. Enter a valid class name from the CSS file. Data type: String
Others	
Export to Applet	Specifies whether to include the object when exporting the report to Applet. Data type: Boolean
Export to Excel	Specifies whether to include the object when exporting the report to Excel. Data type: Boolean
Export to HTML	Specifies whether to include the object when exporting the report to HTML. Data type: Boolean
Export to PDF	Specifies whether to include the object when exporting the report to PDF. Data type: Boolean
Export to PostScript	Specifies whether to include the object when exporting the report to PostScript. Data type: Boolean
Export to Report Result	Specifies whether to include the object when exporting the report to Report Result. Data type: Boolean
Export to RTF	Specifies whether to include the object when exporting the report to RTF. Data type: Boolean
Horizontal Alignment	Specifies the horizontal alignment of the image in the container. Choose an option from the drop-down list. Data type: Enumeration
Invisible	Specifies whether to show the object in the design area and in the report results. All formulas and calculations will still be performed if the property is set to true. Data type: Boolean
Link	Specifies to link the object to another report, a website, an e-mail address or a Blob data type field. Click ... in the value cell to set the link target. For details, see Binding a link to a label. Data type: String
Logic Column	Specifies whether to show the object in the next visible table cell in the same row when the column which holds the object is hidden. Choose an option from the drop-down list. Data type: Enumeration **Notes:** • The next visible table cell should be completely empty, that is, the cell should not hold any content including blank space. • When the property for several objects in the same row is set to next visible column, and the columns holding these objects are all hidden, then only the object in the rightmost column will be shown in the next visible cell.
Record Location	Specifies the calculation point for the properties which use formulas. Choose an option from the drop-down list. • **Default** - The values of the properties are calculated in the default location where they are located. • **Page Header** - The values of the properties are calculated in the page header. • **Page Footer** - The values of the properties are calculated in the page footer. Data type: Enumeration
Transfer Style	Specifies whether the style group of the primary report will be applied to the linked report if the object is linked to a report. Data type: Boolean
Vertical Alignment	Specifies the vertical alignment of the image in the container. Choose an option from the drop-down list. Data type: Enumeration
Image Property	
Alternate Text	Specifies the text that will be displayed instead if the image cannot be displayed. Data type: String
Disabled	Specifies whether to disable the actions on the image. Data type: Boolean
Image Name	Specifies the file name of the image. The image can be one on your local file system or retrieved by using a URL. Data type: String
Maximum Scaling Ratio	Specifies the maximum ratio up to which the image can be scaled. Data type: Float
Name	Specifies the name of the image. Data type: String
Rotation	Specifies the angle at which to rotate the image, in degrees. The following is the meaning of different values: • 0 - No rotation. • positive value - Rotates the image clockwise. • negative value - Rotates the image anticlockwise. Data type: Float
Scaling Mode	Specifies the scaling mode of the image, which controls the image behavior when you try to adjust the image field size. Choose an option from the drop-down list. • **actual size** - The image will remain its original size. If the image field size is smaller than the image size, part of the image will be cut off. • **fit image** - The image size will be automatically adjusted to fill up the image field, with its original perspective remained but under the limitation of Maximum Scaling Ratio. • **fit width** - The image will be automatically adjusted to fit the width of the image field but under the limitation of Maximum Scaling Ratio. • **fit height** - The image will be automatically adjusted to fit the height of the image field but under the limitation of Maximum Scaling Ratio. • **customize** - The image size will be automatically adjusted to fit the image field, regardless of the Maximum Scaling Ratio's value. Data type: Enumeration
Tab Index	Specifies the index that defines the tab order for the image. Data type: Integer
Title	Specifies the tip information about the image, which will be displayed when you hover the mouse pointer over the image at runtime. Data type: String
Value	Specifies the initial value of the image. Data type: String
Event	
On Blur	Specifies the action when the object loses input focus. Data type: String
On Click	Specifies the action when the user clicks the left mouse button on the object. Data type: String
On Double Click	Specifies the action when the user double-clicks the object. Data type: String
On Focus	Specifies the action when the object receives focus. Data type: String
On Key Down	Specifies the action when the user presses a key. Data type: String
On Key Press	Specifies the action when the user presses an alphanumeric key. Data type: String
On Key Up	Specifies the action when the user releases a key. Data type: String
On Mouse Down	Specifies the action when the user clicks the object with either mouse button. Data type: String
On Mouse Move	Specifies the action when the user moves the mouse over the object. Data type: String
On Mouse Out	Specifies the action when the user moves the mouse pointer outside the boundaries of the object. Data type: String
On Mouse Over	Specifies the action when the user moves the mouse pointer into the object. Data type: String
On Mouse Up	Specifies the action when the user releases a mouse button while the mouse is over the object. Data type: String

Label

The properties of a label in a library component are the same as those in a web report.

Table

The properties of a table in a library component are:

Property Name	Description
Color	
Background	Specifies the background color of the object. Choose a color from the drop-down list, enter a hexadecimal RGB value (for example, 0xff0000), or select Custom from the drop-down list to specify the color. Data type: String
CSS	
Class	Specifies a CSS class to be applied to the object which is a valid class in the CSS file. Data type: String
Style	The property can be used in two ways. • Specifies a style to be applied to the object. Choose a style from the drop-down list which is available when a Style Group has been selected at the report tab level and when there are styles in the Style Group that can be applied to the object, or type in the style name. • Specifies a CSS selector to be applied to the object. Enter a valid class name from the CSS file. Data type: String
Others	
Auto Expands	Specifies whether the objects in cells of the table will expand to their empty neighbouring cells horizontally when the actual size of the objects is bigger than the cells due to no enough column or page width for auto fit. This property requires that the property Auto Fit for the objects in the table cells is set to true. Data type: Boolean **Note:** The property does not apply to horizontal tables.
Export to Applet	Specifies whether to include the object when exporting the report to Applet. Data type: Boolean
Export to CSV	Specifies whether to include the object when exporting the report to CSV. Data type: Boolean
Export to Excel	Specifies whether to include the object when exporting the report to Excel. Data type: Boolean
Export to HTML	Specifies whether to include the object when exporting the report to HTML. Data type: Boolean
Export to PDF	Specifies whether to include the object when exporting the report to PDF. Data type: Boolean
Export to PostScript	Specifies whether to include the object when exporting the report to PostScript. Data type: Boolean
Export to Report Result	Specifies whether to include the object when exporting the report to Report Result. Data type: Boolean
Export to RTF	Specifies whether to include the object when exporting the report to RTF. Data type: Boolean
Export to Text	Specifies whether to include the object when exporting the report to Text. Data type: Boolean
Export to XML	Specifies whether to include the object when exporting the report to XML. Data type: Boolean
Invisible	Specifies whether to show the object in the design area and in the report results. All formulas and calculations will still be performed if the property is set to true. Data type: Boolean
Navigation Height	Specifies the height of the page for the table to display multiple pages. Enter a numeric value to specify the height. If no value is given here, JReport will calculate the table page height at runtime. Data type: Float
Navigation Width	Specifies the width of the page for the table to display multiple pages. Enter a numeric value to specify the width. If no value is given here, JReport will calculate the table page width at runtime. Data type: Float
Record Location	Specifies the calculation point for the properties which use formulas. Choose an option from the drop-down list. • **Default** - The values of the properties are calculated in the default location where they are located. • **Page Header** - The values of the properties are calculated in the page header. • **Page Footer** - The values of the properties are calculated in the page footer. Data type: Enumeration
Suppress	Specifies whether to show the object in the design area and in the report results. All formulas and calculations will be skipped if the property is set to true. Data type: Boolean **Note:** When both the Invisible and Suppress properties of an object are set to true, Suppress has the higher priority.
Suppress When No Records	Specifies whether to display the object in the report results when no record is returned to its parent data container. Data type: Boolean
Border	
Border Color	Specifies the color of the border of the object. Choose a color from the drop-down list, enter a hexadecimal RGB value (for example, 0xCCAA00), or select Custom from the drop-down list to specify the color. Data type: String
Border Thickness	Specifies the width of the border in inches or centimeters. Enter a numeric value to change the thickness. Data type: Float
Bottom Line	Specifies the line style of the bottom border of the object. Choose a style from the drop-down list. Data type: Enumeration
Left Line	Specifies the line style of the left border of the object. Choose a style from the drop-down list. Data type: Enumeration
Right Line	Specifies the line style of the right border of the object. Choose a style from the drop-down list. Data type: Enumeration
Top Line	Specifies the line style of the top border of the object. Choose a style from the drop-down list. Data type: Enumeration

Except for table group, other child objects (table header row, table footer row, table column, table cell, table group header row, table group footer row and table detail row) of a table in a library component are the same as those in a web report.

Table group

The properties of a table group in a library component are:

Property Name	Description
Others	
Current Block Index	The two properties, Current Block Index and Items per Block, work together to control the data of the object to be displayed in continuous page mode. Current Block Index specifies the index of the data block that will be displayed. 0 means the first block index, and 1 the second, and so on. Data type: Integer
Expand Detail Data	Specifies whether to expand the groups to show the details when the report is opened in JDashboard. Only works in continuous page mode. Data type: Boolean
Items per Block	The two properties, Current Block Index and Items per Block, work together to control the data of the object to be displayed in continuous page mode. Item per Block specifies the number of records in each data block. Data type: Integer
Group Layout	
Keep Group Together	Specifies whether to keep the whole group together. Data type: Boolean
Security	
Cascade	Specifies whether to allow the security identifiers specified by the following three properties to view the group's detail or child groups. • **true** - The detail row, the group header row, and the group footer row will be displayed. • **false** - Only the group header and footer rows will be displayed. Data type: Boolean
Grant	Enables to set cached report bursting. The property accepts user names delimited by "|" as the value. You can specify which group of data can be viewed by which user. Data type: String
Groups	Enables to set cached report bursting. The property accepts group names delimited by "|" as the value. You can specify which group of data can be viewed by which group of users. Data type: String
Roles	Enables to set cached report bursting. The property accepts role names delimited by "|" as the value. You can specify which group of data can be viewed by which role. Data type: String

Notes:

- Only these table properties can be rendered in JDashboard:
  - Invisible
  - Navigation Height
  - Navigation Width
  - Properties of the table object in the Web Report Studio Inspector panel.
- For a table header row and table footer row, only Invisible and the properties available with the table header object in Web Report Studio can be rendered in JDashboard.
- For a table cell, only the properties available with the table cell object in Web Report Studio can be rendered in JDashboard.
- All the table group, table group header row, table group footer row and table detail row properties are not supported in JDashboard.

Web control

The properties of web controls in a library component are listed as follows:

- Multivalue container
- Slider

Filter control, navigation control, parameter control and parameter form control in a library component share the same properties as those in a web report.

Multivalue container

The properties of a multivalue container object in a library component are:

Property Name	Description
Geometry	
Height	Specifies the height of the object, in inches or centimeters. Enter a numeric value to change the height. Data type: Float
Width	Specifies the width of the object, in inches or centimeters. Enter a numeric value to change the width. Data type: Float
Color	
Background	Specifies the background color of the object. Choose a color from the drop-down list, enter a hexadecimal RGB value (for example, 0xff0000), or select Custom from the drop-down list to specify the color. Data type: String
CSS	
Class	Specifies a CSS class to be applied to the object which is a valid class in the CSS file. Data type: String
ID	Specifies the identifier of the object, which must be unique in the report. The ID property can be a style sheet selector. Data type: String
Style	The property can be used in two ways. • Specifies a style to be applied to the object. Choose a style from the drop-down list which is available when a Style Group has been selected at the report tab level and when there are styles in the Style Group that can be applied to the object, or type in the style name. • Specifies a CSS selector to be applied to the object. Enter a valid class name from the CSS file. Data type: String
Others	
Export to Applet	Specifies whether to include the object when exporting the report to Applet. Data type: Boolean
Export to CSV	Specifies whether to include the object when exporting the report to CSV. Data type: Boolean
Export to Excel	Specifies whether to include the object when exporting the report to Excel. Data type: Boolean
Export to HTML	Specifies whether to include the object when exporting the report to HTML. Data type: Boolean
Export to PDF	Specifies whether to include the object when exporting the report to PDF. Data type: Boolean
Export to PostScript	Specifies whether to include the object when exporting the report to PostScript. Data type: Boolean
Export to Report Result	Specifies whether to include the object when exporting the report to Report Result. Data type: Boolean
Export to RTF	Specifies whether to include the object when exporting the report to RTF. Data type: Boolean
Export to Text	Specifies whether to include the object when exporting the report to Text. Data type: Boolean
Export to XML	Specifies whether to include the object when exporting the report to XML. Data type: Boolean
Position	Specifies the position of the object. Choose an option from the drop-down list. • **static** - The object is positioned at the location in which it is inserted. If selected, the X, Y and other position-related properties will be hidden or disabled. The position of the object is affected by preceding objects in the same container. • **absolute** - The object will be located at the position specified by dragging and dropping or by setting its X and Y property values. The position of the object is not affected by preceding objects in the same container. Data type: Enumeration
Border	
Border Color	Specifies the color of the border of the object. Choose a color from the drop-down list, enter a hexadecimal RGB value (for example, 0xCCAA00), or select Custom from the drop-down list to specify the color. Data type: String
Border Thickness	Specifies the width of the border in inches or centimeters. Enter a numeric value to change the thickness. Data type: Float
Bottom Line	Specifies the line style of the bottom border of the object. Choose a style from the drop-down list. Data type: Enumeration
Left Line	Specifies the line style of the left border of the object. Choose a style from the drop-down list. Data type: Enumeration
Right Line	Specifies the line style of the right border of the object. Choose a style from the drop-down list. Data type: Enumeration
Top Line	Specifies the line style of the top border of the object. Choose a style from the drop-down list. Data type: Enumeration
Text Format	
Bold	Specifies whether to make the text bold. Data type: Boolean
Font Face	Specifies the text font of the items in the multivalue container. Choose an option from the drop-down list. Data type: Enumeration
Font Size	Specifies the text font size of the items. Enter an integer value to change the size. Data type: Integer
Foreground	Specifies the color of the item text. Choose a color from the drop-down list, enter a hexadecimal RGB value (for example, 0xff0000) , or select Custom from the drop-down list to specify the color and fill effect. Data type: String
Italic	Specifies whether to make the text italic. Data type: Boolean

Slider

The properties of a slider in a library component are:

Property Name	Description
Slider	
Font Face	Specifies the font of the tick mark label text. Choose an option from the drop-down list. Data type: Enumeration
Font Size	Specifies the font size of the tick mark label text. Enter an integer value to change the size. Data type: Integer
Format	Specifies the display format of the text in the report result. It varies with data type and can be manually defined. Data type: String **Note:** For the BigDecimal type, to avoid precision loss, you should enter a prefix JRD when setting the format property value.
Number Slider Unit Increment	Specifies the increment between two pixels on the slider. Data type: Float
Geometry	
Height	Specifies the height of the object, in inches or centimeters. Enter a numeric value to change the height. Data type: Float
Width	Specifies the width of the object, in inches or centimeters. Enter a numeric value to change the width. Data type: Float
Color	
Background	Specifies the background color of the object. Choose a color from the drop-down list, enter a hexadecimal RGB value (for example, 0xff0000), or select Custom from the drop-down list to specify the color. Data type: String
Foreground	Specifies the foreground color of the object. Choose a color from the drop-down list, enter a hexadecimal RGB value (for example, 0xff0000), or select Custom from the drop-down list to specify the color. Data type: String
Selection	Specifies the color of the selected part of the slider bar. This property takes effect only when the type of the slider is Range of Values. Choose a color from the drop-down list, enter a hexadecimal RGB value (for example, 0xff0000), or select Custom from the drop-down list to specify the color. Data type: String
Slider Bar	Specifies the color of the slider bar except the selected part. This property takes effect only when the type of the slider is Range of Values. Choose a color from the drop-down list, enter a hexadecimal RGB value (for example, 0xff0000), or select Custom from the drop-down list to specify the color. Data type: String
CSS	
Class	Specifies a CSS class to be applied to the object which is a valid class in the CSS file. Data type: String
ID	Specifies the identifier of the object, which must be unique in the report. The ID property can be a style sheet selector. Data type: String
Style	The property can be used in two ways. • Specifies a style to be applied to the object. Choose a style from the drop-down list which is available when a Style Group has been selected at the report tab level and when there are styles in the Style Group that can be applied to the object, or type in the style name. • Specifies a CSS selector to be applied to the object. Enter a valid class name from the CSS file. Data type: String
Others	
Export to Applet	Specifies whether to include the object when exporting the report to Applet. Data type: Boolean
Export to CSV	Specifies whether to include the object when exporting the report to CSV. Data type: Boolean
Export to Excel	Specifies whether to include the object when exporting the report to Excel. Data type: Boolean
Export to HTML	Specifies whether to include the object when exporting the report to HTML. Data type: Boolean
Export to PDF	Specifies whether to include the object when exporting the report to PDF. Data type: Boolean
Export to PostScript	Specifies whether to include the object when exporting the report to PostScript. Data type: Boolean
Export to Report Result	Specifies whether to include the object when exporting the report to Report Result. Data type: Boolean
Export to RTF	Specifies whether to include the object when exporting the report to RTF. Data type: Boolean
Export to Text	Specifies whether to include the object when exporting the report to Text. Data type: Boolean
Export to XML	Specifies whether to include the object when exporting the report to XML. Data type: Boolean
Invisible	Specifies whether to show the object in the design area and in the report results. All formulas and calculations will still be performed if the property is set to true. Data type: Boolean

Wrapper

The properties of the wrapper of a library component are:

Property Name	Description
General	
Author	Specifies the author of the library component. Data type: String
Author Email	Specifies the author's e-mail address. Data type: String
Description	Specifies the description of the library component. Data type: String
Title	Specifies the title of the library component. Data type: String
CSS	
Class	Indicates the CSS class that will be applied to the title bar of the library component. This property is read only.
Refresh	
Enable Auto Refresh	Specifies whether to automatically refresh the objects in the library component which are not bound with data at runtime based on a defined interval. Data type: Boolean
Interval	Specifies the time interval between two auto refreshes. It ranges from 0 to 86400 seconds. Enter a numeric value to change the interval. It is meaningful to specify the property when Enable Auto Refresh is set to true. Data type: Integer

Properties in the Catalog Browser

The following are object properties included in the Catalog Browser, which are listed in alphabetical order:

Notes:

- The default ones are the recommended values to be used, so it is better not to randomly modify other properties except the two properties - Name and Description. Make changes only after you are sure of the meaning of each property.
- To modify the properties, you should first uncheck Forbid editing data object properties in the Catalog category of the Options dialog (to access the Options dialog, click **File** > **Options** on the Designer menu bar, or the **Options** button on the Catalog Browser toolbar).
- Make sure that the precision set in catalog is same as that in database. This is to avoid inconsistency between the database data values and the displayed report result.

- Business view
- Data source
- Data source security
- Formula
- Hierarchical data source
- Imported SQL
- JDBC/HIVE connection
- JSON connection
- MongoDB connection
- Parameter
- Query
- Query modifier
- Stored procedure
- Summary
- Table/View/Synonym
- User defined data source
- Web service connection
- XML connection

Business view

The properties of a business view are:

Property Name	Description
Description	Specifies the description of the business view, which will be shown when you hover the mouse pointer over the business view in the Resources panel of Web Report Studio or the Resource View panel of Page Report Studio. Data type: String
Display Name	Specifies the display name of the business view. Data type: String
Maximum Duration	Specifies the maximum elapsed time allowed to fetch data from the data source when the business view runs, measured in seconds. If the value is set to zero or is blank, it means the time will be unlimited. For usage about the property, see Limiting the query run time and number of records. Data type: Integer
Maximum Rows	Specifies the maximum number of rows that will be fetched from the data source when the business view runs. If the value is set to zero or is blank, it means the number will be unlimited. For usage about the property, see Limiting the query run time and number of records. Data type: Integer
Prefetch	If true, when end user runs a web report or dashboard created on the business view, JReport will fetch all used data from the business view and reuse them afterwards. If false, JReport will query data on demand. When several business view elements are based on the same columns, they will reuse the columns. The property works on JReport Server. When the property is true, the List<ColumnInfo> parameter in the implementation class of the QueryOptimizer interface will include all columns in the business view, otherwise, it will only include columns used by the web report or dashboard. For business views in MongoDB and HIVE connections, the default value is false; for those in other connections, the default value is true. Data type: Boolean

The following topics show properties of the business view elements:

- Aggregation object
- Category
- Detail object
- Group object

Aggregation object

The properties of an aggregation object are:

Property Name	Description
Aggregate Function	Specifies the aggregate function of the object. Data type: String
Customized Contents	Displays the contents of the custom aggregation. This property is read only.
Data Mapping File	Specifies the data mapping file to the object for NLS use. For example, if the data mapping file name is Category_de_DE.properties, set the value as Category (without the language and locale part). Data type: String
Description	Specifies the description of the object, which will be shown when you hover the mouse pointer over the object in the Resources panel of Web Report Studio or the Resource View panel of Page Report Studio. Data type: String
Display Name	Specifies the display name of the object. Data type: String
Is Customized	Specifies whether the aggregation object is a custom aggregation. Data type: Boolean
Mapping Name	Specifies the mapping name of the column to which the object is mapped. Data type: String
Tip	Specifies the tooltip of the object, which will be shown when you hover the mouse pointer over the object in the Resources panel of Web Report Studio or the Resource View panel of Page Report Studio. Data type: String
Type	Indicates the type of the view element. This property is read only.

Category

The properties of a category are:

Property Name	Description
Description	Specifies the description of the object, which will be shown when you hover the mouse pointer over the object in the Resources panel of Web Report Studio or the Resource View panel of Page Report Studio. Data type: String
Display Name	Specifies the display name of the object. Data type: String
Type	Indicates the type of the view element. This property is read only.

Detail object

The properties of a detail object are:

Property Name	Description
Data Mapping File	Specifies the data mapping file to the object for NLS use. • **<Auto>** - The data mapping file set for the column to which the object is mapped is applied. For example, if the data mapping file name is Category_de_DE.properties, set the value as Category (without the language and locale part). Data type: String
Description	Specifies the description of the object, which will be shown when you hover the mouse pointer over the object in the Resources panel of Web Report Studio or the Resource View panel of Page Report Studio. Data type: String
Display Name	Specifies the display name of the object. Data type: String
Mapping Name	Specifies the mapping name of the column to which the object is mapped. Data type: String
Tip	Specifies the tooltip of the object, which will be shown when you hover the mouse pointer over the object in the Resources panel of Web Report Studio or the Resource View panel of Page Report Studio. Data type: String
Type	Indicates the type of the view element. This property is read only.

Group object

The properties of a group object are:

Property Name	Description
Data Mapping File	Specifies the data mapping file to the object for NLS use. • **<Auto>** - The data mapping file set for the column to which the object is mapped is applied. For example, if the data mapping file name is Category_de_DE.properties, set the value as Category (without the language and locale part). Data type: String
Description	Specifies the description of the object, which will be shown when you hover the mouse pointer over the object in the Resources panel of Web Report Studio or the Resource View panel of Page Report Studio. Data type: String
Display Name	Specifies the display name of the object. Data type: String
Mapping Name	Specifies the mapping name of the column to which the object is mapped. Data type: String
Tip	Specifies the tooltip of the object, which will be shown when you hover the mouse pointer over the object in the Resources panel of Web Report Studio or the Resource View panel of Page Report Studio. Data type: String
Type	Indicates the type of the view element. This property is read only.

Data source

The properties of a data source are:

Property Name	Description
Display Rounding Mode	Specifies the rounding mode for displaying numeric data values in reports and dashboards that are created on this data source. Data Type: Enumeration The following are details about each rounding mode. For more information, refer to http://docs.oracle.com/javase/7/docs/api/java/math/RoundingMode.html. • **Up** Rounding mode to round away from zero. Always increments the digit prior to a non-zero discarded fraction. Example: • **Down** Rounding mode to round towards zero. Never increments the digit prior to a discarded fraction (that is, truncates). Example: • **Ceiling** Rounding mode to round towards positive infinity. If the result is positive, behaves as for Up; if negative, behaves as for Down. Example: • **Floor** Rounding mode to round towards negative infinity. If the result is positive, behaves as for Down; if negative, behaves as for Up. Example: • **Half up** Rounding mode to round towards "nearest neighbor" unless both neighbors are equidistant, in which case round up. Behaves as for Up if the discarded fraction is >= 0.5; otherwise, behaves as for Down. Example: • **Half down** Rounding mode to round towards "nearest neighbor" unless both neighbors are equidistant, in which case round down. Behaves as for Up if the discarded fraction is > 0.5; otherwise, behaves as for Down. Example: • **Half even** Rounding mode to round towards the "nearest neighbor" unless both neighbors are equidistant, in which case, round towards the even neighbor. Behaves as for Half up if the digit to the left of the discarded fraction is odd; behaves as for Half down if it's even. Example:
Is Default	Specifies whether the data source is the default data source for the current catalog. One catalog must have one and only one default data source. Data type: Boolean
Name	Specifies the name of the data source. Data type: String
Pre-join	Specifies whether or not to apply the pre-join information defined on the data source when building a query or defining join relationships in business views in the same data source. Data type: Boolean

Input Number	Input rounded to one digit with Up rounding
5.5	6
2.5	3
1.6	2
1.1	2
1.0	1
-1.0	-1
-1.1	-2
-1.6	-2
-2.5	-3
-5.5	-6

Input Number	Input rounded to one digit with Down rounding
5.5	5
2.5	2
1.6	1
1.1	1
1.0	1
-1.0	-1
-1.1	-1
-1.6	-1
-2.5	-2
-5.5	-5

Input Number	Input rounded to one digit with Ceiling rounding
5.5	6
2.5	3
1.6	2
1.1	2
1.0	1
-1.0	-1
-1.1	-1
-1.6	-1
-2.5	-2
-5.5	-5

Input Number	Input rounded to one digit with Floor rounding
5.5	5
2.5	2
1.6	1
1.1	1
1.0	1
-1.0	-1
-1.1	-2
-1.6	-2
-2.5	-3
-5.5	-6

Input Number	Input rounded to one digit with Half up rounding
5.5	6
2.5	3
1.6	2
1.1	1
1.0	1
-1.0	-1
-1.1	-1
-1.6	-2
-2.5	-3
-5.5	-6

Input Number	Input rounded to one digit with Half down rounding
5.5	5
2.5	2
1.6	2
1.1	1
1.0	1
-1.0	-1
-1.1	-1
-1.6	-2
-2.5	-2
-5.5	-5

Input Number	Input rounded to one digit with Half even rounding
5.5	6
2.5	2
1.6	2
1.1	1
1.0	1
-1.0	-1
-1.1	-1
-1.6	-2
-2.5	-2
-5.5	-6

Data source security

The properties of a data source security are:

Property Name	Description
Description	Specifies the description of the data source security. Data type: String
Name	Specifies the name of the data source security. Data type: String

Formula

The properties of a formula are:

Property Name	Description
General	
Description	Specifies the description of the formula. Data type: String
Display Width	Specifies the display width of the formula. Enter a numeric value to change the width. Data type: Float
Expression	Displays the whole statement of the formula. This property is read only.
Length	Specifies the length of the formula's returned value, in byte. Data type: Integer
Name	Specifies the name of the formula. Data type: String
Number Base	Specifies the number base of the formula. Data type: Integer
Precision	Specifies the precision of the formula's returned value. The default value comes from data source meta data and it specifies the value's largest number of digits. Data type: Integer
Scale	Specifies the number of digits to the right of the decimal point of the formula's returned value. Data type: Integer
SQL Type	Specifies the SQL type of the formula defined in Java. Choose an option from the drop-down list. Data type: Integer
Text Format	
Auto Fit	Specifies whether to adjust the width and height of the formula according to the contents. Data type: Boolean
Bold	Specifies whether to make the text bold. Data type: Boolean
Font Face	Specifies the font of the text. Choose an option from the drop-down list. Data type: Enumeration
Font Size	Specifies the font size of the text. Enter an integer value to change the size. Data type: Integer
Format	Specifies the display format of the formula's returned value. The format varies with the formula's data type and can be defined manually. Data type: String
Horizontal Alignment	Specifies the horizontal justification of the text in the formula. Choose an option from the drop-down list. Data type: Enumeration
Italic	Specifies whether to make the text italic. Data type: Boolean
Strikethrough	Specifies whether to add a strikethrough line to the text. Data type: Boolean
Underline	Specifies whether to underline the text. Data type: Boolean
Vertical Alignment	Specifies the vertical justification of the text in the formula. Choose an option from the drop-down list. Data type: Enumeration
Word Wrap	Specifies whether to wrap the text to the width. Data type: Boolean
Padding	
Bottom Padding	Specifies the space in inches or centimeters between the text and the bottom border of the formula. Enter a numeric value to change the padding. Data type: Float
Left Padding	Specifies the space in inches or centimeters between the text and the left border of the formula. Enter a numeric value to change the padding. Data type: Float
Right Padding	Specifies the space in inches or centimeters between the text and the right border of the formula. Enter a numeric value to change the padding. Data type: Float
Top Padding	Specifies the space in inches or centimeters between the text and the top border of the formula. Enter a numeric value to change the padding. Data type: Float
Geometry	
Height	Specifies the height of the formula, which takes effect when the formula is inserted into a report. Enter a numeric value to change the height. The default is 0 which means a proper height will be specified to the formula by JReport automatically. Data type: Float
Width	Specifies the width of the formula, which takes effect when the formula is inserted into a report. Enter a numeric value to change the width. The default is 0 which means a proper width will be specified to the formula by JReport automatically. Data type: Float
Color	
Background	Specifies the background color of the formula. Choose a color from the drop-down list, enter a hexadecimal RGB value (0xCCAA00), or select Custom from the drop-down list to specify the color. Data type: String
Foreground	Specifies the foreground color of the formula. Choose a color from the drop-down list, enter a hexadecimal RGB value (0xCCAA00), or select Custom from the drop-down list to specify the color. Data type: String
Border	
Border Color	Specifies the color of the border of the formula. Choose a color from the drop-down list, enter a hexadecimal RGB value (0xCCAA00), or select Custom from the drop-down list to specify the color. Data type: String
Border Thickness	Specifies the width of the border in inches or centimeters. Enter a numeric value to change the thickness. Data type: Float
Bottom Line	Specifies the line style of the bottom border of the formula. Choose a style from the drop-down list. Data type: Enumeration
Left Line	Specifies the line style of the left border of the formula. Choose a style from the drop-down list. Data type: Enumeration
Right Line	Specifies the line style of the right border of the formula. Choose a style from the drop-down list. Data type: Enumeration
Shadow	Specifies whether or not to draw two shadowed borders, beneath and to the right of the formula. Data type: Boolean
Shadow Color	Specifies the color of the border shadow. Choose a color from the drop-down list, enter a hexadecimal RGB value (0xCCAA00), or select Custom from the drop-down list to specify the color. Data type: String
Top Line	Specifies the line style of the top border of the formula. Choose a style from the drop-down list. Data type: Enumeration
Pattern	
Pattern Color	Specifies the color with which to draw a pattern to fill the formula. Choose a color from the drop-down list, enter a hexadecimal RGB value (0xCCAA00), or select Custom from the drop-down list to specify the color. Data type: String
Pattern Style	Specifies the style of the pattern. Choose an option from the drop-down list. • **none** - No pattern will be applied to the formula. • **50%** - The formula will be filled in the specified color of 50%-transparency. • **horizontal** - The pattern will be made of horizontal lines of the specified color. • **vertical** - The pattern will be made of vertical lines of the specified color. • **grid** - The pattern will be an overlapping of horizontal and vertical lines of the specified color. • **diagonal** - The pattern will be made of slash lines of the specified color. Data type: Enumeration
Others	
Data Mapping File	Specifies the data mapping file to the object for NLS use. For example, if the data mapping file name is Category_de_DE.properties, set the value as Category (without the language and locale part). Data type: String

SQL Type

Option	Type	Option	Type
-7	BIT	-6	TINYINT
-5	BIGINT	-4	LONGVARBINARY
-3	VARBINARY	-2	BINARY
-1	LONGVARCHAR	1	CHAR
2	NUMERIC	3	DECIMAL
4	INTEGER	5	SMALLINT
6	FLOAT	7	REAL
8	DOUBLE	12	VARCHAR
16	BOOLEAN	91	DATE
92	TIME	93	TIMESTAMP
2003	ARRAY	2004	BLOB
2005	CLOB		

Hierarchical data source

The properties of a hierarchical data source are:

Property Name	Description
Attributes	Specifies whether or not columns of this HDS can be used to group by. If true, they cannot. Data type: Boolean
Class Name	Displays the name of the class provided by JReport that is used to implement hierarchical data sources. This property is read only.
Description	Specifies the description of the HDS. Data type: String
Full Name	Displays the full path name of the HDS automatically generated by JReport. This property is read only.
Maximum Duration	Specifies the maximum elapsed time allowed to fetch data from the data source when the HDS runs, measured in seconds. If the value is set to zero or is blank, it means the time will be unlimited. For usage about the property, see Limiting the query run time and number of records. Data type: Integer
Maximum Rows	Specifies the maximum number of rows that will be fetched from the data source when the HDS runs. If the value is set to zero or is blank, it means the number will be unlimited. For usage about the property, see Limiting the query run time and number of records. Data type: Integer
Name	Specifies the name of the HDS. Data type: String
Parameter	Specifies the value of the HDS's parameter. Data type: String
Root Name	Specifies the name of the root node in the XML file. Data type: String
URI	Specifies the URI of the XML file. Data type: String
Version	Displays the version number of the HDS API. This property is read only.
XSD URI	Specifies the URI of the XSD file. Data type: String

An HDS contains some tables and columns in those tables. The following topics show their properties in detail:

- HDS table
- HDS column

HDS table

The properties of an HDS table are:

Property Name	Description
Attributes	Specifies whether or not the HDS columns of this HDS table can be used to group by. If true, they cannot. Data type: Boolean
Description	Specifies the description of the HDS table. Data type: String
Full Name	Displays the full path name of the HDS table automatically generated by JReport. This property is read only.
Maximum Duration	Specifies the maximum elapsed time allowed to fetch data from the data source when the HDS table runs, measured in seconds. If the value is set to zero or is blank, it means the time will be unlimited. For usage about the property, see Limiting the query run time and number of records. Data type: Integer
Maximum Rows	Specifies the maximum number of rows that will be fetched from the data source when the HDS table runs. If the value is set to zero or is blank, it means the number will be unlimited. For usage about the property, see Limiting the query run time and number of records. Data type: Integer
Name	Specifies the name of the HDS table. Data type: String

HDS column

The properties of an HDS column are:

Property Name	Description
General	
Array	Specifies whether the column is an array type or not. Data type: Boolean
Column Name	Specifies the name of the column in the raw data source. Data type: String
Currency	Specifies whether to control the SQL type of formulas or summaries in which the BigDecimal type fields are imported. • **true** - The formula or summary which is built with the BigDecimal type field will be BigDecimal type, and its SQL Type value will be set to 3. • **false** - The normal data type will be used for the formula or summary. Data type: Boolean
Description	Specifies the description of the column. Data type: String
Display Width	Specifies the display width of the column. Enter a numeric value to change the width. Data type: Float
Full Name	Displays the full path name of the column automatically generated by JReport. This property is read only.
HDS Format	Specifies the data format of the column. Data type: String
Name	Specifies the mapped name of the column in the JReport catalog. Data type: String
Nullable	Specifies the nullability of the column's value. Choose an option from the drop-down list. Data type: Enumeration
Precision	Specifies the precision of the column's value. The default value comes from data source meta data and it specifies the column's largest number of digits. Data type: Integer
Scale	Specifies the number of digits to the right of the decimal point for the column's value. Data type: Integer
SQL Type	Specifies the SQL type of the column defined in Java. Choose an option from the drop-down list. Data type: Integer
Text Format	
Auto Fit	Specifies whether to adjust the width and height of the column according to the contents. Data type: Boolean
Bold	Specifies whether to make the text bold. Data type: Boolean
Font Face	Specifies the font of the text. Choose an option from the drop-down list. Data type: Enumeration
Font Size	Specifies the font size of the text. Enter an integer value to change the size. Data type: Integer
Format	Specifies the display format of the column's value. The format varies with the column's data type and can be defined manually. Data type: String
Horizontal Alignment	Specifies the horizontal justification of the text in the column. Choose an option from the drop-down list. Data type: Enumeration
Italic	Specifies whether to make the text italic. Data type: Boolean
Strikethrough	Specifies whether to add a strikethrough line to the text. Data type: Boolean
Underline	Specifies whether to underline the text. Data type: Boolean
Vertical Alignment	Specifies the vertical justification of the text in the column. Choose an option from the drop-down list. Data type: Enumeration
Word Wrap	Specifies whether to wrap the text to the width. Data type: Boolean
Padding	
Bottom Padding	Specifies the space in inches or centimeters between the text and the bottom border of the column. Enter a numeric value to change the padding. Data type: Float
Left Padding	Specifies the space in inches or centimeters between the text and the left border of the column. Enter a numeric value to change the padding. Data type: Float
Right Padding	Specifies the space in inches or centimeters between the text and the right border of the column. Enter a numeric value to change the padding. Data type: Float
Top Padding	Specifies the space in inches or centimeters between the text and the top border of the column. Enter a numeric value to change the padding. Data type: Float
Geometry	
Height	Specifies the height of the column, which takes effect when the column is inserted into a report. By default, no value is specified to the property, which means a height will be assigned to the column by JReport automatically. Enter a numeric value to change the height. Data type: Float
Width	Specifies the width of the column, which takes effect when the column is inserted into a report. By default, no value is specified to the property, which means a height will be assigned to the column by JReport automatically. Enter a numeric value to change the width. Data type: Float
Color	
Background	Specifies the background color of the column. Choose a color from the drop-down list, enter a hexadecimal RGB value (0xCCAA00), or select Custom from the drop-down list to specify the color. Data type: String
Foreground	Specifies the foreground color of the column. Choose a color from the drop-down list, enter a hexadecimal RGB value (0xCCAA00), or select Custom from the drop-down list to specify the color. Data type: String
Border	
Border Color	Specifies the color of the border of the column. Choose a color from the drop-down list, enter a hexadecimal RGB value (0xCCAA00), or select Custom from the drop-down list to specify the color. Data type: String
Border Thickness	Specifies the width of the border in inches or centimeters. Enter a numeric value to change the thickness. Data type: Float
Bottom Line	Specifies the line style of the bottom border of the column. Choose a style from the drop-down list. Data type: Enumeration
Left Line	Specifies the line style of the left border of the column. Choose a style from the drop-down list. Data type: Enumeration
Right Line	Specifies the line style of the right border of the column. Choose a style from the drop-down list. Data type: Enumeration
Shadow	Specifies whether or not to draw two shadowed borders, beneath and to the right of the column. Data type: Boolean
Shadow Color	Specifies the color of the border shadow. Choose a color from the drop-down list, enter a hexadecimal RGB value (0xCCAA00), or select Custom from the drop-down list to specify the color. Data type: String
Top Line	Specifies the line style of the top border of the column. Choose a style from the drop-down list. Data type: Enumeration
Pattern	
Pattern Color	Specifies the color with which to draw a pattern to fill the column. Choose a color from the drop-down list, enter a hexadecimal RGB value (0xCCAA00), or select Custom from the drop-down list to specify the color. Data type: String
Pattern Style	Specifies the style of the pattern. Choose an option from the drop-down list. • **none** - No pattern will be applied to the column. • **50%** - The column will be filled in the specified color of 50%-transparency. • **horizontal** - The pattern will be made of horizontal lines of the specified color. • **vertical** - The pattern will be made of vertical lines of the specified color. • **grid** - The pattern will be an overlapping of horizontal and vertical lines of the specified color. • **diagonal** - The pattern will be made of slash lines of the specified color. Data type: Enumeration
Others	
Data Mapping File	Specifies the data mapping file to the object for NLS use. For example, if the data mapping file name is Category_de_DE.properties, set the value as Category (without the language and locale part). Data type: String

Nullable

The Nullable property decides the nullability of a column's value, and its value can be one of the following:

- **true** - Indicates the column supports null value.
- **false** - Indicates the column's value cannot be null.
- **unknown** - Indicates whether the column's value can be null or not is unknown.

In versions prior to JReport Designer Version 8, when the returned value of a column is null, of which the nullability is false, there will be a default value replacing that null value. When you upgrade JReport Designer from a previous version to Version 8, there is a change considering better use and control of null value, that is, if null value is returned whereas you specify null value is not allowed on the column, the null value would display as it is. If you want to change the null value to any other desired values, formulas can be applied. For example, with the following formula, the null value is changed to 0:

```
if (isnull (@theColumn))
return 0
else
return @theColumn;
```

Imported SQL

The properties of an imported SQL file are:

Property Name	Description
Connection Name	Specifies the connection to execute the imported SQL file. Data type: String
Description	Specifies the description of the imported SQL file. Data type: String
Maximum Duration	Specifies the maximum elapsed time allowed to fetch data from the data source when the SQL file runs, measured in seconds. If the value is set to zero or is blank, it means the time will be unlimited. For usage about the property, see Limiting the query run time and number of records. Data type: Integer
Maximum Rows	Specifies the maximum number of rows that will be fetched from the data source when the imported SQL file runs. If the value is set to zero or is blank, it means the number will be unlimited. For usage about the property, see Limiting the query run time and number of records. Data type: Integer
Name	Specifies the name of the imported SQL file. Data type: String
Qualifier	Specifies the database's catalog name of the imported SQL file. Data type: String
Read Only	Specifies whether or not the imported SQL file will be read only. Data type: Enumeration
Transaction Mode	Specifies the transaction mode for the imported SQL file. Choose an option from the drop-down list. • **default** - Indicates the transaction information cannot be got from JDBC connection. • **none** - Indicates that transactions are not supported. • **read uncommitted** - Dirty reads, non-repeatable reads and phantom reads can occur. This mode will speed up the transaction of the catalog. • **read committed** - Dirty reads are prevented; non-repeatable reads and phantom reads can occur. • **repeatable read** - Dirty reads and non-repeatable reads are prevented; phantom reads can occur. • **serializable** - Dirty reads, non-repeatable reads and phantom reads are prevented. Data type: Enumeration

An imported SQL file contains several columns, the properties of which are shown in the following topic:

- Imported SQL column

Imported SQL column

The properties of an imported SQL column are:

Property Name	Description
General	
Array	Specifies whether the column is an array type or not. Data type: Boolean
Column Index	Specifies the column index in the result set. Data type: Integer
Column Name	Specifies the name of the column in the raw data source. Data type: String
Currency	Specifies whether to control the SQL type of formulas or summaries in which the BigDecimal type fields are imported. • **true** - The formula or summary which is built with the BigDecimal type field will be BigDecimal type, and its SQL Type value will be set to 3. • **false** - The normal data type will be used for the formula or summary. Data type: Boolean
Description	Specifies the description of the column. Data type: String
Display Width	Specifies the display width of the column. Enter a numeric value to change the width. Data type: Float
Name	Specifies the mapped name of the column in the JReport catalog. Data type: String
Nullable	Specifies the nullability of the column's value. Choose an option from the drop-down list. Data type: Enumeration
Precision	Specifies the precision of the column's value. The default value comes from data source meta data and it specifies the column's largest number of digits. Data type: Integer
Scale	Specifies the number of digits to the right of the decimal point for the column's value. Data type: Integer
SQL Type	Specifies the SQL type of the column defined in Java. Choose an option from the drop-down list. Data type: Integer
Text Format	
Auto Fit	Specifies whether to adjust the width and height of the column according to the contents. Data type: Boolean
Bold	Specifies whether to make the text bold. Data type: Boolean
Font Face	Specifies the font of the text. Choose an option from the drop-down list. Data type: Enumeration
Font Size	Specifies the font size of the text. Enter an integer value to change the size. Data type: Integer
Format	Specifies the display format of the column's value. The format varies with the column's data type and can be defined manually. Data type: String
Horizontal Alignment	Specifies the horizontal justification of the text in the column. Choose an option from the drop-down list. Data type: Enumeration
Italic	Specifies whether to make the text italic. Data type: Boolean
Strikethrough	Specifies whether to add a strikethrough line to the text. Data type: Boolean
Underline	Specifies whether to underline the text. Data type: Boolean
Vertical Alignment	Specifies the vertical justification of the text in the column. Choose an option from the drop-down list. Data type: Enumeration
Word Wrap	Specifies whether to wrap the text to the width. Data type: Boolean
Padding	
Bottom Padding	Specifies the space in inches or centimeters between the text and the bottom border of the column. Enter a numeric value to change the padding. Data type: Float
Left Padding	Specifies the space in inches or centimeters between the text and the left border of the column. Enter a numeric value to change the padding. Data type: Float
Right Padding	Specifies the space in inches or centimeters between the text and the right border of the column. Enter a numeric value to change the padding. Data type: Float
Top Padding	Specifies the space in inches or centimeters between the text and the top border of the column. Enter a numeric value to change the padding. Data type: Float
Geometry	
Height	Specifies the height of the column, which takes effect when the column is inserted into a report. By default, no value is specified to the property, which means a height will be assigned to the column by JReport automatically. Enter a numeric value to change the height. Data type: Float
Width	Specifies the width of the column, which takes effect when the column is inserted into a report. By default, no value is specified to the property, which means a height will be assigned to the column by JReport automatically. Enter a numeric value to change the width. Data type: Float
Color	
Background	Specifies the background color of the column. Choose a color from the drop-down list, enter a hexadecimal RGB value (0xCCAA00), or select Custom from the drop-down list to specify the color. Data type: String
Foreground	Specifies the foreground color of the column. Choose a color from the drop-down list, enter a hexadecimal RGB value (0xCCAA00), or select Custom from the drop-down list to specify the color. Data type: String
Border	
Border Color	Specifies the color of the border of the column. Choose a color from the drop-down list, enter a hexadecimal RGB value (0xCCAA00), or select Custom from the drop-down list to specify the color. Data type: String
Border Thickness	Specifies the width of the border in inches or centimeters. Enter a numeric value to change the thickness. Data type: Float
Bottom Line	Specifies the line style of the bottom border of the column. Choose a style from the drop-down list. Data type: Enumeration
Left Line	Specifies the line style of the left border of the column. Choose a style from the drop-down list. Data type: Enumeration
Right Line	Specifies the line style of the right border of the column. Choose a style from the drop-down list. Data type: Enumeration
Shadow	Specifies whether or not to draw two shadowed borders, beneath and to the right of the column. Data type: Boolean
Shadow Color	Specifies the color of the border shadow. Choose a color from the drop-down list, enter a hexadecimal RGB value (0xCCAA00), or select Custom from the drop-down list to specify the color. Data type: String
Top Line	Specifies the line style of the top border of the column. Choose a style from the drop-down list. Data type: Enumeration
Pattern	
Pattern Color	Specifies the color with which to draw a pattern to fill the column. Choose a color from the drop-down list, enter a hexadecimal RGB value (0xCCAA00), or select Custom from the drop-down list to specify the color. Data type: String
Pattern Style	Specifies the style of the pattern. Choose an option from the drop-down list. • **none** - No pattern will be applied to the column. • **50%** - The column will be filled in the specified color of 50%-transparency. • **horizontal** - The pattern will be made of horizontal lines of the specified color. • **vertical** - The pattern will be made of vertical lines of the specified color. • **grid** - The pattern will be an overlapping of horizontal and vertical lines of the specified color. • **diagonal** - The pattern will be made of slash lines of the specified color. Data type: Enumeration
Others	
Data Mapping File	Specifies the data mapping file to the object for NLS use. For example, if the data mapping file name is Category_de_DE.properties, set the value as Category (without the language and locale part). Data type: String

JDBC/HIVE connection

The properties of a JDBC/HIVE connection are:

Property Name	Description
Custom Query Optimizer	Specifies the implementation class of the QueryOptimizer interface for optimizing the query SQL statement before being sent to the DB. The value can be "Package_Name. Class_Name". If the class file has no package name, set the value to "Class_Name". Once Custom Query Optimizer is specified, • Report data cache and in-memory cubes cannot be created or scheduled any more. • The group level pushing down will not be supported. Data type: String
Date Format	Specifies the default Date format corresponding to the database. Data type: String
Default	Specifies whether the connection is the default connection in the current data source. By default, the first added connection in a data source is the default connection of the data source. A data source can have zero or one default connection. Data type: Boolean
Description	Specifies the description of the JDBC connection. Data type: String
Driver	Specifies the class name of the JDBC driver such as oracle.jdbc.driver.OracleDriver. While setting up the JDBC connection with the connection dialog, JReport will use the driver that is specified to connect to the database. If no driver name is filled in, JReport will use the default JDBC driver from the file jdbcdrivers.properties in `<install_root>\bin` file. You can add JDBC driver names into the text file, JReport will load the drivers before building a connection. If your JDBC driver name is not correct, or you do not add JDBC driver names, the message "No suitable driver" will appear. Data type: String
Explicit Inner Join	Specifies whether to use Explicit Join notation or Implicit Join notation in the Where clause for inner joins. For HIVE connection, it can only be true. • **true** - Uses Explicit Join notation: SQL: select ... from A inner join B on (A.c1 = B.c2) • **false** - Uses Implicit Join notation: SQL: select ... from A,B where A.c1 = B.c2 Data type: Boolean
Included Schemas	Displays the schema names specified for the connection in the Get JDBC Connection Information dialog. The default is blank, which indicates that all schemas in the DBMS can be used in the catalog. This property is read only. To modify it, use the Get JDBC Connection Information dialog.
Name	Specifies the name of the connection which, by default, is the same as the connection URL, but can also be a user friendly name for the connection. If the name has already been used, a number starting from 1 will be appended to the name. For example, if aa exists, the new name will be aa1, and if both aa and aa1 exits, it will be aa2, and so on. If you change the name, the imported SQL and stored procedures that reference the connection will be automatically updated to use the new name. Data type: String
Name Pattern	Specifies whether or not catalog or schema is used in data manipulation. Choose an option from the drop-down list. • **unqualified name** - Neither catalog nor schema is included in data manipulation. Example: SELECT t.c FROM t • **2-part names** - Uses schema in data manipulation. Example: SELECT schema.t.c FROM schema.t • **3-part names** - Uses both catalog and schema in data manipulation. Example: SELECT catalog.schema.t.c from catalog.schema.t Data type: Enumeration
Outer Join Marker	Specifies the behavior of the outer joins in the connection. Choose an option from the drop-down list. • **SQL92** - Specifies to use the SQL92 standard. • **+** - Specifies to use Oracle's "+=+" standard. Data type: Enumeration
Password	Specifies the password for connecting with the database, which is determined by the database DBA. Data type: String
Push Down Group Query	Specifies whether to push down group level summary computations in reports to the DBMS at runtime. Choose an option from the drop-down list. For HIVE connection, it can only be true. • **true** - JReport will push down the group level summary computations to the DBMS if the DBMS can do the computations, otherwise, JReport will do the computations itself. • **false** - JReport will do the group level summary computations itself. Data type: Boolean
Quote Qualifier	Specifies the characters, then a qualifier name which contains the characters that will not be quoted. Choose an option from the drop-down list. • **Default (JDBC)** - If selected, the program will get the extra name characters from JDBC. • **User Defined** - If selected, you can modify the quote character according to the database system being used. Data type: Enumeration
Read Only	Specifies the mode to open the connection to the JDBC data source. The initial setting is default which uses the mode specified by the DBMS Administrator which could be read only or read & write. Choose an option from the drop-down list. • **default** - May be read & write or read only depending on the DBMS default setting. • **read only** - Allows the driver to optimize performance for reporting which does not need to write to the DBMS. • **read & write** - Opens the DBMS with updates enabled which requires more processing to ensure concurrency control. Data type: Enumeration
Security Check	Specifies whether or not to check the JDBC connection security at runtime. Data type: Boolean
SQL Statement Creator	Specifies the parameters to implement the SQLStmtCreator interface (for details about the interface and its usage, refer to Dynamic queries). Data type: String
Time Format	Specifies the default Time format corresponding to the database. Data type: String
Timestamp Format	Specifies the default Timestamp format corresponding to the database. Data type: String
Transaction Mode	Specifies the transaction mode for the connection. Choose an option from the drop-down list. • **default** - Indicates the transaction information cannot be got from JDBC connection. • **none** - Indicates that transactions are not supported. • **read uncommitted** - Dirty reads, non-repeatable reads and phantom reads can occur. This mode will speed up the transaction of the catalog. • **read committed** - Dirty reads are prevented; non-repeatable reads and phantom reads can occur. • **repeatable read** - Dirty reads and non-repeatable reads are prevented; phantom reads can occur. • **serializable** - Dirty reads, non-repeatable reads and phantom reads are prevented. Data type: Enumeration
URL	Specifies the JDBC URL which establishes the connection to the database, for example jdbc:oracle:thin:@localhost:1521:ORCL. Data type: String
User	Specifies the user name for connecting to the database, which is determined by the database DBA. Data type: String

Timestamp Format and Date Format

Symbol	Meaning	Presentation
y	year	Number
M	month	Number
d	day in month	Number
H	hour in day (0~23)	Number
h	hour in am/pm (1~12)	Number
m	minute in hour	Number
s	second in minute	Number
S	millisecond	Number

Example (using the US Locale):

"yyyyy.MMMMM.dd hh:mm aaa" ->> 1996.July.10 12:08 PM

Note: The Date and Timestamp format JReport supports follows that of Java. Refer to the Java API Specification java.text package DateFormat interface.

Push Down Group Query

This property is to specify whether to push down group level summary computations in reports to the DBMS at runtime. The group level summary computations can be pushed down to the DBMS when the aggregate function is Count, Sum, Maximum, Minimum, or Square Sum. The aggregate functions Average, PopulationStdDev, PopulationVariance, StdDev and Variance are not pushed down, but instead computed by JReport using the results of the pushed down functions. By pushing down the group level summary computations, the DBMS' computation capability can be made use of, and thus the reports' running efficiency will be improved.

The QueryOptimizer interface

The QueryOptimizer interface is contained in the package toolkit.db.queryoptimization. See JReport Javadoc in `<install_root>\help\api` for usage of the interface.

This interface contains only one method:

```
Optimizer optimizeQuery(QueryInfo queryInfo);
```

One QueryInfo will be automatically passed into the interface, and one Optimizer object should be returned.

To implement the interface and make it work:

1. Define a Java class file that implements the QueryOptimizer interface.
2. Compile the Java file to generate the class file.
3. Append the class path to the ADDCLASSPATH variable of the file setenv.bat in `<install_root>\bin`.
4. Start JReport Designer with the modified batch file.
5. Open the catalog that contains the connection you would like to optimize.
6. Set the Custom Query Optimizer property of the connection to "Package_Name.Class_Name".

Note: The List<ColumnInfo> parameter in the implementation class of the QueryOptimizer interface depends on the Prefetch property of the related business view. When Prefetch is true, List<ColumnInfo> will include all columns in the business view, otherwise, it will only include columns used by the related report.

JSON connection

The properties of a JSON connection are:

Property Name	Description
Default	Specifies whether the connection is the default connection in the current data source. By default, the first added connection in a data source is the default connection of the data source. A data source can have zero or one default connection. Data type: Boolean
Description	Specifies the description of the connection. Data type: String
Name	Specifies the name of the connection. If the name has already been used, a number starting from 1 will be appended to the name. For example, if aa exists, the new name will be aa1, and if both aa and aa1 exits, it will be aa2, and so on. If you change the name, the imported SQL and stored procedures that reference the connection will be automatically updated to use the new name. Data type: String

- JSON schema

JSON schema

The properties of a JSON schema are:

Property Name	Description
Element Type	Displays the type of the element. This property is read only.
Format for Parsing Date/ Time	Specifies the format to parse Date/Time data. This property is available only when Mapped SQL Type is DATE, TIME or TIMESTAMP. Select a format from the drop-down list or input an acceptable format pattern of java.text.SimpleDateFormat class in the text field. If it is blank, the rules in the Data type conversion table are applied. • **ISO 8601 format** - For ISO 8601 format, the value is represented by this constant string. • **JDBC timestamp escape format** - For JDBC Timestamp format, the value is represented by this constant string. Data type: String
JSON Data Type	Displays the JSON data type of the element whose element type is attribute or simple data array. This property is read only.
Mapped SQL Type	Specifies the SQL Type mapped from JSON data type. Available for attribute and simple data array elements only. Data type: String
Name	Displays the name of the element. This property is read only.

MongoDB connection

The properties of a MongoDB connection are:

Property Name	Description
Connection Options	Specifies the options used to connect to the MongoDB server. The options are name=value pairs and the pairs are separated by "&". Data type: String
Default	Specifies whether the connection is the default connection in the current data source. By default, the first added connection in a data source is the default connection of the data source. A data source can have zero or one default connection. Data type: Boolean
Description	Specifies the description of the MongoDB connection. Data type: String
Name	Specifies the name of the MongoDB connection. If the name has already been used, a number starting from 1 will be appended to the name. For example, if aa exists, the new name will be aa1, and if both aa and aa1 exits, it will be aa2, and so on. If you change the name, the imported SQL and stored procedures that reference the connection will be automatically updated to use the new name. Data type: String
Password	Specifies the password for connecting to the MongoDB database, which is determined by the database administrator. When the catalog is saved as xml format, the password will be encrypted by an algorithm and stored in an xml catalog file, and it will be decrypted when the xml format catalog is loaded. Data type: String
Push Down Group Query	Specifies whether to push down group level summary computations in reports to the DBMS at runtime. Choose an option from the drop-down list. • true - JReport will push down the group level summary computations to the DBMS if the DBMS can do the computations, otherwise, JReport will do the computations itself. • false - JReport will do the group level summary computations itself. Data type: Boolean
Server Address	Displays a list of replica set members or MongoDBs, including IP/host name and port pairs. This property is read only. If the list contains no IP/host name and port pairs, it means connecting to the local host with the default port of MongoDB. For each pair in the list, the IP/host name cannot be null, while, the port can be null. When the port is null, it means connecting to the MongoDB by the default port.
User	Specifies the user name for connecting to the database, which is determined by the database administrator. Data type: String

- MongoDB database

MongoDB database

The properties of a MongoDB database are:

Property Name	Description
Description	Specifies the description of the database in the MongoDB connection. Data type: String
Name	Specifies the name of the database in the MongoDB connection. • If the database you specify exists in the MongoDB data source, it will connect to the new database. • If the database you specify does not exist in the MongoDB data source, no data will be got. Data type: String

- MongoDB collection

MongoDB collection

The properties of a MongoDB collection are:

Property Name	Description
Description	Specifies the description of the collection in the MongoDB database. Data type: String
Name	Specifies the name of the collection in the MongoDB database. Data type: String

- MongoDB element

MongoDB element

The properties of a MongoDB element are:

Property Name	Description
BSON Data Type	Specifies the data type of the element. Choose an option from the drop-down list. Data type: Enumeration
Data Type Name	Displays the data type name of the element. This property is read only and will automatically be changed according to the BSON Data Type property.
Name	Specifies the name of the element. Data type: String
Sub Data Type	Specifies the sub data type of the element. The property takes effect only when BSON Data Type is 5. Choose an option from the drop-down list. Data type: Enumeration
Sub Type Name	Displays the sub data type name of the element. This property is read only and will automatically be changed according to the Sub Data Type property.

Parameter

The properties of a parameter are:

Property Name	Description
General	
Bind Column	Specifies the bind column of the parameter. Data type: String
Binding	Specifies the parameter which will be bound with the current one to sort the column controlled by the current parameter. Data type: String
Default Value Number	Specifies the parameter value that will be prompted by default at runtime. If it is set to 0, the first parameter value in the Value property will be prompted; if it is set to -1, the last parameter value in the Value property will be prompted. Data type: Integer
Description	Specifies the description of the parameter. Data type: String
Display Column	Specifies the display column of the parameter. Data type: String
Display Width	Specifies the display width of the parameter. Enter a numeric value to change the width. Data type: Float
Distinct	Specifies whether or not the text of the parameter is to be distinct. Data type: Boolean
Get Value from API Only	Specifies whether the parameter values can be input via GUI or not. If true, the parameter values should only be obtained from API. If false, the parameter can be obtained from both GUI and API. Data type: Boolean
Hide Parameter When Single Value Returned	When a parameter is bound with a column, you can choose to hide the parameter at runtime when only a single value is returned for the parameter, and the returned single value will be used as default value of the parameter. Data type: Boolean
Import SQL	Specifies the statement of the imported SQL file, which is created manually. Data type: String
Maximum Value/Length	Specifies the maximum value allowed for this parameter, or the maximum length for a String type parameter. Data type: String
Minimum Value/Length	Specifies the minimum value allowed for this parameter, or the minimum length for a String type parameter. Data type: String
Name	Specifies the name of the parameter. Data type: String
Object Name	Specifies the name for the OOJDBC class that is related to the parameter. Data type: String
Object Parameter Name	Specifies the parameter name for the OOJDBC class. Data type: String
Operation	Specifies whether or not to check the minimum and maximum values that you specified for the parameter at runtime. If -1, it will not check; if 1, it will check at runtime. Data type: Integer
Prompt Text	Specifies the prompt text of the parameter, which is used for prompting users to provide the value of the parameter. Data type: String
Record Level Security	When a parameter is bound with a column, you can specify a RLS policy defined on the bind column in the data source to the parameter, then at runtime, the specified SID will see the records he is allowed to view in the parameter value drop-down list. Data type: String
Required	Specifies whether or not the parameter will be treated as a required one. Data type: Boolean
String Encoding	Specifies the encoding for the parameter. This property is available only when the parameter's data type is String. Data type: String
String Format	Specifies the user-defined format of the parameter value. This property works only when the parameter's data type is String. Data type: String
Treat Blank as Null	Specifies whether or not to treat blank data value as null when the parameter's data type is String. Data type: Boolean
Use Current Date-Time	Specifies whether or not to use the current system's date time as value of the parameter when the parameter's data type is Date, Time or DateTime. Data type: Boolean
Use Current Date-Time When Blank	Specifies whether or not to use the current system's date time as value of the parameter when the parameter's data type is Date, Time or DateTime, and the parameter value is blank. Data type: Boolean
User Defined Format	Specifies the user-defined format of the parameter value. Data type: String
Value List	Lists values of the parameter. This property is read only.
Value Type	Specifies the data type of the parameter. Choose an option from the drop-down list. Data type: Enumeration
Text Format	
Auto Fit	Specifies whether to adjust the width and height of the parameter according to the contents. Data type: Boolean
Bold	Specifies whether to make the text bold. Data type: Boolean
Font Face	Specifies the font of the text. Choose an option from the drop-down list. Data type: Enumeration
Font Size	Specifies the font size of the text. Enter an integer value to change the size. Data type: Integer
Format	Specifies the display format of the parameter's value. The format varies with the parameter's data type and can be defined manually. Data type: String
Horizontal Alignment	Specifies the horizontal justification of the text in the parameter. Choose an option from the drop-down list. Data type: Enumeration
Italic	Specifies whether to make the text italic. Data type: Boolean
Strikethrough	Specifies whether to add a strikethrough line to the text. Data type: Boolean
Underline	Specifies whether to underline the text. Data type: Boolean
Vertical Alignment	Specifies the vertical justification of the text in the parameter. Choose an option from the drop-down list. Data type: Enumeration
Word Wrap	Specifies whether to wrap the text to the width. Data type: Boolean
Padding	
Bottom Padding	Specifies the space in inches or centimeters between the text and the bottom border of the parameter. Enter a numeric value to change the padding. Data type: Float
Left Padding	Specifies the space in inches or centimeters between the text and the left border of the parameter. Enter a numeric value to change the padding. Data type: Float
Right Padding	Specifies the space in inches or centimeters between the text and the right border of the parameter. Enter a numeric value to change the padding. Data type: Float
Top Padding	Specifies the space in inches or centimeters between the text and the top border of the parameter. Enter a numeric value to change the padding. Data type: Float
Geometry	
Height	Specifies the height of the parameter, which takes effect when the parameter is inserted into a report. Enter a numeric value to change the height. The default is 0 which means a proper height will be specified to the parameter by JReport automatically. Data type: Float
Width	Specifies the width of the parameter, which takes effect when the parameter is inserted into a report. Enter a numeric value to change the width. The default is 0 which means a proper width will be specified to the parameter by JReport automatically. Data type: Float
Color	
Background	Specifies the background color of the parameter. Choose a color from the drop-down list, enter a hexadecimal RGB value (0xCCAA00), or select Custom from the drop-down list to specify the color. Data type: String
Foreground	Specifies the foreground color of the parameter. Choose a color from the drop-down list, enter a hexadecimal RGB value (0xCCAA00), or select Custom from the drop-down list to specify the color. Data type: String
Border	
Border Color	Specifies the color of the border of the parameter. Choose a color from the drop-down list, enter a hexadecimal RGB value (0xCCAA00), or select Custom from the drop-down list to specify the color. Data type: String
Border Thickness	Specifies the width of the border in inches or centimeters. Enter a numeric value to change the thickness. Data type: Float
Bottom Line	Specifies the line style of the bottom border of the parameter. Choose a style from the drop-down list. Data type: Enumeration
Left Line	Specifies the line style of the left border of the parameter. Choose a style from the drop-down list. Data type: Enumeration
Right Line	Specifies the line style of the right border of the parameter. Choose a style from the drop-down list. Data type: Enumeration
Shadow	Specifies whether or not to draw two shadowed borders, beneath and to the right of the parameter. Data type: Boolean
Shadow Color	Specifies the color of the border shadow. Choose a color from the drop-down list, enter a hexadecimal RGB value (0xCCAA00), or select Custom from the drop-down list to specify the color. Data type: String
Top Line	Specifies the line style of the top border of the parameter. Choose a style from the drop-down list. Data type: Enumeration
Pattern	
Pattern Color	Specifies the color with which to draw a pattern to fill the parameter. Choose a color from the drop-down list, enter a hexadecimal RGB value (0xCCAA00), or select Custom from the drop-down list to specify the color. Data type: String
Pattern Style	Specifies the style of the pattern. Choose an option from the drop-down list. • **none** - No pattern will be applied to the parameter. • **50 %** - The parameter will be filled in the specified color of 50%-transparency. • **horizontal** - The pattern will be made of horizontal lines of the specified color. • **vertical** - The pattern will be made of vertical lines of the specified color. • **grid** - The pattern will be an overlapping of horizontal and vertical lines of the specified color. • **diagonal** - The pattern will be made of slash lines of the specified color. Data type: Enumeration
Others	
Data Mapping File	Specifies the data mapping file to the object for NLS use. • **<Auto>** - For a parameter bound with a single column or cascading columns, the data mapping file set for the display column is applied; for other parameters, no data mapping file is applied. For example, if the data mapping file name is Category_de_DE.properties, set the value as Category (without the language and locale part). Data type: String

User Defined Format

For example, you define a parameter of Date type and the following formats can be recognized: M/d/yy; MMM d, yyyy; MMMM d, yyyy; EEEE, MMMM d, yyyy. But if you want to enter a value like 1996. July.10, you should specify the user defined format as yyyy.MMMM.d. But remember it is the entering format, not the display format. To customize the display format, go to its Format property in Report Inspector, enter your customized display format as yyyy.MMMM.d. Then when you enter 1996.July.10, it will be displayed as 1996.July.10.

For details about how to define a user format for different types of parameters, refer to User defined format for parameters.

Note: The letters standing for year, month, and so on are case sensitive and should follow the standard of Java java.text.DateFormat, java.text.NumberFormat, and so on.

Query

The properties of a query are:

Property Name	Description
Description	Specifies the description of the query. Data type: String
Enable SQL Statement Creator	Specifies whether or not the query uses the dynamic query interface to get the result set. When it is set to true, the query can be re-generated at runtime using the dynamic query interface (for more information, see Dynamic queries). Data type: Boolean
Maximum Duration	Specifies the maximum elapsed time allowed to fetch data from the data source when the query runs, measured in seconds. If the value is set to zero or is blank, it means the time will be unlimited. For usage about the property, see Limiting the query run time and number of records. Data type: Integer
Maximum Rows	Specifies the maximum number of rows that will be fetched from the data source when the query runs. If the value is set to zero or is blank, it means the number will be unlimited. For usage about the property, see Limiting the query run time and number of records. Data type: Integer
Name	Specifies the name of the query. Data type: String
Path Name	Specifies the path in pre-join that the query is created on (for more information, see Creating queries using pre-join). Data type: String
Push Down Group Query	Specifies whether to push down group level summary computations in reports that are created based on the query to the DBMS at runtime. Choose an option from the drop-down list. • default - The setting of the Push Down Group Query property on the JDBC connection where the query is created will be applied. • true - JReport will push down the group level summary computations to the DBMS if the DBMS can do the computations, otherwise, JReport will do the computations itself. • false - JReport will do the group level summary computations itself. Data type: Boolean
Read Only	Specifies whether or not the query will be read only. Data type: Enumeration
Share	Specifies whether or not to share the query (for details, see Locking queries). Data type: Boolean
Transaction Mode	Specifies the transaction mode for the query. Choose an option from the drop-down list. • **default** - Indicates the transaction information cannot be got from JDBC connection. • **none** - Indicates that transactions are not supported. • **read uncommitted** - Dirty reads, non-repeatable reads and phantom reads can occur. This mode will speed up the transaction of the catalog. • **read committed** - Dirty reads are prevented; non-repeatable reads and phantom reads can occur. • **repeatable read** - Dirty reads and non-repeatable reads are prevented; phantom reads can occur. • **serializable** - Dirty reads, non-repeatable reads and phantom reads are prevented. Data type: Enumeration

The following topic shows the properties of a query column in detail:

- Query column

Query column

There are two kinds of columns in a query: table columns and computed columns, the properties of which are listed as follows. Note that, all the properties are read only for table columns, and some are editable for computed columns.

Property Name	Description
Array	Specifies whether the column is an array type or not. Data type: Boolean
Column Name	Indicates the name of the column in the raw database. This property is read only.
Currency	Displays whether to control the SQL type of formulas or summaries in which the BigDecimal type fields are imported. • **true** - The formula or summary which is built with the BigDecimal type field will be BigDecimal type, and its SQL Type value will be set to 3. • **false** - The normal data type will be used for the formula or summary. Data type: Boolean
Description	Specifies the description of the column. Data type: String
Display Width	Specifies the display width of the column. Data type: Float
Expression	Specifies the statement of the formula. Available to computed column only. Data type: String
Name	Shows the display name of the column. This property is read only.
Nullable	Specifies the nullability of the column's value. Data type: Enumeration
Precision	Specifies the precision of the column's value. The default value comes from data source meta data and it specifies the column's largest number of digits. Data type: Integer
Scale	Specifies the number of digits to the right of the decimal point for the column's value. Data type: Integer
SQL Type	Specifies the SQL type of the column defined in Java. Choose an option from the drop-down list. Data type: Integer

Query modifier

The properties of a query modifier are:

Property Name	Description
Data Source Name	Specifies the data source that includes this query modifier. Data type: String
Description	Specifies the description of the query modifier. Data type: String
Name	Specifies the name of the query modifier. Data type: String
Query	Specifies the query which the query modifier refers to. Data type: String
Query Modifier	Specifies the WHERE string of the query modifier. Data type: String
Structured	Specifies the type for defining the WHERE string. If true, it is Structured; if false, it is String. Data type: Boolean

Stored procedure

The properties of a stored procedure are:

Property Name	Description
Connection Name	Specifies the connection to execute the stored procedure. Data type: String
Description	Specifies the description of the stored procedure. Data type: String
Maximum Duration	Specifies the maximum elapsed time allowed to fetch data from the data source when the stored procedure runs, measured in seconds. If the value is set to zero or is blank, it means the time will be unlimited. For usage about the property, see Limiting the query run time and number of records. Data type: Integer
Maximum Rows	Specifies the maximum number of rows that will be fetched from the data source when the stored procedure runs. If the value is set to zero or is blank, it means the number will be unlimited. For usage about the property, see Limiting the query run time and number of records. Data type: Integer
Name	Specifies the name of the stored procedure. Data type: String
Procedure Name	Specifies the stored procedure name in the raw database. Data type: String
Qualifier	Specifies the database's catalog name of the stored procedure. Data type: String
Read Only	Specifies whether or not the stored procedure will be read only. Choose an option from the drop-down list. • **read&write** - Users will be able to access the database in read-write mode. • **default** - Accepts the settings from the database that you are connecting to. • **read only** - Users will only be able to access the database in read-only mode. Setting the property value to read only will speed up the transaction of the catalog. Data type: Enumeration
Schema	Specifies the stored procedure schema. Data type: String
Transaction Mode	Specifies the transaction mode for the stored procedure. Choose an option from the drop-down list. • **default** - Indicates the transaction information cannot be got from JDBC connection. • **none** - Indicates that transactions are not supported. • **read uncommitted** - Dirty reads, non-repeatable reads and phantom reads can occur. This mode will speed up the transaction of the catalog. • **read committed** - Dirty reads are prevented; non-repeatable reads and phantom reads can occur. • **repeatable read** - Dirty reads and non-repeatable reads are prevented; phantom reads can occur. • **serializable** - Dirty reads, non-repeatable reads and phantom reads are prevented. Data type: Enumeration

A stored procedure contains several columns, the properties of which are shown in the following topic:

- Stored procedure column

Stored procedure column

The properties of a stored procedure column are:

Property Name	Description
General	
Array	Specifies whether the column is an array type or not. Data type: Boolean
Column Index	Specifies the column index in the result set. Data type: Integer
Column Name	Specifies the name of the column in the raw data source. Data type: String
Currency	Specifies whether to control the SQL type of formulas or summaries in which the BigDecimal type fields are imported. • **true** - The formula or summary which is built with the BigDecimal type field will be BigDecimal type, and its SQL Type value will be set to 3. • **false** - The normal data type will be used for the formula or summary. Data type: Boolean
Description	Specifies the description of the column. Data type: String
Display Width	Specifies the display width of the column. Enter a numeric value to change the width. Data type: Float
Length	Specifies the length of the column's value, in bytes. Data type: Integer
Name	Specifies the mapped name of the column in the JReport catalog. Data type: String
Nullable	Specifies the nullability of the column's value. Choose an option from the drop-down list. Data type: Enumeration
Number Base	Specifies the number base of the column. Data type: Integer
Precision	Specifies the precision of the column's value. The default value comes from data source meta data and it specifies the column's largest number of digits. Data type: Integer
Scale	Specifies the number of digits to the right of the decimal point for the column's value. Data type: Integer
SQL Type	Specifies the SQL type of the column defined in Java. Choose an option from the drop-down list. Data type: Integer
Text Format	
Auto Fit	Specifies whether to adjust the width and height of the column according to the contents. Data type: Boolean
Bold	Specifies whether to make the text bold. Data type: Boolean
Font Face	Specifies the font of the text. Choose an option from the drop-down list. Data type: Enumeration
Font Size	Specifies the font size of the text. Enter an integer value to change the size. Data type: Integer
Format	Specifies the display format of the column's value. The format varies with the column's data type and can be defined manually. Data type: String
Horizontal Alignment	Specifies the horizontal justification of the text in the column. Choose an option from the drop-down list. Data type: Enumeration
Italic	Specifies whether to make the text italic. Data type: Boolean
Strikethrough	Specifies whether to add a strikethrough line to the text. Data type: Boolean
Underline	Specifies whether to underline the text. Data type: Boolean
Vertical Alignment	Specifies the vertical justification of the text in the column. Choose an option from the drop-down list. Data type: Enumeration
Word Wrap	Specifies whether to wrap the text to the width. Data type: Boolean
Padding	
Bottom Padding	Specifies the space in inches or centimeters between the text and the bottom border of the column. Enter a numeric value to change the padding. Data type: Float
Left Padding	Specifies the space in inches or centimeters between the text and the left border of the column. Enter a numeric value to change the padding. Data type: Float
Right Padding	Specifies the space in inches or centimeters between the text and the right border of the column. Enter a numeric value to change the padding. Data type: Float
Top Padding	Specifies the space in inches or centimeters between the text and the top border of the column. Enter a numeric value to change the padding. Data type: Float
Geometry	
Height	Specifies the height of the column, which takes effect when the column is inserted into a report. By default, no value is specified to the property, which means a height will be assigned to the column by JReport automatically. Enter a numeric value to change the height. Data type: Float
Width	Specifies the width of the column, which takes effect when the column is inserted into a report. By default, no value is specified to the property, which means a height will be assigned to the column by JReport automatically. Enter a numeric value to change the width. Data type: Float
Color	
Background	Specifies the background color of the column. Choose a color from the drop-down list, enter a hexadecimal RGB value (0xCCAA00), or select Custom from the drop-down list to specify the color. Data type: String
Foreground	Specifies the foreground color of the column. Choose a color from the drop-down list, enter a hexadecimal RGB value (0xCCAA00), or select Custom from the drop-down list to specify the color. Data type: String
Border	
Border Color	Specifies the color of the border of the column. Choose a color from the drop-down list, enter a hexadecimal RGB value (0xCCAA00), or select Custom from the drop-down list to specify the color. Data type: String
Border Thickness	Specifies the width of the border in inches or centimeters. Enter a numeric value to change the thickness. Data type: Float
Bottom Line	Specifies the line style of the bottom border of the column. Choose a style from the drop-down list. Data type: Enumeration
Left Line	Specifies the line style of the left border of the column. Choose a style from the drop-down list. Data type: Enumeration
Right Line	Specifies the line style of the right border of the column. Choose a style from the drop-down list. Data type: Enumeration
Shadow	Specifies whether or not to draw two shadowed borders, beneath and to the right of the column. Data type: Boolean
Shadow Color	Specifies the color of the border shadow. Choose a color from the drop-down list, enter a hexadecimal RGB value (0xCCAA00), or select Custom from the drop-down list to specify the color. Data type: String
Top Line	Specifies the line style of the top border of the column. Choose a style from the drop-down list. Data type: Enumeration
Pattern	
Pattern Color	Specifies the color with which to draw a pattern to fill the column. Choose a color from the drop-down list, enter a hexadecimal RGB value (0xCCAA00), or select Custom from the drop-down list to specify the color. Data type: String
Pattern Style	Specifies the style of the pattern. Choose an option from the drop-down list. • **none** - No pattern will be applied to the column. • **50%** - The column will be filled in the specified color of 50%-transparency. • **horizontal** - The pattern will be made of horizontal lines of the specified color. • **vertical** - The pattern will be made of vertical lines of the specified color. • **grid** - The pattern will be an overlapping of horizontal and vertical lines of the specified color. • **diagonal** - The pattern will be made of slash lines of the specified color. Data type: Enumeration
Others	
Data Mapping File	Specifies the data mapping file to the object for NLS use. For example, if the data mapping file name is Category_de_DE.properties, set the value as Category (without the language and locale part). Data type: String

Summary

The properties of a summary are:

Property Name	Description
General	
Description	Specifies the description of the summary. Data type: String
Display Width	Specifies the display width of the summary. Enter a numeric value to change the width. Data type: Float
Expression	Displays the statement of the summary. This property is read only.
Function	Specifies the function of the summary (for more information, refer to Summary functions). Data type: String
Group By	Specifies the criteria (break-by group) that the summary will be performed. Data type: String
Length	Specifies the length of the summary's value, in bytes. Data type: Integer
Name	Specifies the name of the summary. Data type: String
Number Base	Specifies the number base of the summary. Data type: Integer
Precision	Specifies the precision of the summary's value. The default value comes from data source meta data and it specifies the value's largest number of digits. Data type: Integer
Scale	Specifies the number of digits to the right of the decimal point for the summary's value. Data type: Integer
Special Function	If the group by field is of Numeric, String, Date or Time type, you can further define to calculate the summary by specifying a special function. Data type: String
SQL Type	Specifies the SQL type of the summary defined in Java. Choose an option from the drop-down list. Data type: Integer
Summary On	Specifies the field on which the summary performs. Data type: String
Text Format	
Auto Fit	Specifies whether to adjust the width and height of the summary according to the contents. Data type: Boolean
Bold	Specifies whether to make the text bold. Data type: Boolean
Font Face	Specifies the font of the text. Choose an option from the drop-down list. Data type: Enumeration
Font Size	Specifies the font size of the text. Enter an integer value to change the size. Data type: Integer
Format	Specifies the display format of the summary's value. The format varies with the summary's data type and can be defined manually. Data type: String
Horizontal Alignment	Specifies the horizontal justification of the text in the summary. Choose an option from the drop-down list. Data type: Enumeration
Italic	Specifies whether to make the text italic. Data type: Boolean
Strikethrough	Specifies whether to add a strikethrough line to the text. Data type: Boolean
Underline	Specifies whether to underline the text. Data type: Boolean
Vertical Alignment	Specifies the vertical justification of the text in the summary. Choose an option from the drop-down list. Data type: Enumeration
Word Wrap	Specifies whether to wrap the text to the width. Data type: Boolean
Padding	
Bottom Padding	Specifies the space in inches or centimeters between the text and the bottom border of the summary. Enter a numeric value to change the padding. Data type: Float
Left Padding	Specifies the space in inches or centimeters between the text and the left border of the summary. Enter a numeric value to change the padding. Data type: Float
Right Padding	Specifies the space in inches or centimeters between the text and the right border of the summary. Enter a numeric value to change the padding. Data type: Float
Top Padding	Specifies the space in inches or centimeters between the text and the top border of the summary. Enter a numeric value to change the padding. Data type: Float
Geometry	
Height	Specifies the height of the summary, which takes effect when the summary is inserted into a report. Enter a numeric value to change the height. The default is 0 which means a proper height will be specified to the summary by JReport automatically. Data type: Float
Width	Specifies the width of the summary, which takes effect when the summary is inserted into a report. Enter a numeric value to change the width. The default is 0 which means a proper width will be specified to the summary by JReport automatically. Data type: Float
Color	
Background	Specifies the background color of the summary. Choose a color from the drop-down list, enter a hexadecimal RGB value (0xCCAA00), or select Custom from the drop-down list to specify the color. Data type: String
Foreground	Specifies the foreground color of the summary. Choose a color from the drop-down list, enter a hexadecimal RGB value (0xCCAA00), or select Custom from the drop-down list to specify the color. Data type: String
Border	
Border Color	Specifies the color of the border of the summary. Choose a color from the drop-down list, enter a hexadecimal RGB value (0xCCAA00), or select Custom from the drop-down list to specify the color. Data type: String
Border Thickness	Specifies the width of the border in inches or centimeters. Enter a numeric value to change the thickness. Data type: Float
Bottom Line	Specifies the line style of the bottom border of the summary. Choose a style from the drop-down list. Data type: Enumeration
Left Line	Specifies the line style of the left border of the summary. Choose a style from the drop-down list. Data type: Enumeration
Right Line	Specifies the line style of the right border of the summary. Choose a style from the drop-down list. Data type: Enumeration
Shadow	Specifies whether or not to draw two shadowed borders, beneath and to the right of the summary. Data type: Boolean
Shadow Color	Specifies the color of the border shadow. Choose a color from the drop-down list, enter a hexadecimal RGB value (0xCCAA00), or select Custom from the drop-down list to specify the color. Data type: String
Top Line	Specifies the line style of the top border of the summary. Choose a style from the drop-down list. Data type: Enumeration
Pattern	
Pattern Color	Specifies the color with which to draw a pattern to fill the summary. Choose a color from the drop-down list, enter a hexadecimal RGB value (0xCCAA00), or select Custom from the drop-down list to specify the color. Data type: String
Pattern Style	Specifies the style of the pattern. Choose an option from the drop-down list. • **none** - No pattern will be applied to the summary. • **50%** - The summary will be filled in the specified color of 50%-transparency. • **horizontal** - The pattern will be made of horizontal lines of the specified color. • **vertical** - The pattern will be made of vertical lines of the specified color. • **grid** - The pattern will be an overlapping of horizontal and vertical lines of the specified color. • **diagonal** - The pattern will be made of slash lines of the specified color. Data type: Enumeration
Others	
Data Mapping File	Specifies the data mapping file to the object for NLS use. For example, if the data mapping file name is Category_de_DE.properties, set the value as Category (without the language and locale part). Data type: String

Table/View/Synonym

The properties of a table/view/synonym are:

Property Name	Description
Collection Name	Specifies the name of the collection in the MongoDB database. Available for tables in MongoDB connection only. Data type: String
Database Name	Specifies the name of the database in the MongoDB connection. Available for tables in MongoDB connection only. • If the database you specify exists in the MongoDB data source, it will connect to the new database. • If the database you specify does not exist in the MongoDB data source, no data will be got. Data type: String
Description	Specifies the description of the table/view/synonym. Data type: String
Element Path	• For tables in MongoDB connection: Specifies the element name path from the top level document if the table is mapped to a document array element. The path are the element names separated by ".". It will be blank if the table is mapped to the top level element. '\' is used as an escaped char. If the '.' and '\' are used in the element name, it will be recorded as '\.' and '\\'. • For tables in JSON connection: Displays the element path from the root element to current element. This property is read only. Data type: String
Linked Parameter	Specifies the parameter to bind it to the parameter in OOJDBC. Not available for tables in JSON connection. Data type: String
Name	Specifies the mapped table/view/synonym name in the JReport catalog. Data type: String
Qualifier	Specifies the name of the catalog which contains the table/view/synonym. Not available for tables in MongoDB connection and JSON connection. Data type: String
Schema	Specifies the name of the schema which contains the table/view/synonym. Not available for tables in MongoDB connection and JSON connection. Data type: String
Table Name	Specifies the name of the table/view/synonym in the raw database. Not available for tables in MongoDB connection and JSON connection. Data type: String
Type	Specifies the type of the table/view/synonym. The table type can be Table or Alias Table, the view type can be View or Alias View, and the synonym type can be Alias, Synonym or Alias Synonym. You can define whether an object is a table, a view or a synonym by selecting a type from the drop-down list, then the object will be put under the corresponding node automatically. Not available for tables in JSON connection. • **Table** Indicates the type is Table in the database. • **View** Indicates the type is View in the database. • **Alias** Indicates the type is Alias in the database. • **Alias Table** Indicates the type is Alias Table in the database. It will be created when a table is imported more than one time. • **Alias View** Indicates the type is Alias View in the database. It will be created when a view is imported more than one time. • **Synonym** Indicates the type is Synonym in the database. • **Alias Synonym** Indicates the type is Alias Synonym in the database. It will be created when a synonym is imported more than one time. Data type: Enumeration

A table/view/synonym contains several columns, the properties of which are shown in the following topic:

- Table/View/Synonym column

Table/View/Synonym column

The properties of a table/view/synonym column are:

Property Name	Description
General	
Array	Specifies whether the column is an array type or not. Data type: Boolean
Column Name	Specifies the name of the column in the raw database. Not available for table columns in MongoDB connection and JSON connection. Data type: String
Column Type	Displays the type of the column. This property is read only and available only for table columns in MongoDB connection and JSON connection.
Currency	Specifies whether to control the SQL type of formulas or summaries in which the BigDecimal type fields are imported. • **true** - The formula or summary which is built with the BigDecimal type field will be BigDecimal type, and its SQL Type value will be set to 3. • **false** - The normal data type will be used for the formula or summary. Data type: Boolean
Description	Specifies the description of the column. Data type: String
Display Width	Specifies the display width of the column. Enter a numeric value to change the width. Data type: Float
Element Name	• For table columns in MongoDB connection: Specifies the mapped element name in a document. If the column is mapped from an element of an embedded document element, the name will be a path from the embedded document element. For 'PrimaryKey' and 'ForeignKey' columns, it is blank. '\' is used as an escaped char. If the '.' and '\' are used in the element name, it will be recorded as '\.' and '\\'. • For table columns in JSON connection: Displays the mapped element name in JSON metadata. For the columns Primary Key and Foreign Key, it is blank. This property is read only. Data type: String
Name	Specifies the mapped name of the column in the JReport catalog. Data type: String
Nullable	Displays the nullability of the column's value. This property is read only.
Precision	Specifies the precision of the column's value. The default value comes from data source meta data and it specifies the column's largest number of digits. Data type: Integer
Scale	Specifies the number of digits to the right of the decimal point for the column's value. Data type: Integer
SQL Type	Specifies the SQL type of the column defined in Java. Choose an option from the drop-down list. Data type: Integer
Text Format	
Auto Fit	Specifies whether to adjust the width and height of the column according to the contents. Data type: Boolean
Bold	Specifies whether to make the text bold. Data type: Boolean
Font Face	Specifies the font of the text. Choose an option from the drop-down list. Data type: Enumeration
Font Size	Specifies the font size of the text. Enter an integer value to change the size. Data type: Integer
Format	Specifies the display format of the column's value. The format varies with the column's data type and can be defined manually. Data type: String
Horizontal Alignment	Specifies the horizontal justification of the text in the column. Choose an option from the drop-down list. Data type: Enumeration
Italic	Specifies whether to make the text italic. Data type: Boolean
Strikethrough	Specifies whether to add a strikethrough line to the text. Data type: Boolean
Underline	Specifies whether to underline the text. Data type: Boolean
Vertical Alignment	Specifies the vertical justification of the text in the column. Choose an option from the drop-down list. Data type: Enumeration
Word Wrap	Specifies whether to wrap the text to the width. Data type: Boolean
Padding	
Bottom Padding	Specifies the space in inches or centimeters between the text and the bottom border of the column. Enter a numeric value to change the padding. Data type: Float
Left Padding	Specifies the space in inches or centimeters between the text and the left border of the column. Enter a numeric value to change the padding. Data type: Float
Right Padding	Specifies the space in inches or centimeters between the text and the right border of the column. Enter a numeric value to change the padding. Data type: Float
Top Padding	Specifies the space in inches or centimeters between the text and the top border of the column. Enter a numeric value to change the padding. Data type: Float
Geometry	
Height	Specifies the height of the column, which takes effect when the column is inserted into a report. By default, no value is specified to the property, which means a height will be assigned to the column by JReport automatically. Enter a numeric value to change the height. Data type: Float
Width	Specifies the width of the column, which takes effect when the column is inserted into a report. By default, no value is specified to the property, which means a height will be assigned to the column by JReport automatically. Enter a numeric value to change the width. Data type: Float
Color	
Background	Specifies the background color of the column. Choose a color from the drop-down list, enter a hexadecimal RGB value (0xCCAA00), or select Custom from the drop-down list to specify the color. Data type: String
Foreground	Specifies the foreground color of the column. Choose a color from the drop-down list, enter a hexadecimal RGB value (0xCCAA00), or select Custom from the drop-down list to specify the color. Data type: String
Border	
Border Color	Specifies the color of the border of the column. Choose a color from the drop-down list, enter a hexadecimal RGB value (0xCCAA00), or select Custom from the drop-down list to specify the color. Data type: String
Border Thickness	Specifies the width of the border in inches or centimeters. Enter a numeric value to change the thickness. Data type: Float
Bottom Line	Specifies the line style of the bottom border of the column. Choose a style from the drop-down list. Data type: Enumeration
Left Line	Specifies the line style of the left border of the column. Choose a style from the drop-down list. Data type: Enumeration
Right Line	Specifies the line style of the right border of the column. Choose a style from the drop-down list. Data type: Enumeration
Shadow	Specifies whether or not to draw two shadowed borders, beneath and to the right of the column. Data type: Boolean
Shadow Color	Specifies the color of the border shadow. Choose a color from the drop-down list, enter a hexadecimal RGB value (0xCCAA00), or select Custom from the drop-down list to specify the color. Data type: String
Top Line	Specifies the line style of the top border of the column. Choose a style from the drop-down list. Data type: Enumeration
Pattern	
Pattern Color	Specifies the color with which to draw a pattern to fill the column. Choose a color from the drop-down list, enter a hexadecimal RGB value (0xCCAA00), or select Custom from the drop-down list to specify the color. Data type: String
Pattern Style	Specifies the style of the pattern. Choose an option from the drop-down list. • **none** - No pattern will be applied to the column. • **50%** - The column will be filled in the specified color of 50%-transparency. • **horizontal** - The pattern will be made of horizontal lines of the specified color. • **vertical** - The pattern will be made of vertical lines of the specified color. • **grid** - The pattern will be an overlapping of horizontal and vertical lines of the specified color. • **diagonal** - The pattern will be made of slash lines of the specified color. Data type: Enumeration
Others	
Data Mapping File	Specifies the data mapping file to the object for NLS use. For example, if the data mapping file name is Category_de_DE.properties, set the value as Category (without the language and locale part). Data type: String

User defined data source

The properties of a user defined data source are:

Property Name	Description
Class Name	Specifies the full name (including package name) of the class represented by the UDS. The class you entered should actually exist and can be found by JReport Designer, which means the class should be in the class path of the system environment. Data type: String
Description	Specifies the description of the UDS. Data type: String
Maximum Duration	Specifies the maximum elapsed time allowed to fetch data from the data source when the UDS runs, measured in seconds. If the value is set to zero or is blank, it means the time will be unlimited. For usage about the property, see Limiting the query run time and number of records. Data type: Integer
Maximum Rows	Specifies the maximum number of rows that will be fetched from the data source when the UDS runs. If the value is set to zero or is blank, it means the number will be unlimited. For usage about the property, see Limiting the query run time and number of records. Data type: Integer
Name	Specifies the name of the UDS. Data type: String
Parameter	Specifies the value of the UDS's parameter. Data type: String
Specify Metadata	Specifies whether or not to apply the column properties specified by users. Data type: Boolean

A UDS contains several columns, the properties of which are shown in the following topic:

- UDS column

UDS column

The properties of a UDS column are:

Property Name	Description
General	
Array	Specifies whether the column is an array type or not. Data type: Boolean
Auto Group Index	Specifies the index of automatic group. The start index is 0, and if do not group by this field automatically, the index is -1. Data type: Integer
Column Index	Specifies the column index in the result set. Data type: Integer
Column Mapping Index	Specifies the column index in the result set. Data type: Integer
Column Name	Specifies the name of the column in the raw data source. Data type: String
Currency	Specifies whether to control the SQL type of formulas or summaries in which the BigDecimal type fields are imported. • **true** - The formula or summary which is built with the BigDecimal type field will be BigDecimal type, and its SQL Type value will be set to 3. • **false** - The normal data type will be used for the formula or summary. Data type: Boolean
Description	Specifies the description of the column. Data type: String
Display Width	Specifies the display width of the column. Enter a numeric value to change the width. Data type: Float
Length	Specifies the length of the column's value, in bytes. Data type: Integer
Name	Specifies the mapped name of the column in the JReport catalog. Data type: String
Nullable	Specifies the nullability of the column's value. Choose an option from the drop-down list. Data type: Enumeration
Number Base	Specifies the number base of the column. Data type: Integer
Path String	Specifies the path of the column, which is used to identify the column in tree. Data type: String
Precision	Specifies the precision of the column's value. The default value comes from data source meta data and it specifies the column's largest number of digits. Data type: Integer
Scale	Specifies the number of digits to the right of the decimal point for the column's value. Data type: Integer
SQL Type	Specifies the SQL type of the column defined in Java. Choose an option from the drop-down list. Data type: Integer
Text Format	
Auto Fit	Specifies whether to adjust the width and height of the column according to the contents. Data type: Boolean
Bold	Specifies whether to make the text bold. Data type: Boolean
Font Face	Specifies the font of the text. Choose an option from the drop-down list. Data type: Enumeration
Font Size	Specifies the font size of the text. Enter an integer value to change the size. Data type: Integer
Horizontal Alignment	Specifies the horizontal justification of the text in the column. Choose an option from the drop-down list. Data type: Enumeration
Italic	Specifies whether to make the text italic. Data type: Boolean
Strikethrough	Specifies whether to add a strikethrough line to the text. Data type: Boolean
Underline	Specifies whether to underline the text. Data type: Boolean
Vertical Alignment	Specifies the vertical justification of the text in the column. Choose an option from the drop-down list. Data type: Enumeration
Word Wrap	Specifies whether to wrap the text to the width. Data type: Boolean
Padding	
Bottom Padding	Specifies the space in inches or centimeters between the text and the bottom border of the column. Enter a numeric value to change the padding. Data type: Float
Left Padding	Specifies the space in inches or centimeters between the text and the left border of the column. Enter a numeric value to change the padding. Data type: Float
Right Padding	Specifies the space in inches or centimeters between the text and the right border of the column. Enter a numeric value to change the padding. Data type: Float
Top Padding	Specifies the space in inches or centimeters between the text and the top border of the column. Enter a numeric value to change the padding. Data type: Float
Geometry	
Height	Specifies the height of the column, which takes effect when the column is inserted into a report. By default, no value is specified to the property, which means a height will be assigned to the column by JReport automatically. Enter a numeric value to change the height. Data type: Float
Width	Specifies the width of the column, which takes effect when the column is inserted into a report. By default, no value is specified to the property, which means a height will be assigned to the column by JReport automatically. Enter a numeric value to change the width. Data type: Float
Color	
Background	Specifies the background color of the column. Choose a color from the drop-down list, enter a hexadecimal RGB value (0xCCAA00), or select Custom from the drop-down list to specify the color. Data type: String
Foreground	Specifies the foreground color of the column. Choose a color from the drop-down list, enter a hexadecimal RGB value (0xCCAA00), or select Custom from the drop-down list to specify the color. Data type: String
Border	
Border Color	Specifies the color of the border of the column. Choose a color from the drop-down list, enter a hexadecimal RGB value (0xCCAA00), or select Custom from the drop-down list to specify the color. Data type: String
Border Thickness	Specifies the width of the border in inches or centimeters. Enter a numeric value to change the thickness. Data type: Float
Bottom Line	Specifies the line style of the bottom border of the column. Choose a style from the drop-down list. Data type: Enumeration
Left Line	Specifies the line style of the left border of the column. Choose a style from the drop-down list. Data type: Enumeration
Right Line	Specifies the line style of the right border of the column. Choose a style from the drop-down list. Data type: Enumeration
Shadow	Specifies whether or not to draw two shadowed borders, beneath and to the right of the column. Data type: Boolean
Shadow Color	Specifies the color of the border shadow. Choose a color from the drop-down list, enter a hexadecimal RGB value (0xCCAA00), or select Custom from the drop-down list to specify the color. Data type: String
Top Line	Specifies the line style of the top border of the column. Choose a style from the drop-down list. Data type: Enumeration
Pattern	
Pattern Color	Specifies the color with which to draw a pattern to fill the column. Choose a color from the drop-down list, enter a hexadecimal RGB value (0xCCAA00), or select Custom from the drop-down list to specify the color. Data type: String
Pattern Style	Specifies the style of the pattern. Choose an option from the drop-down list. • **none** - No pattern will be applied to the column. • **50%** - The column will be filled in the specified color of 50%-transparency. • **horizontal** - The pattern will be made of horizontal lines of the specified color. • **vertical** - The pattern will be made of vertical lines of the specified color. • **grid** - The pattern will be an overlapping of horizontal and vertical lines of the specified color. • **diagonal** - The pattern will be made of slash lines of the specified color. Data type: Enumeration
Others	
Data Mapping File	Specifies the data mapping file to the object for NLS use. For example, if the data mapping file name is Category_de_DE.properties, set the value as Category (without the language and locale part). Data type: String

Web service connection

The properties of a web service connection are:

Property Name	Description
General	
Date Format	Specifies the default format for data of Date type. Data type: String
Default	Specifies whether the connection is the default connection in the current data source. By default, the first added connection in a data source is the default connection of the data source. A data source can have zero or one default connection. Data type: Boolean
Description	Specifies the description of the WSDL connection. Data type: String
Name	Specifies the name of the connection which, by default, is the same as the connection URL, but can also be a user defined name for the connection. Data type: String
Name Pattern	Specifies whether or not catalog or schema is used in data manipulation. Choose an option from the drop-down list. • **unqualified name** - Neither catalog nor schema is included in data manipulation. Example: SELECT t.c FROM t • **2-part names** - Uses schema in data manipulation. Example: SELECT schema.t.c FROM schema.t • **3-part names** - Uses both catalog and schema in data manipulation. Example: SELECT catalog.schema.t.c from catalog.schema.t Data type: Enumeration
Password	Specifies the password used for accessing the WSDL file through the WSDL URI. Data type: String
Time Format	Specifies the default Time format corresponding to the database. Data type: String
Time Out	Specifies how long to wait to get the WSDL file, in seconds. Data type: Integer
Timestamp Format	Specifies the default format for data of Timestamp type. Data type: String
URI	Displays the URI to get the WSDL file. This property is read only.
User	Specifies the user name used for accessing the WSDL file through the WSDL URI. Data type: String
Security Configuration	
Client Key Alias Name	Displays the alias name which is used as client signature in the key store. This property is read only.
Client Key Alias Password	Displays the password for the alias name which is used as client signature in the key store. This property is read only.
Key Store File	Displays the URI to get the key store file. This property is read only.
Key Store Password	Displays the password to open the key store file. This property is read only.
Key Store Type	Displays the type for the key store. This property is read only.
Security Password	Displays the password for the user name token to be used. This property is read only.
Security User Name	Displays the user name for the user name token to be used. This property is read only.
Server Key Alias Name	Displays the alias name which is used to get the server-side certification or public key in the key store. This property is read only.
Server Key Alias Password	Displays the password for the alias name which is used to get the server-side certification or public key in the key store. This property is read only.

XML connection

The properties of an XML connection are:

Property Name	Description
Date Format	Specifies the default format for data of Date type. Data type: String
Default	Specifies whether the connection is the default connection in the current data source. By default, the first added connection in a data source is the default connection of the data source. A data source can have zero or one default connection. Data type: Boolean
Description	Specifies the description of the XML connection. Data type: String
Name	Specifies the full path of the XSD file. Data type: String
Name Pattern	Specifies whether or not catalog or schema is used in data manipulation. Choose an option from the drop-down list. • **unqualified name** - Neither catalog nor schema is included in data manipulation. Example: SELECT t.c FROM t • **2-part names** - Uses schema in data manipulation. Example: SELECT schema.t.c FROM schema.t • **3-part names** - Uses both catalog and schema in data manipulation. Example: SELECT catalog.schema.t.c from catalog.schema.t Data type: Enumeration
Time Format	Specifies the default format for data of Time type. Data type: String
Timestamp Format	Specifies the default format for data of Timestamp type. Data type: String

- XML schema

XML schema

The properties of an XML schema are:

Property Name	Description
Data Type	Specifies the data type of the element or attribute. This property on the reference of element cannot be modified.
Default Value	Displays the value for the element or attribute, of which there is no specific value in the XML instance. This property is read only.
Fixed Value	Displays the value for the element or attribute. This property is read only.
Format Pattern	Displays the format pattern for data of Date, Time or DateTime type. This property is read only.
Format Pattern Type	Specifies the format pattern type for the element, attribute or reference of element. It is editable only for data of Date, Time or DateTime type.
Is Multiple	Indicates whether or not the element occurs more than once in the XML instance. This property is read only.
List	Specifies whether or not to set the data type of the element or attribute as List. This property on the reference of element cannot be modified.
List Delimiter	Displays the delimiter to separate items in a list. This property is read only.
Name	Displays the name of the element, attribute or reference of element in the XML instance. This property is read only.

Related topics:

- Supplementing information for the XML schema

Appendixes

The following are some additional information about JReport Designer:

- Appendix 1: Details of format patterns
- Appendix 2: Barcode symbologies
- Appendix 3: Parameters of user defined web actions

Appendix 1: Details of format patterns

A format pattern is a string and can be used for specifying patterns to format strings, which are in XML instance. JReport Format Pattern only supports specifying patterns to format time data type strings, such as date, time and datetime. You can customize the format pattern for date, time and datetime formats or use default ones.

In the process of transforming an XML schema to relational tables, you can choose to customize the format patterns for date, time and datetime formats in the XML schema or apply the default ones, which will comply with the W3C XML Schema 1.1 specification (for details, refer to the page http://www.w3.org/TR/2004/REC-xmlschema-2-20041028/datatypes.html#dateTime). See the following contents of the section to know more about the customized format patterns.

Customized format patterns

You can specify the date, time and datetime formats in XML schema by the date, time and datetime pattern strings or customize the format pattern by the Format Pattern dialog. In the dialog, the pattern string you select from the Pattern Fragments list box will be interpreted, and the delimiter that you input will not be interpreted. Any character is acceptable and it will be quoted if necessary.

Within date, time and datetime pattern strings in the XML schema, unquoted letters from A to Z and from a to z are interpreted as pattern letters representing the components of a date, time or datetime string. Text can be quoted using single quotes (') to avoid interpretation. '' represents a single quote. All other characters are not interpreted; they're simply copied into the output string during formatting or matched against the input string during parsing the XML schema.

DateTime format patterns

A format pattern consists of some pattern strings and delimiters. The following pattern strings that JReport Designer provides you to use in the format pattern: G, yyyy, yy, MMMM, MM, dd, D, WW, ww, F, E, a, HH, kk, KK, hh, mm, ss, SSS, z, Z.

The following table describes the detail information of the pattern strings listed above:

Letter	Date Component	Presentation	Examples
G	Era designator	Text	AD
y	Year	Year	1996; 96
M	Month in year	Month	July; Jul; 07
w	Week in year	Number	27
D	Day in year	Number	189
d	Day in month	Number	10
F	Day of week in month	Number	2
E	Day in Week	Text	Tuesday; Tue
a	Am/pm marker	Text	PM
H	Hour in day (0-23)	Number	0
k	Hour in day (1-24)	Number	24
K	Hour in am/pm (0-11)	Number	0
h	Hour in am/pm (1-12)	Number	12
m	Minute in hour	Number	30
s	Second in minute	Number	55
S	Millisecond	Number	978
z	Time zone	General time zone	Pacific Standard Time; PST; GMT-08:00
Z	Time zone	RFC 822 time zone	-0800

- **Year**
  The pattern string of year can be formatted as two types - yyyy and yy:
  - If the pattern string - yyyy is used to format year, the year will be interpreted literally, regardless of the number of digits. Therefore, if the pattern MM/dd/yyyy is used, 08/25/22 will be parsed to Aug 25, 22 A.D.
  - If the abbreviated pattern string - yy is used to format year, the abbreviated year relative to some century must be interpreted. To do this, dates will be adjusted to be within 80 years before and 20 years after the current time. For example, a pattern of MM/dd/yy and the current time is Aug 25, 1998, the string 05/21/15 would be interpreted as May 21, 2015 while the string 06/02/88 would be interpreted as Jun 2, 1988.

  When the year string of a datetime data in the XML schema is parsing, only strings consisting of exactly two digits will be parsed into the default century. Any other numeric string, such as a one-digit string, a three or more digit string, or a two-digit string but one is a sign (for example, -2), is interpreted literally. So 03/06/4 or 03/06/004 is parsed, using the pattern of MM/dd/yy, as Mar 6, 4 AD. Also, 03/06/-4 is parsed as Mar 6, 5 BC.

- **Month**
  If the pattern string is MMMM, the month is interpreted as text; if the pattern string is MM, the month will be interpreted as a number.

- **General time zone**
  Time zones are interpreted as text if they have names. For time zones representing a GMT offset value, the following syntax is used:

```
GMTOffsetTimeZone:
    GMT Sign Hours: Minutes
Sign: one of
    + -
Hours:
    Digit
    Digit Digit
Minutes:
    Digit Digit
Digit: one of
    0 1 2 3 4 5 6 7 8 9
```

  Hours must be between 0 and 23, and Minutes must be between 00 and 59.

- **RFC 822 time zone:**

```
RFC822TimeZone:
   Sign TwoDigitHours Minutes
TwoDigitHours:
   Digit Digit
```

  TwoDigitHours must be between 00 and 23.

Format pattern also supports localized date, time and datetime pattern strings. In these strings, the text presentation of pattern letters described in the above table may be varied with the locale.

Date format patterns

The representation of the format pattern for date data type in XML schema is same as datetime but the pattern strings are less than it. Only the following strings can be used: G, yyyy, yy, MMMM, MM, dd, D, WW, ww, F, E.

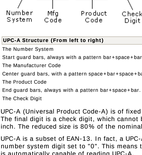
Time format patterns

The representation of the format pattern for time data type in XML schema is same as datetime but the pattern strings are less than it. Only the following strings can be used: a, HH, kk, KK, hh, mm, ss, SSS, z, Z.

Appendix 2: Barcode symbologies

JReport Designer supports ten types of barcodes. This appendix briefly describes each one. For more detailed information on the symbologies of barcodes, refer to professional barcode technical documentation.

UPC-A



UPC-A Structure (From left to right)
The Number System
Start guard bars, always with a pattern bar+space+bar.
The Manufacturer Code
Center guard bars, with a pattern space+bar+space+bar+space.
The Product Code
End guard bars, always with a pattern bar+space+bar.
The Check Digit

UPC-A (Universal Product Code-A) is of fixed length. It encodes a twelve-digit numeric only number. The final digit is a check digit, which cannot be omitted. The usual height of a UPC-A bar code is one inch. The reduced size is 80% of the nominal size.

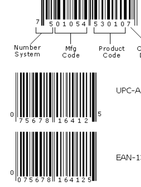
UPC-A is a subset of EAN-13. In fact, a UPC-A barcode is an EAN-13 barcode with the first EAN-13 number system digit set to "0". This means that any hardware or software capable of reading EAN-13 is automatically capable of reading UPC-A.

Usually, the number system digit is printed on the left of the barcode, and the check digit on the right; while the manufacturer and product codes are printed just below the barcode, separated by a guard bar.

Both the left and the right guard bar are encoded with 101 (bar space bar).

Usual X dimension is 13 mils. A magnification factor of 0.8 to 2.0 is permitted and, as a result, enables a printable X dimension values range of 10.4 to 24 mils.

UPC-E

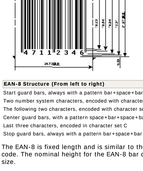


UPC-E Structure (From left to right)
Start guard bars, always with a pattern bar+space+bar.
Left half, five digits calculated from the equivalent UPC number.
Check digit.
Stop guard bars, always with a pattern bar+space+bar.

UPC-E (Universal Product Code-E) is fixed in length and is a compressed six-digit code used for marking small packages including magazines and paperback books. The printed value of the UPC-E code is a twelve-digit code. UPC-E symbols are UPC-A symbols that have been zero suppressed (that is, consecutive zeros are not included in the symbol). The nominal height for the UPC-E bar code is one inch. The reduced size is 80% of the nominal size.

The left guard bar is encoded with 101, and the right guard bar is encoded with 010101.

EAN-13



EAN-13 Structure (From left to right)
The Number System
Start guard bars, always with a pattern bar+space+bar
The Manufacturer Code
Central guard bars, always with a pattern space+bar+space+bar+space
The Product Code
The Check Digit
End guard bars, always with a pattern bar+space+bar

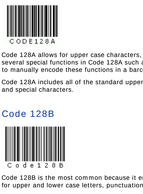
The EAN/JAN-13 is of fixed length and is similar to the UPC-A symbology, but encodes a 13th digit. Also, the 12th and 13th digits of an EAN-13 may represent a country code in its entirety or just the beginning of one, and can vary from 2 to 3 digits. The codes 00-04 and 06-09 have been assigned to the United States. The nominal height for the EAN-13 bar code is one inch. The reduced size is 80% of the nominal size.

EAN-8

EAN-8 Structure (From left to right)
Start guard bars, always with a pattern bar+space+bar
Two number system characters, encoded with character set A
The following two characters, encoded with character set A
Center guard bars, with a pattern space+bar+space+bar+space
Last three characters, encoded in character set C
Stop guard bars, always with a pattern bar+space+bar

The EAN-8 is fixed length and is similar to the UPC-E code, but includes two more digits for the country code. The nominal height for the EAN-8 bar code is one inch. The reduced size is 80% of the nominal size.

Code 39



Code39 Structure (From left to right)
A start character -- the asterisk (*)
Message encoded
A stop character -- the second asterisk (*)

Code 39 is of variable length and is the most frequently used symbology in industrial bar code systems today. The principal feature is that it encodes messages using a full alphanumeric character set. Three of the nine elements (bars) are wide and six elements are narrow. The Code 39 barcode uses four special characters "$", "/", "+", and "%" which allow a full ASCII character set when paired with alphanumeric characters.

Code 39 is designed to encode 26 upper case letters, 10 digits and 7 special characters:

A, B, C, D, E, F, G, H, I, J, K, L, M, N, O, P, Q, R, S, T, U, V, W, X, Y, Z
0, 1, 2, 3, 4, 5, 6, 7, 8, 9
-, ., *, $, /, +, %, SPACE.

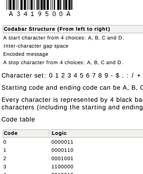
It is called code 39 or three of nine because each encoded character is made up of 5 bars and 4 spaces to make a total of 9 elements. Also 3 out of the 9 elements are always wide.

The height of the bars must be at least 0.15 times the symbol's length or 0.25 inches, whichever is larger.

Code 39 is a discrete symbology. Two adjacent characters are separated by an inter-character gap. To achieve good barcode quality, the width of the inter-character gap should equal the width of the narrowest element, called X.

Code39 requires a starting quiet zone with a minimum 10 times the X dimension or 0.10 inch, whichever is greater. The same width requirement also applies to the trailing quiet zone.

Code 128



Code128 Structure (From left to right)
A start character
Message encoded
Check character
Stop Character
Termination bar (bar+space+bar)

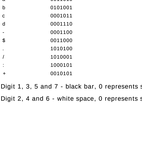
Code 128 is of variable length and encodes a full 128 ASCII character set. Each character is represented by 11 modules that can be one of four bar widths. Also, the encoding should not exceed 232 characters (including the starting and ending codes).

Among all the common linear symbologies, Code 128 is the most flexible. It supports both alpha and numeric characters easily, has the highest number of characters per inch, and is of variable length. Code 128 is usually the best choice when implementing a new symbology.

Check code is optional. Refer to the Code 128 code table to calculate it.

In an open system, the minimum value of the X dimension is 7.5 mils, and the minimum bar height is 15 percent of the symbol length or 0.25 inches, whichever is greater. The starting and trailing quiet zones are at least 0.25 inches wide.

Code 128A



Code 128A allows for upper case characters, punctuation and numbers. Lower case characters create several special functions in Code 128A such as a return or tab. It may be necessary to use Code 128A to manually encode these functions in a barcode.

Code 128A includes all of the standard upper case alphanumeric keyboard characters plus the control and special characters.

Code 128B

Code 128B is the most common because it encodes everything from ASCII 32 to ASCII 126. It allows for upper and lower case letters, punctuation, numbers and a few select functions.

Code 128B includes all of the standard upper and lower case alphanumeric keyboard characters and special characters.

Code 128C

Code 128C encodes *only* numbers containing an even number of characters. Since the numbers are "interleaved" into pairs, with two numbers encoded into every barcode character, a very high-density barcode is created. If the number that is to be encoded does not contain an even number of characters, a leading zero will be added.

Code 128C includes a set of 100 numeric data character pairs from 00 through 99 inclusive, and also includes special characters. This enables double-density numeric digits (two digits per bar coded character) to be encoded. Data encoded in a Code 128C must contain an even number of characters. If the data contains an odd number of characters, a zero (0) will be inserted at the beginning of the bar code.

Codabar

Codabar Structure (From left to right)
A start character from 4 choices: A, B, C and D.
Inter-character gap space
Encoded message
A stop character from 4 choices: A, B, C and D.

Character set: 0 1 2 3 4 5 6 7 8 9 - $. : / + *

Starting code and ending code can be A, B, C or D.

Every character is represented by 4 black bars + 3 white spaces. Variable length, maximum 32 characters (including the starting and ending codes).

Code table

Code	Logic
0	0000011
1	0000110
2	0001001
3	1100000
4	0010010
5	1000010
6	0100001
7	0100100
8	0110000
9	1001000
a	0011010
b	0101001
c	0001011
d	0001110
-	0001100
$	0011000
.	1010100
/	1010001
:	1000101
+	0010101

Digit 1, 3, 5 and 7 - black bar, 0 represents slim bar, and 1 represents thick bar.

Digit 2, 4 and 6 - white space, 0 represents slim space, and 1 represents thick space.

Appendix 3: Parameters of user defined web actions

Except for Filter, Sort, Parameter, Property and SendMessage, all the other web actions do not have a web action builder. These web actions can be applied directly or with some parameter values.

The table below lists the usage of these web actions and the parameters each web action requires:

Web Action	Parameter	Description
user_atuoRefreshALLDataSet		Refreshes the report data automatically at certain interval.
	time	Specifies the interval at which JReport will automatically refresh the report data. Data type: Integer (positive)
user_bottomN		Shows records with the field value equal to one of the last N field values in a component.
	instanceName	Specifies the instance name of the component. Since instance name is hidden from users, the display name can be used as the value here. However, the value should follow certain format. Data type: String Value Format: &DisplayName
	columnMappingName	Specifies the DBField name based on query. Data type: String Value Format: MappingName
	value	Specifies the value for N. Data type: Integer (positive)
user_changeComboChartType		Changes combo chart type.
	instanceName	Specifies the instance name of the chart. Since instance name is hidden from users, the display name can be used as the value here. However, the value should follow certain format. Data type: String Value Format: &DisplayName
	isPrimaryAxis	Specifies whether the axis displaying the type is the primary axis or not. Data type: Boolean
	index	Specifies the index of the type. Data type: Integer Value: An integer equal to or larger than 0 and equal to or less than the value after the total number of all chart types minus 1.
	type	Specifies the chart type to change the original type to. Data type: String Value: You can choose among these 2-D types: bar/bench, line, area, and stock. Valid values should use the chart type names listed right after the sample images in the Chart types document.
user_changeCompProperty		Changes a specific property of a component.
	instanceName	Specifies the instance name of the component. Since instance name is hidden from users, the display name can be used as the value here. However, the value should follow certain format. Data type: String Value Format: &DisplayName
	propertyName	Specifies the property name.
	value	Specifies the property value.
	isBy	Specifies whether the property value is a formula or real value. Data type: Boolean
user_changeGenericChartType		Changes single chart type.
	instanceName	Specifies the instance name of the chart. Since instance name is hidden from users, the display name can be used as the value here. However, the value should follow certain format. Data type: String Value Format: &DisplayName
	isPrimaryAxis	Specifies whether the axis displaying the type is the primary axis or not. Data type: Boolean
	type	Specifies the chart type to change the original type to. Data type: String Value: Valid values should use the chart type names listed right after the sample images in the Chart types document.
user_changeStyleTo		Changes the report style.
	styleName	Specifies the name of a CSS or XSD style, for example, CSS style name: Classic, XSD style name: Classic*.
user_chartToCrosstab		Shows the To Crosstab dialog for converting a chart to a crosstab.
	instanceName	Specifies the instance name of the chart. Since instance name is hidden from users, the display name can be used as the value here. However, the value should follow certain format. Data type: String Value Format: &DisplayName
user_closeReportSet	-	Closes the report.
user_closeRpt	-	Closes the report tab.
user_crosstabToChart		Shows the To Chart dialog for converting a crosstab to a chart.
	instanceName	Specifies the instance name of the crosstab. Since instance name is hidden from users, the display name can be used as the value here. However, the value should follow certain format. Data type: String Value Format: &DisplayName
user_deleteRpt	-	Deletes the report.
user_downloadReport		Downloads report result in the specified format. Its usage is similar to user_oneStepExport.
	type	Represents the format. Data type: Integer Value: 0 for HTML; 2 for PDF; 3 for Postscript; 4 for RTF; 5 for TEXT; 6 for Excel; and 7 for XML.
	optionValues	Specifies the downloading options. Data type: String array Value: The value format is: *['Export_Option_Key=Value','Export_Option_Key=Value',...]*
user_drillTo		Drills to a specific dimension on a component.
	instanceName	Specifies the instance name of the component. Since instance name is hidden from users, the display name can be used as the value here. However, the value should follow certain format. Data type: String Value Format: &DisplayName
	columnBLName	Specifies the qualified name of the group based on business view. Data type: String Value Format: BusinessViewTreeRootNode.ParentNode. ViewElementName There can be no parent node or 1 or more levels of parent nodes.
user_exit	-	Closes the report.
user_filter		Filters the data in a component on a field.
	instanceName	Specifies the instance name of the component. Since instance name is hidden from users, the display name can be used as the value here. However, the value should follow certain format. Data type: String Value Format: &DisplayName
	columnMappingName	Specifies the DBField name based on query. Data type: String Value Format: MappingName
	operator	Specifies the operator for the filter criterion. Data type: String =: "eq" >: "gt" >=: "geq" <: "lt" <=: "leq" !=: "neq" IN: "in" NOT IN: "notin" LIKE: "like" NOT LIKE: "notlike" BETWEEN:"between" NOT BETWEEN:"notbetween" IS NULL:"isnull" IS NOT NULL:"isnotnull"
	value	Specifies the operand. Data type: String
	logic	Specifies the logic relationship between two neighboring filter criterions. The logic can be "AND" or "OR". Data type: String
	valueFormat	Specifies the format for the value. Usually this parameter is specified only for fields of Date/Time type. If it is not specified, the default format for the value will be accepted. Data type: String
user_firstPage	-	Turns to the first page.
user_groupFlex		Expands or collapses a specific group.
	groupControlObj	Specifies the group to be expanded or collapsed. Data type: HTML object. You can use "this" to represent the current group.
user_lastPage	-	Turns to the last page.
user_nextPage	-	Turns to the next page.
user_oneStepExport		Exports report result to a specified format.
	type	Represents the format. Data type: Integer Value: 0 for HTML; 2 for PDF; 3 for Postscript; 4 for RTF; 5 for TEXT; 6 for Excel; and 7 for XML.
	optionValues	Specifies the export options and their values. Data type: String array Value: The value format is: *['Export_Option_Key=Value','Export_Option_Key=Value',...]*
user_oneStepFilter		Filters the data in a component on multiple fields.
	columnMappingNames	Specifies the DBField names based on query. Data type: String array Value Format: [MappingName1,MappingName2, MappingName3,...]
	operators	Specifies the operators for the filter criteria. Data type: String array Available values are the same as those for operator of user_filter.
	values	Specifies the operands. Data type: String array
	logics	Specifies the logic relationships between two neighboring filter criterions. The logic can be "AND" or "OR". Data type: String array
	instanceName	Specifies the instance name of the component. Since instance name is hidden from users, the display name can be used as the value here. However, the value should follow certain format. Data type: String Value Format: &DisplayName
user_oneStepSearch		Locates a string in the report.
	value	Specifies the string to find. Data type: String
	isContent	Specifies whether to search the whole report content or a specified field. Data type: Boolean
	columnMappingName	Specifies the DBField name based on query. Data type: String Value Format: MappingName
	isUp	Specifies whether or not to search from the bottom to the top. Data type: Boolean
	isMatchCase	Specifies whether or not to match case when searching. Data type: Boolean
	isWordOnly	Specifies whether or not to match whole word. Data type: Boolean
user_oneStepSort		Sorts the data in a component on multiple fields.
	columnMappingNames	Specifies the DBField names based on query. Data type: String array Value Format: [MappingName1,MappingName2, MappingName3,...]
	orders	Specifies the sorting orders. Data type: Boolean array Value Format: For example: [true,false,false,...] (true for ascending, false for descending).
	instanceName	Specifies the instance name of the component. Since instance name is hidden from users, the display name can be used as the value here. However, the value should follow certain format. Data type: String Value Format: &DisplayName
user_openNewReport		Opens a report.
	catalog	Specifies the catalog of the report. Data type: String Value: You should use the catalog file path in the server resource tree. /USERFOLDERPATH/admin/ stands for the My Reports folder, and / stands for the Public Reports folder.
	rptName	Specifies the report name. Data type: String Value: You should use the report file path in the server resource tree. /USERFOLDERPATH/admin/ stands for the My Reports folder, and / stands for the Public Reports folder.
	newWindow	Specifies whether or not to open a new browser window for the report. Data type: Boolean
	params	Specifies parameter names and values for opening the report. Data type: 2-dimensional string array Value Format: [[[name1],[value1]],[[name2],[value2]],...]
user_prevPage	-	Turns to the previous page.
user_popCustomizedPage()	-	Pops up the Customized Page dialog, which by default displays all the parameters used in the current report if it contains parameters.
user_redo	-	Redoes an action.
user_refresh	-	Refreshes the report.
user_removeFilter		Removes the filters on the specified field or all the fields in a component.
	instanceName	Specifies the instance name of the component. Since instance name is hidden from users, the display name can be used as the value here. However, the value should follow certain format. Data type: String Value Format: &DisplayName
	columnMappingName	Specifies the DBField name based on query. If specified, the filters on this field in the component will be removed; if not specified, filters on all the fields of the component will be removed. Data type: String Value Format: MappingName
user_renameRpt	-	Shows the Rename Report dialog.
user_reset		Resets the report.
	instanceName	Specifies the instance name of the component. Since instance name is hidden from users, the display name can be used as the value here. However, the value should follow certain format. Data type: String Value Format: &DisplayName
user_rotateCrosstab		Rotates a crosstab.
	instanceName	Specifies the instance name of the crosstab. Since instance name is hidden from users, the display name can be used as the value here. However, the value should follow certain format. Data type: String Value Format: &DisplayName
user_saveRpt	-	Saves the report.
user_showDHTMLView	-	Shows the Resource View panel.
user_showFilterDialog	-	Shows the Filter dialog.
user_showHelp		Shows the help about a specific dialog, or shows the index page of the Page Report Studio help documentation if null.
	helpId	ID of the specific Page Report Studio dialog help file. Data type: Five-digit integer starting with 4, for example, 40034. You can refer to the file Hyperlink.txt in the JReport Designer installation root to check for the help files that are linked with IDs starting with 4.
user_showNewRptDialog	-	Shows the New Report dialog.
user_showOpenRptDialog	-	Shows the Open Report dialog.
user_showPageSetupDialog	-	Shows the Page Properties dialog.
user_showPrintDialog	-	Shows the Printable Version dialog.
user_showSaveAsDialog	-	Shows the Save As dialog.
user_showSaveResultDialog	-	Shows the Export dialog.
user_showSearchDialog	-	Shows the Search dialog.
user_showSortDialog	-	Shows the Sort dialog.
user_showTOC	-	Shows the TOC Browser.
user_showToolbox	-	Shows the Toolbox.
user_showUserPanel	-	Shows the User Information bar.
user_sort		Sorts the data in a component on a specific field.
	instanceName	Specifies the instance name of the component. Since instance name is hidden from users, the display name can be used as the value here. However, the value should follow certain format. Data type: String Value Format: &DisplayName
	columnMappingName	Specifies the DBField name based on query. Data type: String Value Format: MappingName
	order	Specifies the sorting order. Data type: Boolean Value: true for ascending, and false for descending.
user_stopAutoRefreshDataSet	-	Stops JReport from refreshing the report data automatically.
user_topN		Shows records with the field value equal to one of the top N field values in a component.
	instanceName	Specifies the instance name of the component. Since instance name is hidden from users, the display name can be used as the value here. However, the value should follow certain format. Data type: String Value Format: &DisplayName
	columnMappingName	Specifies the DBField name based on query. Data type: String Value Format: MappingName
	value	Specifies the value for N. Data type: Integer (positive)
user_undo	-	Undoes an action.
user_zoomTo		Zooms the report in or out to a specific magnification.
	zoomSize	The magnification ratio in percent. It should be an integer between 0 and 400.

Note: If you are very familiar with usages of the web actions, you can select the first item Input an Action, and then input the action name with parameter (if necessary).

Component property names

The following list is for your reference when specifying the value of the parameter propertyName applied in the web action user_changeCompProperty. Note that the property names here can be recognized by classes and are different from the names provided in the Report Inspector, which are just display names of component properties. However, you can figure out the corresponding display names according to the property names.

For details about the descriptions of the component properties and their values, see Properties in the Report Inspector where display names are used.

<table>
<tr><th>Component</th><th>Property Name</th></tr>
<tr><td>ArcShape</td><td>[TopAttachPosX, TopAttachPosY, BottomAttachPosX, BottomAttachPosY]||[Background]||[Style, class]||[ExportToDHTML, Invisible, SuppressedIfNoRecords, RecordLocation]||[StartAngle, ArcAngle, BorderColor, BorderStyle, BorderWidth]||[ExternalCSSClassValue, ExternalID, ExternalStyle, Lang, ExternalTitle]</td></tr>
<tr><td>Area</td><td>[Name, Alt, DetailReport, Link, FillColor]||[Type, Coordinates]||[BoundaryWidth, BoundaryColor, BoundaryStyle]||[ExternalCSSClassValue, ExternalAccessKey, HrefLang, ExternalTabIndex, ExternalTitle]||[UseDefault]</td></tr>
<tr><td>AreaDBText</td><td>[X, Y, Width, Height]||[Background, Foreground]||[padding-top, padding-bottom, padding-left, padding-right]||[BorderWidth, TopLine, BottomLine, LeftLine, RightLine, Shadow, BorderColor, ShadowColor]||[PatternColor, PatternStyle]||[FontFace, FontSize, Bold, Italic, StrikeOut, Underline, Format, AutoFit, MaxWidth, WordWrap, VerticalAlignment, HorizontalAlignment]||[Invisible, DataMappingFile, DisplayNull, ColumnName, CacheValue, AutoConnector]||[ExternalCSSClassValue, ExternalID, ExternalStyle, Lang, ExternalAccessKey, ExternalDir, ExternalTabIndex, ExternalTitle, LongDesc, HrefLang]||[UseDefault]</td></tr>
<tr><td>AreaText</td><td>[Text, FontFace, FontSize, Bold, Italic, StrikeOut, Underline, AutoFit, MaxWidth, WordWrap, VerticalAlignment, HorizontalAlignment]||[X, Y, Width, Height]||[Background, Foreground]||[padding-top, padding-bottom, padding-left, padding-right]||[BorderWidth, TopLine, BottomLine, LeftLine, RightLine, Shadow, BorderColor, ShadowColor]||[PatternColor, PatternStyle]||[Invisible, AutoConnector]||[ExternalCSSClassValue, ExternalID, ExternalStyle, Lang, ExternalAccessKey, ExternalDir, ExternalTabIndex, ExternalTitle, LongDesc, HrefLang]||[UseDefault]</td></tr>
<tr><td>BandedColumnInfos</td><td>[Background]||[Style, class]||[position, ExportToDHTML, SuppressedIfNoRecords, RecordLocation, ExportToCSV, ExportToXLS, Pattern, ShowColumns, ColumnWidths]||[BorderWidth, TopLine, BottomLine, LeftLine, RightLine, BorderColor, ShowBorder]||[ExternalCSSClassValue, ExternalID, Lang]</td></tr>
<tr><td>BandedFooter</td><td>[Width, Height]||[Background]||[Style, class]||[ExportToDHTML, Invisible, Suppressed, SuppressedIfNoRecords, RecordLocation, ExportToCSV, ExportToXLS, Label, OnNewPage, Underlay, CanCrossPage, SuppressBlankSection, FillWholePage, ShowBottomLine, ToBottom]||[BorderWidth, TopLine, BottomLine, LeftLine, RightLine, BorderColor]||[ExternalCSSClassValue, ExternalID, Lang]</td></tr>
<tr><td>BandedHeader</td><td>[Width, Height]||[Background]||[Style, class]||[ExportToDHTML, Invisible, Suppressed, SuppressedIfNoRecords, RecordLocation, ExportToCSV, ExportToXLS, MergeToNextPanel, Label, OnNewPage, Underlay, CanCrossPage, SuppressBlankSection, FillWholePage, ShowBottomLine]||[BorderWidth, TopLine, BottomLine, LeftLine, RightLine, BorderColor]||[ExternalCSSClassValue, ExternalID, Lang]</td></tr>
<tr><td>BandedObject</td><td>[X, Y, Width, Height]||[Background]||[Style, class]||[position, ExportToDHTML, Invisible, Suppressed, SuppressedIfNoRecords, RecordLocation, ExportToCSV, ExportToXLS, Cache, SuppressEmpty]||[ColumnIndex, RowIndex]||[BorderWidth, TopLine, BottomLine, LeftLine, RightLine, BorderColor, BorderJoint, RoundJointRadius]||[TOCAnchor, AnchorDispValue]||[ExternalCSSClassValue, ExternalID, Lang]</td></tr>
<tr><td>BandedPageFooter</td><td>[Width, Height]||[Background]||[Style, class]||[ExportToDHTML, Invisible, Suppressed, SuppressedIfNoRecords, RecordLocation, ExportToCSV, ExportToXLS, Underlay, SuppressBlankSection, ShowBottomLine, SuppressInLastPage]||[BorderWidth, TopLine, BottomLine, LeftLine, RightLine, BorderColor]||[ExternalCSSClassValue, ExternalID, ExternalStyle, Lang]</td></tr>
<tr><td>BandedPageHeader</td><td>[Width, Height]||[Background]||[Style, class]||[ExportToDHTML, Invisible, Suppressed, SuppressedIfNoRecords, RecordLocation, ExportToCSV, ExportToXLS, MergeToNextPanel, RepeatInDetailPanel, RemoveBlankRow, Label, Underlay, SuppressBlankSection, ShowBottomLine]||[BorderWidth, TopLine, BottomLine, LeftLine, RightLine, BorderColor]||[ExternalCSSClassValue, ExternalID, Lang]</td></tr>
<tr><td>BandedPagePanel</td><td>[PageType, Orientation, Width, Height]||[TopMargin, LeftMargin, BottomMargin, RightMargin]</td></tr>
<tr><td>Box</td><td>[TopAttachPosX, TopAttachPosY, BottomAttachPosX, BottomAttachPosY]||[Background]||[Style, class]||[ExportToDHTML, Invisible, SuppressedIfNoRecords, RecordLocation, ExportToXLS]||[TopAttachCol, TopAttachRow, BottomAttachCol, BottomAttachRow]||[BorderColor, BorderStyle, BorderWidth]||[ExternalCSSClassValue, ExternalID, ExternalStyle, Lang, ExternalTitle]</td></tr>
<tr><td>ChartLabelObject</td><td>[X, Y, Width, Height]||[Style, class]||[ExternalCSSClassValue]||[Text, LabelTextFontName, LabelTextFontSize, LabelTextFontStyle, LabelTextRotation, LabelTextShearing, LabelTextUnderline, LabelTextStrikethrough, LabelTextSpecialEffect1, LabelTextSpecialEffect2, LabelTextWordWrap, LabelTextPenStyle, LabelTextPenColor, LabelTextPenTransparency, LabelTextPenThickness, LabelTextPenEndCaps, LabelTextPenLineJoint, LabelTextPenOutlined, LabelTextPenVariabledDash, LabelTextPatternType, LabelTextPatternTransparency, LabelTextPatternColor, LabelTextPatternTextureStyle, LabelTextPatternTextureForeColor, LabelTextPatternTextureBackColor, LabelTextPatternGradientStyle, LabelTextPatternGradientStartColor, LabelTextPatternGradientEndColor, LabelTextPatternGradientStartX, LabelTextPatternGradientStartY, LabelTextPatternGradientEndX, LabelTextPatternGradientEndY, TextAlignment, TopMargin, LeftMargin, RightMargin, BottomMargin]||[LabelBackgroundBorderStyle, LabelBackgroundBorderPenStyle, LabelBackgroundBorderPenColor, LabelBackgroundBorderPenTransparency, LabelBackgroundBorderPenThickness, LabelBackgroundBorderPenEndCaps, LabelBackgroundBorderPenLineJoint, LabelBackgroundBorderPenOutlined, LabelBackgroundBorderPenVariabledDash, LabelBackgroundFillPatternType, LabelBackgroundFillPatternTransparency, LabelBackgroundFillPatternColor, LabelBackgroundFillPatternTextureStyle, LabelBackgroundFillPatternTextureForeColor, LabelBackgroundFillPatternTextureBackColor, LabelBackgroundFillPatternGradientStyle, LabelBackgroundFillPatternGradientStartColor, LabelBackgroundFillPatternGradientEndColor, LabelBackgroundFillPatternGradientStartX, LabelBackgroundFillPatternGradientStartY, LabelBackgroundFillPatternGradientEndX, LabelBackgroundFillPatternGradientEndY, LabelBackgroundFillPatternImageFile, LabelBackgroundFillPatternImageLayout, LabelBackgroundFillPatternImageX, LabelBackgroundFillPatternImageY, LabelBackgroundFillPatternImageWidth, LabelBackgroundFillPatternImageHeight, LabelBackgroundTop, LabelBackgroundLeft, LabelBackgroundRight, LabelBackgroundBottom]||[IconStyle, IconWidth, IconHeight, IconAlignment, IconBorderPenStyle, IconBorderPenColor, IconBorderPenTransparency, IconBorderPenThickness, IconBorderPenEndCaps, IconBorderPenLineJoint, IconBorderPenOutlined, IconBorderPenVariabledDash, IconFillPatternType, IconFillPatternTransparency, IconFillPatternColor, IconFillPatternTextureStyle, IconFillPatternTextureForeColor, IconFillPatternTextureBackColor, IconFillPatternGradientStyle, IconFillPatternGradientStartColor, IconFillPatternGradientEndColor, IconFillPatternGradientStartX, IconFillPatternGradientStartY, IconFillPatternGradientEndX, IconFillPatternGradientEndY, IconFillPatternImageFile, IconFillPatternImageLayout, IconFillPatternImageX, IconFillPatternImageY, IconFillPatternImageWidth, IconFillPatternImageHeight, IconTextGap]</td></tr>
<tr><td>ChartLegendObject</td><td>[X, Y, Width, Height]||[Style, class]||[ExternalCSSClassValue]||[AutoSize, Placement, SubPlacement, LabelFilterSource, LegendFilter, LabelVSpacing, LabelHSpacing, TopMargin, LeftMargin, RightMargin, BottomMargin, ShowTips]||[LegendBackgroundBorderStyle, LegendBackgroundBorderPenStyle, LegendBackgroundBorderPenColor, LegendBackgroundBorderPenTransparency, LegendBackgroundBorderPenThickness, LegendBackgroundBorderPenEndCaps, LegendBackgroundBorderPenLineJoint, LegendBackgroundBorderPenOutlined, LegendBackgroundBorderPenVariabledDash, LegendBackgroundFillPatternType, LegendBackgroundFillPatternTransparency, LegendBackgroundFillPatternColor, LegendBackgroundFillPatternTextureStyle, LegendBackgroundFillPatternTextureForeColor, LegendBackgroundFillPatternTextureBackColor, LegendBackgroundFillPatternGradientStyle, LegendBackgroundFillPatternGradientStartColor, LegendBackgroundFillPatternGradientEndColor, LegendBackgroundFillPatternGradientStartX, LegendBackgroundFillPatternGradientStartY, LegendBackgroundFillPatternGradientEndX, LegendBackgroundFillPatternGradientEndY, LegendBackgroundFillPatternImageFile, LegendBackgroundFillPatternImageLayout, LegendBackgroundFillPatternImageX, LegendBackgroundFillPatternImageY, LegendBackgroundFillPatternImageWidth, LegendBackgroundFillPatternImageHeight, LegendBackgroundTop, LegendBackgroundLeft, LegendBackgroundRight, LegendBackgroundBottom]||[ReverseLabels, LabelTextFontName, LabelTextFontSize, LabelTextFontStyle, LabelTextRotation, LabelTextShearing, LabelTextUnderline, LabelTextStrikethrough, LabelTextSpecialEffect1, LabelTextSpecialEffect2, LabelTextWordWrap, LabelTextPenStyle, LabelTextPenColor, LabelTextPenTransparency, LabelTextPenThickness, LabelTextPenEndCaps, LabelTextPenLineJoint, LabelTextPenOutlined, LabelTextPenVariabledDash, LabelTextPatternType, LabelTextPatternTransparency, LabelTextPatternColor, LabelTextPatternTextureStyle, LabelTextPatternTextureForeColor, LabelTextPatternTextureBackColor, LabelTextPatternGradientStyle, LabelTextPatternGradientStartColor, LabelTextPatternGradientEndColor, LabelTextPatternGradientStartX, LabelTextPatternGradientStartY, LabelTextPatternGradientEndX, LabelTextPatternGradientEndY, ShowValues, ValueFormat, ShowPercent, PercentFormat]||[IconWidth, IconHeight, IconAlignment, IconBorderPenStyle, IconBorderPenColor, IconBorderPenTransparency, IconBorderPenThickness, IconBorderPenEndCaps, IconBorderPenLineJoint, IconBorderPenOutlined, IconBorderPenVariabledDash, IconTextGap, MarkNodeUsage]</td></tr>
<tr><td>ChartObject</td><td>[X, Y, Width, Height]||[Style, class]||[position, ExportToDHTML, Invisible, Suppressed, SuppressedIfNoRecords, ExportToCSV, ExportToXLS, Cache, Alt, LogicColumn]||[ColumnIndex, RowIndex, ColumnNumber, RowNumber]||[TOCAnchor, AnchorDispValue]||[ShowLegend, AntiAliasing, PatternList]||[PlatformBackgroundBorderStyle, PlatformBackgroundBorderPenStyle, PlatformBackgroundBorderPenColor, PlatformBackgroundBorderPenTransparency, PlatformBackgroundBorderPenThickness, PlatformBackgroundBorderPenEndCaps, PlatformBackgroundBorderPenLineJoint, PlatformBackgroundBorderPenOutlined, PlatformBackgroundBorderPenLineJointRadius, PlatformBackgroundBorderPenVariabledDash, PlatformBackgroundFillPatternType, PlatformBackgroundFillPatternTransparency, PlatformBackgroundFillPatternColor, PlatformBackgroundFillPatternTextureStyle, PlatformBackgroundFillPatternTextureForeColor, PlatformBackgroundFillPatternTextureBackColor, PlatformBackgroundFillPatternGradientStyle, PlatformBackgroundFillPatternGradientStartColor, PlatformBackgroundFillPatternGradientEndColor, PlatformBackgroundFillPatternGradientStartX, PlatformBackgroundFillPatternGradientStartY, PlatformBackgroundFillPatternGradientEndX, PlatformBackgroundFillPatternGradientEndY, PlatformBackgroundFillPatternImageFile, PlatformBackgroundFillPatternImageLayout, PlatformBackgroundFillPatternImageX, PlatformBackgroundFillPatternImageY, PlatformBackgroundFillPatternImageWidth, PlatformBackgroundFillPatternImageHeight, PlatformBackgroundTop, PlatformBackgroundLeft, PlatformBackgroundRight, PlatformBackgroundBottom]||[SortCategory, SortSeries, ReverseCategory, ReverseSeries, CategoryStartOffset, CategoryEndOffset, SeriesStartOffset, SeriesEndOffset, CategoryFilter, SeriesFilter, PrimaryDataFilter, SecondaryDataFilter, HintFilter, GotoDetail, HyperLink, XHyperLink, ZHyperLink, GroupValueFormat, SeriesValueEncoding, SwapGroup, HyperLinkTarget, XHyperLinkTarget, ZHyperLinkTarget, HintPercentFilter]||[ExternalCSSClassValue, LongDesc, ExternalTitle, ExternalID]</td></tr>
<tr><td>ChartPaperObject</td><td>[X, Y, Width, Height]||[Background]||[Style, class]||[position, ExportToDHTML, Invisible, Suppressed, SuppressedIfNoRecords, RecordLocation, ExportToCSV, ExportToXLS]||[ColumnIndex, RowIndex]||[BorderWidth, TopLine, BottomLine, LeftLine, RightLine, BorderColor]||[ExternalCSSClassValue]</td></tr>
<tr><td>ColAttribute</td><td>[ResourceName, DisplayName, Sort, Filter, Drill, Search]</td></tr>
<tr><td>Column</td><td>[Height]||[Background]||[Style, class]||[position, ExportToDHTML, Suppressed, SuppressedIfNoRecords, RecordLocation, ExportToCSV, ExportToXLS]||[ShowBorder]||[ExternalCSSClassValue, ExternalID, ExternalStyle, Lang, ExternalTitle]</td></tr>
<tr><td>CTAggField</td><td>[X, Y, Width, Height]||[Background, Foreground]||[Style, id, class]||[padding-top, padding-bottom, padding-left, padding-right]||[BorderWidth, TopLine, BottomLine, LeftLine, RightLine, Shadow, BorderColor, ShadowColor]||[PatternColor, PatternStyle]||[FontFace, FontSize, Bold, Italic, StrikeOut, Underline, Format, AutoFit, MaxWidth, WordWrap, TransWhileToHtml, VerticalAlignment, HorizontalAlignment]||[ExportToDHTML, Suppressed, SuppressedIfNoRecords, RecordLocation, ExportToCSV, ExportToXLS, Link, EnableHyperlinkOnPDF, EnableHyperlinkOnExcel, EnableHyperlinkOnHTML, TransferStyle, GotoDetail, DetailTargetFrame, ExportToPDF, FilterOptions, DataMappingFile, SuppressedIfNull, DisplayNull, ColumnName, ValueDelimiter, CacheValue]||[ExternalCSSClassValue, ExternalID, ExternalStyle, Lang, ExternalAccessKey, ExternalDir, ExternalTabIndex, ExternalTitle, LongDesc, HrefLang, Abbr, Axis, Headers, Scope]</td></tr>
<tr><td>CTAggSpecialField</td><td>[X, Y, Width, Height]||[Background, Foreground]||[Style, id, class]||[padding-top, padding-bottom, padding-left, padding-right]||[BorderWidth, TopLine, BottomLine, LeftLine, RightLine, Shadow, BorderColor, ShadowColor]||[PatternColor, PatternStyle]||[FontFace, FontSize, Bold, Italic, StrikeOut, Underline, Format, AutoFit, MaxWidth, WordWrap, TransWhileToHtml, VerticalAlignment, HorizontalAlignment]||[ExportToDHTML, Suppressed, SuppressedIfNoRecords, RecordLocation, ExportToCSV, ExportToXLS, Link, EnableHyperlinkOnPDF, EnableHyperlinkOnExcel, EnableHyperlinkOnHTML, TransferStyle, GotoDetail, DetailTargetFrame, ExportToPDF, FilterOptions, DataMappingFile, SuppressedIfNull, DisplayNull, ColumnName, ValueDelimiter, CacheValue]||[ExternalCSSClassValue, ExternalID, ExternalStyle, Lang, ExternalAccessKey, ExternalDir, ExternalTabIndex, ExternalTitle, LongDesc, HrefLang, Abbr, Axis, Headers, Scope]</td></tr>
<tr><td>CTCrossTab</td><td>[X, Y, Width, Height]||[Background]||[EnableAlternativeLineColor, AlternativeLineColorDirection, AlternativeLineColorPatternList]||[Style, class]||[position, ExportToDHTML, Invisible, Suppressed, SuppressedIfNoRecords, RecordLocation, ExportToCSV, ExportToXLS, currentRowBlockIndex, currentColBlockIndex, itemsPerRowBlock, itemsPerColBlock, Cache, SuppressEmpty, CDNUsed, CDNHeight, CDNWidth]||[ColumnIndex, RowIndex]||[BorderColor, Hasborder]||[HorizontalGap, VerticalGap, VerticalLayout, BoundaryValue, RowTotalOnTop, ColumnTotalOnLeft, RepeatRowHeader, AvoidOrphanHeader, BlockGap, SuppressRowHeader, SuppressColumnHeader, isTableStyle, repeatAgg, canExpandData, isOutSideAggTitle]||[TOCAnchor, AnchorDispValue]||[ExternalCSSClassValue, ExternalID, ExternalStyle, Lang, Summary]</td></tr>
<tr><td>CTDBField</td><td>[Background, Foreground]||[Style, id, class]||[padding-top, padding-bottom, padding-left, padding-right]||[BorderWidth, TopLine, BottomLine, LeftLine, RightLine, Shadow, BorderColor, ShadowColor]||[PatternColor, PatternStyle]||[FontFace, FontSize, Bold, Italic, StrikeOut, Underline, Format, AutoFit, MaxWidth, WordWrap, TransWhileToHtml, VerticalAlignment, HorizontalAlignment]||[ExportToDHTML, Suppressed, SuppressedIfNoRecords, RecordLocation, ExportToCSV, ExportToXLS, Link, EnableHyperlinkOnPDF, EnableHyperlinkOnExcel, EnableHyperlinkOnHTML, TransferStyle, GotoDetail, DetailTargetFrame, ExportToPDF, FilterOptions, DataMappingFile, SuppressedIfNull, DisplayNull, ColumnName, ValueDelimiter, CacheValue, Orderby, isExpandData, ExpandData]||[ExternalCSSClassValue, ExternalID, ExternalStyle, Lang, ExternalAccessKey, ExternalDir, ExternalTabIndex, ExternalTitle, LongDesc, HrefLang, Abbr, Axis, Headers, Scope]</td></tr>
<tr><td>CTDBTitleField</td><td>[Text, FontFace, FontSize, Bold, Italic, StrikeOut, Underline, MaxWidth, WordWrap, TransWhileToHtml, VerticalAlignment, HorizontalAlignment]||[Background, Foreground]||[Style, id, class]||[padding-top, padding-bottom, padding-left, padding-right]||[BorderWidth, TopLine, BottomLine, LeftLine, RightLine, Shadow, BorderColor, ShadowColor]||[PatternColor, PatternStyle]||[ExportToDHTML, Invisible, Suppressed, SuppressedIfNoRecords, RecordLocation, ExportToCSV, ExportToXLS, Link, EnableHyperlinkOnPDF, EnableHyperlinkOnExcel, EnableHyperlinkOnHTML, TransferStyle, GotoDetail, DetailTargetFrame, ExportToPDF, FilterOptions, Sortable, Filterable, BindColumn, SuppressAggreate]||[ExternalCSSClassValue, ExternalID, ExternalStyle, Lang, ExternalAccessKey, ExternalDir, ExternalTabIndex, ExternalTitle, LongDesc, HrefLang, Abbr, Axis, Headers, Scope]</td></tr>
<tr><td>CTLabelField</td><td>[Text, FontFace, FontSize, Bold, Italic, StrikeOut, Underline, MaxWidth, WordWrap, TransWhileToHtml, VerticalAlignment, HorizontalAlignment]||[Background, Foreground]||[Style, id, class]||[padding-top, padding-bottom, padding-left, padding-right]||[BorderWidth, TopLine, BottomLine, LeftLine, RightLine, Shadow, BorderColor, ShadowColor]||[PatternColor, PatternStyle]||[ExportToDHTML, Invisible, Suppressed, SuppressedIfNoRecords, RecordLocation, ExportToCSV, ExportToXLS, Link, EnableHyperlinkOnPDF, EnableHyperlinkOnExcel, EnableHyperlinkOnHTML, TransferStyle, GotoDetail, DetailTargetFrame, ExportToPDF, FilterOptions, Sortable, Filterable, BindColumn]||[ExternalCSSClassValue, ExternalID, ExternalStyle, Lang, ExternalAccessKey, ExternalDir, ExternalTabIndex, ExternalTitle, LongDesc, HrefLang, Abbr, Axis, Headers, Scope]</td></tr>
<tr><td>CTTextField</td><td>[Text, FontFace, FontSize, Bold, Italic, StrikeOut, Underline, MaxWidth, WordWrap, TransWhileToHtml, VerticalAlignment, HorizontalAlignment]||[Background, Foreground]||[Style, id, class]||[padding-top, padding-bottom, padding-left, padding-right]||[BorderWidth, TopLine, BottomLine, LeftLine, RightLine, Shadow, BorderColor, ShadowColor]||[PatternColor, PatternStyle]||[ExportToDHTML, Invisible, Suppressed, SuppressedIfNoRecords, RecordLocation, ExportToCSV, ExportToXLS, Link, EnableHyperlinkOnPDF, EnableHyperlinkOnExcel, EnableHyperlinkOnHTML, TransferStyle, GotoDetail, DetailTargetFrame, ExportToPDF, FilterOptions, Sortable, Filterable, BindColumn, SuppressAggreate]||[ExternalCSSClassValue, ExternalID, ExternalStyle, Lang, ExternalAccessKey, ExternalDir, ExternalTabIndex, ExternalTitle, LongDesc, HrefLang, Abbr, Axis, Headers, Scope]</td></tr>
<tr><td>DBField</td><td>[X, Y, Width, Height]||[Background, Foreground]||[Style, id, class]||[ColumnIndex, RowIndex]||[padding-top, padding-bottom, padding-left, padding-right]||[BorderWidth, TopLine, BottomLine, LeftLine, RightLine, Shadow, BorderColor, ShadowColor]||[PatternColor, PatternStyle]||[FontFace, FontSize, Bold, Italic, StrikeOut, Underline, Format, AutoFit, MaxWidth, WordWrap, TransWhileToHtml, VerticalAlignment, HorizontalAlignment]||[position, ExportToDHTML, Invisible, Suppressed, SuppressedIfNoRecords, RecordLocation, ExportToCSV, ExportToXLS, Link, EnableHyperlinkOnPDF, EnableHyperlinkOnExcel, EnableHyperlinkOnHTML, DetailReport, TransferStyle, GotoDetail, DetailTargetFrame, ExportToPDF, FilterOptions, DataMappingFile, SuppressedIfNull, DisplayNull, ColumnName, ValueDelimiter, CacheValue]||[TOCAnchor, AnchorDispValue]||[ExternalCSSClassValue, ExternalID, ExternalStyle, Lang, ExternalAccessKey, ExternalDir, ExternalTabIndex, ExternalTitle, LongDesc, HrefLang]</td></tr>
<tr><td>DetailPanel</td><td>[Width, Height]||[Background]||[Style, class]||[ExportToDHTML, Invisible, Suppressed, SuppressedIfNoRecords, RecordLocation, ExportToCSV, ExportToXLS, RemoveBlankRow, Label, OnNewPage, Underlay, CanCrossPage, SuppressBlankSection, FillWholePage, ShowBottomLine, TileDetailSection, TileHorizontal, CacheSection, currentBlockIndex, itemsPerBlock]||[BorderWidth, TopLine, BottomLine, LeftLine, RightLine, BorderColor]||[ExternalCSSClassValue, ExternalID, ExternalStyle, Lang]</td></tr>
<tr><td>ExtendsDataSet</td><td>[MaxRecords]||[SecurityName]||[RecordsPerPage, MaxPageNumber]||[dsName, queryName]</td></tr>
<tr><td>FilterControl</td><td>[X, Y, Width, Height]||[Background, Foreground]||[Style, class, id]||[position, ExportToDHTML, Invisible, ExportToCSV, ExportToXLS]||[ColumnIndex, RowIndex]||[BorderColor, BorderWidth, TopLine, BottomLine, LeftLine, RightLine]||[IsShowTitle, TitleText, TitleBold, TitleItalic, TitleFontFace, TitleFontSize, TitleForeground, TitleUnderline, TitleBackground, TitleMapName, TitleHorizontalAlignment]||[Bold, Italic, FontFace, FontSize, Underline, Format]||[ExternalCSSClassValue, ExternalID, ExternalStyle, Lang, ExternalTitle]</td></tr>
<tr><td>Form</td><td>[Style, class]||[LogicColumn]||[FormName, FormId, FormAction, Disabled, FormMethod, FormTarget]||[ExternalCSSClassValue, ExternalStyle, Lang, ExternalTitle]</td></tr>
<tr><td>GMarker</td><td>[ExportToDHTML, ExportToXLS, Invisible, Suppressed, SuppressedIfNoRecords]</td></tr>
<tr><td>GoogleMap</td><td>[X, Y, Width, Height]||[position, ExportToDHTML, Invisible, Suppressed, SuppressedIfNoRecords, ExportToXLS]||[ColumnIndex, RowIndex]||[TOCAnchor, AnchorDispValue]||[ExternalCSSClassValue, ExternalID, ExternalStyle, Lang]</td></tr>
<tr><td>GroupControlImg</td><td>[X, Y, Width, Height]||[Background]||[Style, class]||[position, ExportToDHTML, Invisible, Suppressed]||[ID, Name, Disabled, Title, TabIndex, Class, Dir, Size, Value, Checked, Accesskey, Alt, Src]||[onfocus, onblur, onchange, onclick, ondblclick, onmousedown, onmouseup, onmouseover, onmousemove, onmouseout, onkeypress, onkeydown, onkeyup]||[ExternalCSSClassValue, ExternalID, ExternalStyle, Lang]</td></tr>
<tr><td>GroupFilter</td><td>[X, Y, Width, Height]||[Background]||[Style, class]||[position, ExportToDHTML, Invisible, Suppressed, SuppressedIfNoRecords, RecordLocation, ExportToCSV, ExportToXLS]||[ColumnIndex, RowIndex]||[ExternalCSSClassValue, ExternalID, ExternalStyle, Lang]</td></tr>
<tr><td>GroupFooterPanel</td><td>[Width, Height]||[Background]||[Style, class]||[ExportToDHTML, Invisible, Suppressed, SuppressedIfNoRecords, RecordLocation, ExportToCSV, ExportToXLS, MergeToNextPanel, Label, OnNewPage, Underlay, CanCrossPage, SuppressBlankSection, FillWholePage, ShowBottomLine]||[BorderWidth, TopLine, BottomLine, LeftLine, RightLine, BorderColor]||[ExternalCSSClassValue, ExternalID, ExternalStyle, Lang]</td></tr>
<tr><td>GroupHeaderPanel</td><td>[Width, Height]||[Background]||[Style, class]||[ExportToDHTML, Invisible, Suppressed, SuppressedIfNoRecords, RecordLocation, ExportToCSV, ExportToXLS, MergeToNextPanel, RepeatInDetailPanel, RemoveBlankRow, Label, OnNewPage, Underlay, CanCrossPage, SuppressBlankSection, FillWholePage, ShowBottomLine, Repeat]||[BorderWidth, TopLine, BottomLine, LeftLine, RightLine, BorderColor]||[ExternalCSSClassValue, ExternalID, ExternalStyle, Lang]</td></tr>
<tr><td>GroupPanel</td><td>[isExpandData, ShrinkFooter, currentBlockIndex, itemsPerBlock]||[KeepGroupTogether, RepeatWhileGF]||[TOCAnchor, AnchorDispValue]||[Grant, Cascade, Groups, Roles]</td></tr>
<tr><td>HDSDataSet</td><td>[MaxRecords]||[SecurityName]||[RecordsPerPage, MaxPageNumber]||[dsName, queryName]</td></tr>
<tr><td>Image</td><td>[X, Y, Width, Height]||[Style, class, ID]||[position, ExportToDHTML, Invisible, Suppressed, SuppressedIfNoRecords, RecordLocation, ExportToCSV, ExportToXLS, GotoDetail, DetailTargetFrame, Link, TransferStyle, DetailReport, VerticalAlignment, HorizontalAlignment, LogicColumn]||[ColumnIndex, RowIndex, ColumnNumber, RowNumber]||[MaxScalingRatio, ScalingMode, Rotation, PictureName, Alt, Name, Value, Disabled, Title, TabIndex]||[onfocus, onblur, onclick, ondblclick, onmousedown, onmouseup, onmouseover, onmousemove, onmouseout, onkeypress, onkeydown, onkeyup]||[TOCAnchor, AnchorDispValue]||[ExternalCSSClassValue, ExternalID, ExternalStyle, Lang, ExternalAccessKey, HrefLang, LongDesc]</td></tr>
<tr><td>JHyperLink</td><td>[X, Y, Width, Height]||[Style, class]||[position, ExportToDHTML, Invisible, Suppressed, SuppressedIfNoRecords, RecordLocation, ExportToCSV, ExportToXLS, URL, ExecClassName, autoFit, target, toolTipText, displayValue, VerticalAlignment, HorizontalAlignment, wordWrap, EnableHyperlinkOnPDF, EnableHyperlinkOnExcel, EnableHyperlinkOnHTML, HtmlStyle]||[ColumnIndex, RowIndex, ColumnNumber, RowNumber]||[foreground, background]||[fontFace, fontSize, fontItalic, fontBold]||[displayImage, original, Alt]||[ExternalCSSClassValue, ExternalAccessKey, HrefLang]</td></tr>
<tr><td>JRotator</td><td>[X, Y, Width, Height]||[Background, Foreground]||[Style, class]||[position, ExportToDHTML, Invisible, Suppressed, SuppressedIfNoRecords, RecordLocation, ExportToCSV, ExportToXLS, Rotate, DisplayValue, DisplayImage, VerticalAlignment, HorizontalAlignment]||[ColumnIndex, RowIndex]||[FontFace, FontSize, FontItalic, FontBold]||[ExternalCSSClassValue]</td></tr>
<tr><td>Line</td><td>[TopAttachPosX, TopAttachPosY, BottomAttachPosX, BottomAttachPosY]||[Style, class]||[ExportToDHTML, Invisible, SuppressedIfNoRecords, RecordLocation, ExportToXLS]||[TopAttachCol, TopAttachRow, BottomAttachCol, BottomAttachRow]||[LineColor, LineStyle, LineWidth]||[ExternalCSSClassValue, ExternalID, ExternalStyle, Lang, ExternalTitle]</td></tr>
<tr><td>Map</td><td>[X, Y, Width, Height]||[Background]||[position, ExportToDHTML, Invisible, Suppressed, SuppressedIfNoRecords, ExportToXLS, GotoDetail, DetailTargetFrame]||[ColumnIndex, RowIndex]||[BorderWidth, TopLine, BottomLine, LeftLine, RightLine, BorderColor]||[Name, ImageSource, AltContentType, ColumnName]||[TOCAnchor, AnchorDispValue]||[ExternalCSSClassValue, ExternalID, ExternalStyle, Lang]</td></tr>
<tr><td>MapAddTextField</td><td>[Text, FontFace, FontSize, Bold, Italic, StrikeOut, Underline, AutoFit, MaxWidth, WordWrap, VerticalAlignment, HorizontalAlignment]||[X, Y, Width, Height]||[Background, Foreground]||[padding-top, padding-bottom, padding-left, padding-right]||[BorderWidth, TopLine, BottomLine, LeftLine, RightLine, Shadow, BorderColor, ShadowColor]||[PatternColor, PatternStyle]||[Invisible]||[ExternalCSSClassValue, ExternalID, ExternalStyle, Lang, ExternalAccessKey, ExternalDir, ExternalTabIndex, ExternalTitle, LongDesc, HrefLang]</td></tr>
<tr><td>MapDBField</td><td>[Background, Foreground]||[padding-top, padding-bottom, padding-left, padding-right]||[BorderWidth, TopLine, BottomLine, LeftLine, RightLine, Shadow, BorderColor, ShadowColor]||[PatternColor, PatternStyle]||[FontFace, FontSize, Bold, Italic, StrikeOut, Underline, Format, AutoFit, MaxWidth, WordWrap, VerticalAlignment, HorizontalAlignment]||[Invisible, DataMappingFile, DisplayNull, ColumnName, CacheValue, AutoConnector]||[ExternalCSSClassValue, ExternalID, ExternalStyle, Lang, ExternalAccessKey, ExternalDir, ExternalTabIndex, ExternalTitle, LongDesc, HrefLang]</td></tr>
<tr><td>MapTextField</td><td>[Text, FontFace, FontSize, Bold, Italic, StrikeOut, Underline, AutoFit, MaxWidth, WordWrap, VerticalAlignment, HorizontalAlignment]||[Background, Foreground]||[padding-top, padding-bottom, padding-left, padding-right]||[BorderWidth, TopLine, BottomLine, LeftLine, RightLine, Shadow, BorderColor, ShadowColor]||[PatternColor, PatternStyle]||[Invisible, AutoConnector]||[ExternalCSSClassValue, ExternalID, ExternalStyle, Lang, ExternalAccessKey, ExternalDir, ExternalTabIndex, ExternalTitle, LongDesc, HrefLang]</td></tr>
<tr><td>ModelingDataSet</td><td>[MaxRecords]||[SecurityName]||[RecordsPerPage, MaxPageNumber]||[dsName, queryName]</td></tr>
<tr><td>MultiValueContainer</td><td>[X, Y, Width, Height]||[Background]||[Style, class, id]||[position, ExportToDHTML, Invisible, Suppressed, SuppressedIfNoRecords, RecordLocation, ExportToCSV, ExportToXLS, ExportToPDF]||[ColumnIndex, RowIndex]||[BorderWidth, TopLine, BottomLine, LeftLine, RightLine, BorderColor]||[FontFace, FontSize, Bold, Italic, Foreground]||[ExternalCSSClassValue, ExternalStyle, ExternalTitle, Lang]</td></tr>
<tr><td>NavigationCtrl</td><td>[X, Y, Width, Height]||[Background]||[Style, class]||[position, ExportToDHTML, Invisible, ExportToCSV, ExportToXLS]||[ColumnIndex, RowIndex]||[BorderColor, BorderWidth, TopLine, BottomLine, LeftLine, RightLine]</td></tr>
<tr><td>NavigationItem</td><td>[AutoFit, Bold, Italic, FontFace, FontSize, Underline, Text, StrikeOut, WordWrap, HorizontalAlignment, VerticalAlignment]||[Background, Foreground]||[padding-top, padding-bottom, padding-left, padding-right]||[BorderColor, BorderWidth, TopLine, BottomLine, LeftLine, RightLine, Shadow, ShadowColor]||[PatternStyle, PatternColor]</td></tr>
<tr><td>OLEObject</td><td>[X, Y, Width, Height]||[Background]||[Style, class]||[position, ExportToDHTML, Invisible, Suppressed, SuppressedIfNoRecords, RecordLocation]||[ColumnIndex, RowIndex]||[BorderWidth, TopLine, BottomLine, LeftLine, RightLine, BorderColor]||[Declare, ClassID, CodeBase, Type, Archive, StandBy, Usemap, Name, TabIndex, Accesskey, Alt, Title, VSpace, HSpace]||[ExternalCSSClassValue]</td></tr>
<tr><td>OnDemandDataSet</td><td>[MaxRecords]||[SecurityName]||[RecordsPerPage, MaxPageNumber]||[dsName, queryName]</td></tr>
<tr><td>OvalShape</td><td>[TopAttachPosX, TopAttachPosY, BottomAttachPosX, BottomAttachPosY]||[Background]||[Style, class]||[ExportToDHTML, Invisible, SuppressedIfNoRecords, RecordLocation, ExportToXLS]||[TopAttachCol, TopAttachRow, BottomAttachCol, BottomAttachRow]||[BorderColor, BorderStyle, BorderWidth]||[ExternalCSSClassValue, ExternalID, ExternalStyle, Lang, ExternalTitle]</td></tr>
<tr><td>PageFooterPanel</td><td>[Width, Height]||[Background]||[Style, class]||[ExportToDHTML, Invisible, Suppressed, SuppressedIfNoRecords, RecordLocation, ExportToCSV, ExportToXLS, Underlay, SuppressBlankSection, ShowBottomLine, SuppressInLastPage]||[BorderWidth, TopLine, BottomLine, LeftLine, RightLine, BorderColor]||[ExternalCSSClassValue, ExternalID, ExternalStyle, Lang]</td></tr>
<tr><td>PageHeaderPanel</td><td>[Width, Height]||[Background]||[Style, class]||[ExportToDHTML, Invisible, Suppressed, SuppressedIfNoRecords, RecordLocation, ExportToCSV, ExportToXLS, Underlay, SuppressBlankSection, ShowBottomLine]||[BorderWidth, TopLine, BottomLine, LeftLine, RightLine, BorderColor]||[ExternalCSSClassValue, ExternalID, ExternalStyle, Lang]</td></tr>
<tr><td>PagePanel</td><td>[PageType, Orientation, Width, Height]||[TopMargin, LeftMargin, BottomMargin, RightMargin]</td></tr>
<tr><td>ParallelDetailPanel</td><td>[Width, Height]||[Background]||[Style, class]||[ExportToDHTML, Invisible, Suppressed, SuppressedIfNoRecords, RecordLocation, ExportToCSV, ExportToXLS, RemoveBlankRow, Label, OnNewPage, Underlay, CanCrossPage, SuppressBlankSection, FillWholePage, ShowBottomLine, TileDetailSection, TileHorizontal, CacheSection, currentBlockIndex, itemsPerBlock, BranchName]||[BorderWidth, TopLine, BottomLine, LeftLine, RightLine, BorderColor]||[ExternalCSSClassValue, ExternalID, ExternalStyle, Lang]</td></tr>
<tr><td>ParallelGroupPanel</td><td>[isExpandData, ShrinkFooter, currentBlockIndex, itemsPerBlock, BranchName]||[KeepGroupTogether, RepeatWhileGF]||[TOCAnchor, AnchorDispValue]||[Grant, Cascade, Groups, Roles]</td></tr>
<tr><td>ParameterField</td><td>[X, Y, Width, Height]||[Background, Foreground]||[Style, id, class]||[ColumnIndex, RowIndex]||[padding-top, padding-bottom, padding-left, padding-right]||[BorderWidth, TopLine, BottomLine, LeftLine, RightLine, Shadow, BorderColor, ShadowColor]||[PatternColor, PatternStyle]||[FontFace, FontSize, Bold, Italic, StrikeOut, Underline, Format, AutoFit, MaxWidth, WordWrap, TransWhileToHtml, VerticalAlignment, HorizontalAlignment]||[position, ExportToDHTML, Invisible, Suppressed, SuppressedIfNoRecords, RecordLocation, ExportToCSV, ExportToXLS, Link, EnableHyperlinkOnPDF, EnableHyperlinkOnExcel, EnableHyperlinkOnHTML, DetailReport, TransferStyle, GotoDetail, DetailTargetFrame, ExportToPDF, Parameter, SuppressedIfNull, DisplayNull, DataMappingFile, ValueDelimiter]||[TOCAnchor, AnchorDispValue]||[ExternalCSSClassValue, ExternalID, ExternalStyle, Lang, ExternalAccessKey, ExternalDir, ExternalTabIndex, HrefLang]</td></tr>
<tr><td>ParamForm</td><td>[X, Y, Width, Height]||[Background]||[Style, class, id]||[position, ExportToDHTML, Invisible, ExportToCSV, ExportToXLS]||[IsShowTitle, TitleText, TitleBold, TitleItalic, TitleFontFace, TitleFontSize, TitleForeground, TitleUnderline, TitleHorizontalAlignment, TitleBackground]||[ColumnIndex, RowIndex]||[BorderColor, BorderWidth, TopLine, BottomLine, LeftLine, RightLine]||[PatternStyle, PatternColor]||[ExternalCSSClassValue, ExternalStyle, Lang]</td></tr>
<tr><td>PolyLine</td><td>[Invisible]||[LineColor, LineStyle, LineWidth]||[ExternalCSSClassValue, ExternalID, ExternalStyle, Lang, ExternalTitle]</td></tr>
<tr><td>QueryDataSet</td><td>[MaxRecords]||[SecurityName]||[RecordsPerPage, MaxPageNumber]||[dsName, queryName]</td></tr>
<tr><td>RefParameterField</td><td>[X, Y, Width, Height]||[Background, Foreground]||[Style, id, class]||[ColumnIndex, RowIndex]||[BorderWidth, TopLine, BottomLine, LeftLine, RightLine, Shadow, BorderColor, ShadowColor]||[PatternColor, PatternStyle]||[FontFace, FontSize, Bold, Italic, Underline, Format, AutoFit, WordWrap, VerticalAlignment, HorizontalAlignment]||[position, ExportToDHTML, Invisible, ExportToCSV, ExportToXLS, Parameter]||[TOCAnchor, AnchorDispValue]||[ExternalCSSClassValue, ExternalStyle, Lang, ExternalTitle]</td></tr>
<tr><td>Report</td><td>[ResultBufferSize]||[InitReportCube]||[AppletWidth, AppletHeight, PageBackground, BeforeRun, AfterRun, AfterInitParameter, ShowSubHeaderFooter, EmbeddedFonts, StyleGroup, NoTempFile, ImportJavaScript, PrecisionSensitive, PortletReport]||[ExcelBufferSize, RowsPerSheet, ColumnWidthList, FastPass, Columned, SheetName]||[StyleGroupOfDHTML, StyleGroupOfRST, StyleGroupOfHTML, StyleGroupOfExcel, StyleGroupOfPDF, StyleGroupOfText, StyleGroupOfRTF, StyleGroupOfXML, StyleGroupOfPS, StyleGroupOfFax]||[ExportFormat]</td></tr>
<tr><td>ReportBody</td><td>[Background]||[Style, class]||[ExportToDHTML, RecordLocation, ExportToCSV, ExportToXLS, SuppressEmpty]||[ColumnIndex, RowIndex]||[BorderWidth, TopLine, BottomLine, LeftLine, RightLine, BorderColor]||[TOCAnchor]||[ExternalCSSClassValue, ExternalID, ExternalStyle, Lang]</td></tr>
<tr><td>ReportSet</td><td>[ImportParamValues, ParaListAuto]</td></tr>
<tr><td>RoundBoxShape</td><td>[TopAttachPosX, TopAttachPosY, BottomAttachPosX, BottomAttachPosY]||[Background]||[Style, class]||[ExportToDHTML, Invisible, SuppressedIfNoRecords, RecordLocation, ExportToXLS]||[TopAttachCol, TopAttachRow, BottomAttachCol, BottomAttachRow]||[CornerFactor, BorderColor, BorderStyle, BorderWidth]||[ExternalCSSClassValue, ExternalID, ExternalStyle, Lang, ExternalTitle]</td></tr>
<tr><td>SliderControl</td><td>[FontFace, FontSize, Format, NumberSliderUnitIncrement]||[X, Y, Width, Height]||[Background, Foreground]||[Style, class, id]||[position, ExportToDHTML, Invisible, ExportToCSV, ExportToXLS]||[ColumnIndex, RowIndex]||[ExternalCSSClassValue, ExternalID, ExternalStyle, Lang, ExternalTitle]</td></tr>
<tr><td>SpecialField</td><td>[X, Y, Width, Height]||[Background, Foreground]||[Style, id, class]||[ColumnIndex, RowIndex]||[padding-top, padding-bottom, padding-left, padding-right]||[BorderWidth, TopLine, BottomLine, LeftLine, RightLine, Shadow, BorderColor, ShadowColor]||[PatternColor, PatternStyle]||[FontFace, FontSize, Bold, Italic, StrikeOut, Underline, Format, AutoFit, MaxWidth, WordWrap, TransWhileToHtml, VerticalAlignment, HorizontalAlignment]||[position, ExportToDHTML, Invisible, Suppressed, SuppressedIfNoRecords, RecordLocation, ExportToCSV, ExportToXLS, Link, EnableHyperlinkOnPDF, EnableHyperlinkOnExcel, EnableHyperlinkOnHTML, DetailReport, TransferStyle, GotoDetail, DetailTargetFrame, ExportToPDF, FieldType, DisplayNull]||[TOCAnchor, AnchorDispValue]||[ExternalCSSClassValue, ExternalID, ExternalStyle, Lang, ExternalAccessKey, ExternalDir, ExternalTabIndex, ExternalTitle, LongDesc, HrefLang]</td></tr>
<tr><td>Subreport</td><td>[X, Y, Width, Height]||[Background]||[Style, class]||[position, ExportToDHTML, Invisible, Suppressed, SuppressedIfNoRecords, RecordLocation, ExportToCSV, ExportToXLS, SuppressEmpty, Embedded, RecPerPage, MaxPageNum, SRSourceName, SRQueryName, SRSecurity, Cache]||[ColumnIndex, RowIndex, OnNewSheet, SheetName, Postfix]||[BorderWidth, TopLine, BottomLine, LeftLine, RightLine, BorderColor]||[TOCAnchor, AnchorDispValue]||[ExternalCSSClassValue, ExternalID, ExternalStyle, Lang]</td></tr>
<tr><td>TableCell</td><td>[Width, Height]||[Background]||[Style, class]||[ExportToDHTML, SuppressedIfNoRecords, RecordLocation, ExportToCSV, ExportToXLS, JoiningMerge, HorizontalAlignment, VerticalAlignment, Repeat]||[BorderWidth, TopLine, BottomLine, LeftLine, RightLine, BorderColor]||[ExternalCSSClassValue, ExternalID, Abbr, Axis, Headers, Scope]</td></tr>
<tr><td>TableColumn</td><td>[Invisible, AutoFit]</td></tr>
<tr><td>TableComp</td><td>[X, Y, Width, Height]||[Background]||[Style, class]||[position, ExportToDHTML, Invisible, Suppressed, SuppressedIfNoRecords, RecordLocation, ExportToCSV, ExportToXLS, Horizontal, Cache, SuppressEmpty, AutoExpands, CDNUsed, CDNHeight, CDNWidth]||[BorderWidth, TopLine, BottomLine, LeftLine, RightLine, BorderColor]||[TOCAnchor, AnchorDispValue]||[ExternalCSSClassValue, ExternalID, ExternalStyle, Summary]</td></tr>
<tr><td>TableDetail</td><td>[Width, Height]||[Background]||[Style, class]||[ExportToDHTML, Invisible, Suppressed, SuppressedIfNoRecords, RecordLocation, ExportToCSV, ExportToXLS, Label, OnNewPage, FillWholePage, currentBlockIndex, itemsPerBlock]||[BorderWidth, TopLine, BottomLine, LeftLine, RightLine, BorderColor]||[ExternalCSSClassValue]</td></tr>
<tr><td>TableDetailColumn</td><td>[Invisible, AutoFit]</td></tr>
<tr><td>TableFooter</td><td>[Width, Height]||[Background]||[Style, class]||[ExportToDHTML, Invisible, Suppressed, SuppressedIfNoRecords, RecordLocation, ExportToCSV, ExportToXLS, Label, OnNewPage, FillWholePage]||[BorderWidth, TopLine, BottomLine, LeftLine, RightLine, BorderColor]||[ExternalCSSClassValue, ExternalID, ExternalStyle, Lang]</td></tr>
<tr><td>TableGroup</td><td>[isExpandData, currentBlockIndex, itemsPerBlock]||[KeepGroupTogether]||[TOCAnchor, AnchorDispValue]||[Grant, Cascade, Groups, Roles]</td></tr>
<tr><td>TableGroupColumn</td><td>[Invisible, AutoFit]</td></tr>
<tr><td>TableGroupFooter</td><td>[Width, Height]||[Background]||[Style, class]||[ExportToDHTML, Invisible, Suppressed, SuppressedIfNoRecords, RecordLocation, ExportToCSV, ExportToXLS, Label, OnNewPage, FillWholePage]||[BorderWidth, TopLine, BottomLine, LeftLine, RightLine, BorderColor]||[ExternalCSSClassValue, ExternalID, ExternalStyle, Lang]</td></tr>
<tr><td>TableGroupHeader</td><td>[Width, Height]||[Background]||[Style, class]||[ExportToDHTML, Invisible, Suppressed, SuppressedIfNoRecords, RecordLocation, ExportToCSV, ExportToXLS, Label, OnNewPage, FillWholePage, repeat]||[BorderWidth, TopLine, BottomLine, LeftLine, RightLine, BorderColor]||[ExternalCSSClassValue, Lang]</td></tr>
<tr><td>TableHeader</td><td>[Width, Height]||[Background]||[Style, class]||[ExportToDHTML, Invisible, Suppressed, SuppressedIfNoRecords, RecordLocation, ExportToCSV, ExportToXLS, Label, OnNewPage, repeat, RepeatHeaderForExcel, RepeatHeaderForXML, RepeatHeaderForTxt]||[BorderWidth, TopLine, BottomLine, LeftLine, RightLine, BorderColor]||[ExternalCSSClassValue, ExternalID, Lang]</td></tr>
<tr><td>TableSummaryColumn</td><td>[Invisible, AutoFit]</td></tr>
<tr><td>Tabular</td><td>[X, Y, Width, Height]||[Background]||[Style, class]||[position, ExportToDHTML, Invisible, Suppressed, ExportToCSV, ExportToXLS, VerticalAutoSize, HorizontalAutoSize]||[ColumnIndex, RowIndex]||[TOCAnchor, AnchorDispValue]||[ExternalCSSClassValue]</td></tr>
<tr><td>TabularCell</td><td>[X, Y, Width, Height]||[Background]||[Style, class]||[ExportToCSV, ExportToXLS, repeatContent, HorizontalAlignment, VerticalAlignment]||[ColumnIndex, RowIndex]||[BorderWidth, TopLine, BottomLine, LeftLine, RightLine, BorderColor]||[TOCAnchor, AnchorDispValue]||[ExternalCSSClassValue]</td></tr>
<tr><td>TabularRowGroup</td><td>[FillWholePage, OnNewPage, CanCrossPage]</td></tr>
<tr><td>TextBox</td><td>[X, Y, Width, Height]||[Background]||[Style, class]||[position, ExportToDHTML, Invisible, Suppressed, SuppressedIfNoRecords, RecordLocation, ExportToCSV, ExportToXLS, VerticalAlignment, VerticalAutoSize]||[ColumnIndex, RowIndex]||[BorderWidth, TopLine, BottomLine, LeftLine, RightLine, BorderColor]||[TOCAnchor, AnchorDispValue]||[ExternalCSSClassValue]</td></tr>
<tr><td>TextField</td><td>[Text, FontFace, FontSize, Bold, Italic, StrikeOut, Underline, AutoFit, MaxWidth, WordWrap, TransWhileToHtml, VerticalAlignment, HorizontalAlignment]||[X, Y, Width, Height]||[Background, Foreground]||[Style, id, class]||[ColumnIndex, RowIndex]||[padding-top, padding-bottom, padding-left, padding-right]||[BorderWidth, TopLine, BottomLine, LeftLine, RightLine, Shadow, BorderColor, ShadowColor]||[PatternColor, PatternStyle]||[position, ExportToDHTML, Invisible, Suppressed, SuppressedIfNoRecords, RecordLocation, ExportToCSV, ExportToXLS, Link, EnableHyperlinkOnPDF, EnableHyperlinkOnExcel, EnableHyperlinkOnHTML, DetailReport, TransferStyle, GotoDetail, DetailTargetFrame, ExportToPDF, FilterOptions, Sortable, Filterable, BindColumn]||[TOCAnchor, AnchorDispValue]||[ExternalCSSClassValue, ExternalID, ExternalStyle, Lang, ExternalAccessKey, ExternalDir, ExternalTabIndex, ExternalTitle, LongDesc, HrefLang]</td></tr>
<tr><td>TitleText</td><td>[Text, FontFace, FontSize, Bold, Italic, StrikeOut, Underline, AutoFit, MaxWidth, WordWrap, TransWhileToHtml, VerticalAlignment, HorizontalAlignment]||[X, Y, Width, Height]||[Background, Foreground]||[Style, id, class]||[ColumnIndex, RowIndex]||[padding-top, padding-bottom, padding-left, padding-right]||[BorderWidth, TopLine, BottomLine, LeftLine, RightLine, Shadow, BorderColor, ShadowColor]||[PatternColor, PatternStyle]||[position, ExportToDHTML, Invisible, Suppressed, SuppressedIfNoRecords, RecordLocation, ExportToCSV, ExportToXLS, Link, EnableHyperlinkOnPDF, EnableHyperlinkOnExcel, EnableHyperlinkOnHTML, DetailReport, TransferStyle, GotoDetail, DetailTargetFrame, ExportToPDF, FilterOptions, Sortable, Filterable, BindColumn]||[TOCAnchor, AnchorDispValue]||[ExternalCSSClassValue, ExternalID, ExternalStyle, Lang, ExternalAccessKey, ExternalDir, ExternalTabIndex, ExternalTitle, LongDesc, HrefLang]</td></tr>
<tr><td>TOC</td><td>[Background, Foreground]||[Style, class]||[FontFace, FontSize, Bold, Italic, StrikeOut, Underline]||[ExternalCSSClassValue, ExternalID, ExternalStyle, Lang, ExternalTitle]||[RootLabel]</td></tr>
</table>

Export option keys

The following list is for your reference when specifying the value of the parameter optionValues applied in the web actions user_oneStepExport and user_downloadReport. Note that the option keys here can be recognized by classes and are different from the option names provided in Page Report Studio UI which are just display names of the export options.

Export Type	Export Option Key	Available Value	Description
HTML	to_ver	true, false	Specifies whether or not to save the result to version.
	to_open	true, false	Specifies whether or not to export and open the result file.
	to_local	true, false	Specifies whether or not to save the result to a local file.
	browser	0 (Internet Explorer), 2 (Firefox).	Specifies the web browser type.
	imagetype	0 (decided by JReport), 1 (GIF), 2 (JPG), 3 (PNG).	Specifies the type of the images in the result file.
	overflow	0 (VISIBLE), 1 (HIDDEN), 2 (VERFLOWCOUNT).	Specifies the overflow type.
	resolution	Any integer between 1 and 4294967296.	Specifies the HTML resolution.
	title	Any string	Specifies the title for the HTML file.
	applet	true, false	Specifies whether or not to export chart in Java applet format.
	css	true, false	Specifies whether or not to embed the cascading style sheet in the exported HTML files.
	multi	true, false	Specifies whether or not to generate an HTML file for each page of the report result.
	hyperlink	true, false	Specifies whether or not to contain hyperlinks in the HTML file.
	pagenumber	true, false	Specifies whether or not to contain page numbers in the HTML file.
	drilldown	true, false	Specifies whether or not to include the drilled-down file in the exported HTML file.
	no_margin	true, false	Specifies whether or not to remove the original margins.
	absolute	true, false	Specifies whether or not to make the font size fixed in the web browser.
PDF	to_ver	true, false	Specifies whether or not to save the result to version.
	to_open	true, false	Specifies whether or not to export and open the result file.
	to_local	true, false	Specifies whether or not to save the result to a local file.
	no_margin	true, false	Specifies whether or not to remove the margin.
	simulate	true, false	Specifies whether or not the mode is to be Simulated Printing Mode.
	standard	true, false	Specifies whether or not to set the mode as Standard Mode.
	content	true, false	Specifies whether or not to contain the TOC in the exported PDF file.
	drilldown	true, false	Specifies whether or not to include the drilled-down file in the exported PDF file.
	encrypt	true, false	Specifies whether or not to encrypt the report result.
	compress	true, false	Specifies whether or not to compress the images in the report.
	ratio	Any integer between 1 to 100.	Specifies the percentage with which to compress the images in the report.
	compatibitly	0: Acrobat 3.0 and later. 1: Acrobat 5.0 and later.	Specifies the encryption compatibility.
	doc_psw	Any string	Specifies the password for opening the PDF file when encrypt=true.
	permi_psw	Any string	Specifies the password for printing and editing the PDF file when encrypt=true.
	printing	0: Prevents users from printing the file. 4: Allows low resolution-printing. 2052: Allows high-resolution printing.	Specifies the PDF printing mode.
	changes	0: Prevents users from making any changes to the file). 1024: Allows inserting, deleting, and rotating pages. 256: Allows users to fill in form fields and adding digital signatures. 32: Aallows users to fill in form fields and add digital signatures and comments. 40: Allows users to do anything except extracting pages. 2108: Allows all.	Defines which editing actions are allowed in the PDF file.
	enable_copy	true, false	Specifies whether or not to allow users to copy the file contents.
	enable_access	true, false	Specifies whether or not to let visually impaired users read the document with window readers.
Text	to_ver	true, false	Specifies whether or not to save the result to version.
	to_open	true, false	Specifies whether or not to export and open the result file.
	to_local	true, false	Specifies whether or not to save the result to a local file.
	repeat	true, false	Specifies whether or not to replace a field value of a record with that of its previous record if the field value is null.
	compress	true, false	Specifies whether or not to compress the clearance between columns.
	win_linebreak	true, false	Specifies whether or not to use Windows end-of-line characters.
	normal	true, false	Specifies whether or not to use the standard data format.
	quote_mark	true, false	Specifies whether or not to mark the fields in the exported file with quotation marks.
	head_foot	true, false	Specifies whether or not to contain all headers and footers in the report.
	delimiter	Any single character	Specifies the delimiter.
	width	An integer	Specifies the user-defined character width.
	height	An integer	Specifies the user-defined character height.
Excel	to_ver	true, false	Specifies whether or not to save the result to version.
	to_open	true, false	Specifies whether or not to export and open the result file.
	to_local	true, false	Specifies whether or not to save the result to a local file.
	wrap	0 (All Keep Existing), 1 (All Disabled), 2 (All Enabled).	Specifies the word-wrap setting.
	new_layout	true, false	Specifies whether or not to use the new layout mode.
	shapes	true, false	Specifies whether or not to include the shapes (drawing objects) in the exported file.
	excel_2000	true, false	Specifies whether or not to export the result in Data Format, which means only the report data will be exported without format.
	advanced	true, false	Specifies whether or not to apply the advanced options.
	header	Any string	Specifies the page header text.
	footer	Any string	Specifies the page footer text.
	gridline	true, false	Specifies whether or not to print gridlines when printing the exported Excel file.
XML	to_ver	true, false	Specifies whether or not to save the result to version.
	to_open	true, false	Specifies whether or not to export and open the result file.
	to_local	true, false	Specifies whether or not to save the result to a local file.
	only_data	true, false	Specifies whether or not to only contain the database column information in the exported XML file.
	schema	An existing schema file with its full path.	Specifies the name of an existing schema file with its full path with which to generate the XML file.
RTF	to_ver	true, false	Specifies whether or not to save the result to version.
	to_open	true, false	Specifies whether or not to export and open the result file.
	to_local	true, false	Specifies whether or not to save the result to a local file.
	rtf_flow	true, false	Specifies whether or not to apply a flow layout when exporting the report to RTF.
	no_margin	true, false	Specifies whether or not to remove the margins in the RTF file.
PostScript	to_ver	true, false	Specifies whether or not to save the result to version.
	to_open	true, false	Specifies whether or not to export and open the result file.
	to_local	true, false	Specifies whether or not to save the result to a local file.
	no_margin	true, false	Specifies whether or not to remove the margins in the PS file.