Welcome to the JReport Tutorial

Thank you for your interest in JReport, the leading Embedded Reporting Solution for Java Applications. With JReport, intuitive and sophisticated reporting can be customized and integrated as a natural extension of any application.

This guide is intended to provide an overview of the JReport solution, embedded reporting concepts, and detailed step-by-step lessons on using JReport. To allow for lessons to be completed in a reasonable time frame, not all of the features of JReport are covered.

This guide contains three parts:

- Part I: JReport Overview and Setup
  This part briefly introduces reports and JReport.
- Part II: Quick Start
  This part introduces the most powerful yet easy-to-use web tools of JReport: JDashboard, Visual Analysis and Web Report Studio.
- Part III: Advanced Reporting
  This part talks about some advanced reporting features of JReport: how to use JReport Designer to create data resources and use the resources to build reports; how the resources are published to JReport Server and managed on the server, and so on.

Unix platform

This version of the JReport Tutorial is developed and written for Windows. File names and paths, sample JSP code, and screen shots in this guide are based on the Windows 7 operating system. However, you can get a Unix version of the JReport Tutorial by changing the paths to the Unix format.

How to contact us

If you encounter any technical issues that prevent completion of the tasks outlined in this guide, please do one of the following:

- Contact our technical support department via e-mail at http://www.jinfonet.com/support, available 24 hours a day, seven days a week. This service is available free of charge to registered evaluators for up to 30 days after you receive the JReport Evaluation package.
- Search the JReport Knowledge Base in the Customer Care Portal at http://customercare.jinfonet.com for issue resolution. You will need to create a user account as described on the main page of the Customer Care Portal.

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

Click the right arrow ( ) below to begin reading the first track.

Part I: JReport Overview and Setup

This chapter provides an overview of JReport. It contains the following sections:

- Who uses JReport?
- What is a report?
- Lifecycle of a report
- JReport product overview
- JReport sample reports
- Setting up JReport on your system

Click the right arrow ( ) below to begin reading the chapter.

Who uses JReport?

JReport delivers an enterprise-wide solution. Therefore different types of users throughout your organization will use JReport. Each type of user will be able to understand the features and find value in JReport as it relates to their job function or reporting requirements.

There are five general types of JReport users. Each type of user can focus on specific areas of this tutorial as described below:

- **Business Analyst**

  If you are a business analyst, you should understand how JReport Page Report Studio and Web Report Studio allow you to create a special category of reports called ad hoc reports. Unlike the predefined reports in JReport Designer, you build these reports in the runtime environment based on a data model built and published by a report developer.

  You can also create a user portal interface rather than a web report or page report, using predefined data components with JDashboard, or use the WYSIWYG product JReport Visual Analysis to visualize the result of every step of your work.

  Focus on tracks in Part II.

- **Report Developer**

  If you are a report developer, you will use JReport Designer, JReport's visual design environment. This intuitive desktop design tool uses familiar conventions such as property panels, toolbars, style sheets, and drag and drop placement to support every aspect of the report design process. You will quickly become proficient in using the design environment and be able to create professional reports.

  Focus on Track 1, Track 2 and Track 3 in Part III.

- **Application Developer**

  If you are a Java application developer, you should understand that both JReport Designer and JReport Server are 100% Java-based tools that run on almost any platform. By offering a robust set of Java APIs, JReport can be seamlessly embedded as a natural extension of your application. In addition to being directly accessed from any Java program via APIs, all reports and functionality can be accessed through HTTP.

  Focus on Track 1 and then Track 5 in Part III.

- **Systems Analyst or Application Server Administrator**

  If you are a systems analyst or application server administrator, you should know that the JReport solution is managed from a single access point, a web-based console. The JReport solution offers many different deployment options, enabling existing architecture to be leveraged. It can be embedded in a web application via a self-contained WAR/EAR file to provide a reporting service or it can operate as a standalone server.

  See Publishing, running and administering resources.

- **End User**

  If you are an end user of reports, you should understand the many different presentation strategies that are available. You can decide which format best delivers the information that you need to make timely and critical business decisions. With JReport, reports can be viewed and exported to a variety of formats including Applet, HTML, PDF, Excel, XML, RTF, CSV, PostScript, Page Report Result and Web Report Result. JReport's Page Report Result and Web Report Result outputs enable you to interact with and customize report views to obtain exactly the information needed.

  See JReport sample reports and Creating and analyzing ad hoc reports.

What is a report?

A report is comprised of a report layout and a dataset. JReport Designer is used to create report templates that JReport Server combines with the data retrieved from the data source to render report results at runtime.

Reports can be classified into three categories:

- Predefined - Report results that cannot be modified by the end user. The layout and data to be included can only be defined at report design time. The dataset used to populate the report can change each time the report is run.
- Dynamic - Report results that can be modified by the end user. The layout and data to be included is defined at report design time but can also be modified at runtime. Dynamic reports can be modified using parameters, web controls, or in Page Report Studio or Web Report Studio.
- Ad Hoc - Reports are built and can be modified at runtime by the end user. Other than building the business views to be used as the data source for ad hoc reports, no design work is required. Ad hoc reports can be built from scratch using a wizard or drag & drop.

Report template

A report template contains static text and graphics as well as placeholders for data.

When a report is in the runtime environment, it connects to the data source associated with the report, executes the query, and applies the fetched data to the template thereby creating a report result file:

Therefore, each report result represents a unique data set; the one that exists at the time the query is run.

Data sources and datasets

Reports pull data from data sources via queries, and then expose the query results to the report designer. JReport supports the following types of data sources:

- Relational databases through JDBC connections
- Data warehouses
- Java objects
- XML data
- Metadata layers
- User-defined data sources
- Hierarchical data sources
- Web service data sources
- MongoDB data sources

A dataset is the data that the JReport Server returns at runtime from a JReport query. A JReport query is comprised of one of the following:

- Imported SQL query file
- JReport defined query
- Stored Procedure call
- User-defined data source query
- Hierarchical data source query

Multiple data source connections can be used to generate multiple datasets. A report can use one or more datasets. Datasets can be used at design time, or at runtime for ad hoc report creation.

Lifecycle of a report

Just like an application, a report has a distinct life cycle. The life cycle contains the following phases:

Phase 1: Determine requirements (report developer)

The first fundamental requirement comes from the intended end users of the report. First, determine who will be the end user of the report and then identify the general purpose of the report. Ask what decisions those users need to make and how often they need to make them (daily, monthly, or other).

Second, you should determine the specific pieces of data that need to be presented in the report and how the pieces map to the data source. Look for common data elements that span multiple reports—these are candidates for subreports.

Third, you need to determine the security implications associated with the report. Are there pieces of data that need restricted access? Are there regulatory drivers of the report?

Fourth, determine the expected demand of the report result. Will on-demand report results be necessary or can the report be scheduled? Will report results need to be saved, and for how long?

Fifth, determine the desired report output format. For most Java applications delivery via the Web is the preferred method to present information. However, there may be other end users who don't need or want Web-based information. Perhaps they require the report be delivered in a standard business format or printed.

Phase 2: Develop report template (report developer)

A template can be thought of as a report blueprint that contains static text and graphical objects as well as placeholders to display the data pieces needed on the report. The template definition includes the query that needs to execute to provide the data, as well as the database connection on which to execute it.

Share a report prototype that includes sample data with the end users to see if it meets their requirements and also to obtain feedback on the scope and layout of the report.

Phase 3: Publish report results (system administrator)

Publishing a report template executes the query and merges the resulting data set with the template. The result is a report instance that is available in the context of JReport Server. Report results can be saved to other locations, and in various formats such as HTML, PDF, RTF, and others.

Communicate with the end users regarding how they can access the report and then provide training, as needed. Include a way for the end users to provide feedback; acknowledge feedback and build release schedule.

As report production scales up, the system administrator should monitor performance and apply the appropriate load balancing and security measures.

Phase 4: Access report results (end user and business analyst) and administer (system administrator)

After a report is generated, end users can access it in a variety of ways. A report can be viewed through the JReport Console, through a Java application, or routed to a delivery target such as an e-mail address, printer, or Web site. The business analyst can also build ad hoc reports as needed.

The system administer monitors the report access environment through the JReport Administration Console.

Phase 5: Update report template (report developer)

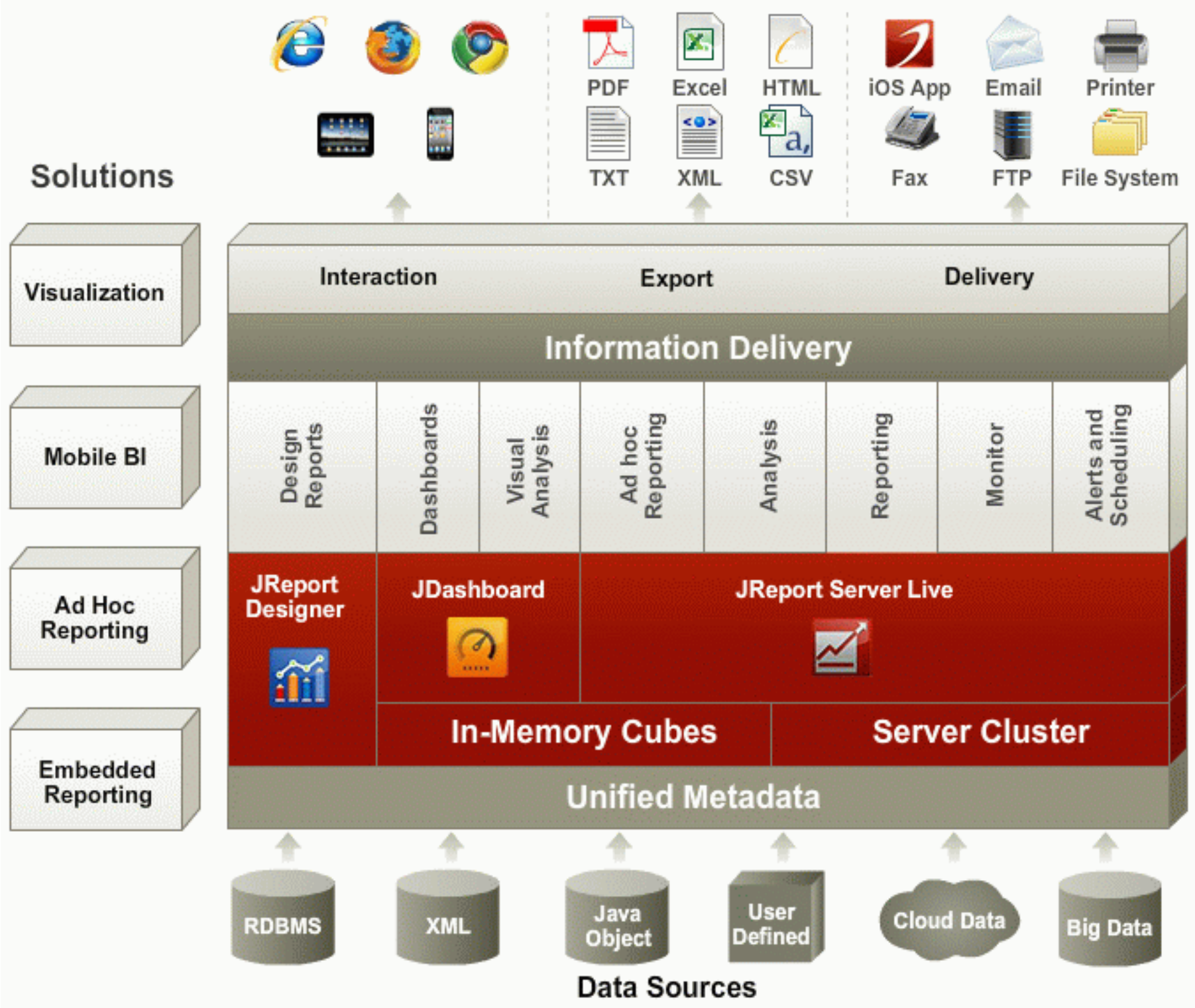
Collect feedback from the end users to determine any needed improvements to the layout or behavior of the report. Also, modify security as needed (add/drop users) and update data source connections.

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'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 Server, 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, etc. 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.

JReport sample reports

JReport comes with many pre-built sample reports. This document describes a few of the sample reports, so that you can see how your own reports may look. The reports that you will be building in this tutorial are simpler than the sample reports.

Report sample 1: Invoice report

This is a typical invoice report, on which end users can perform the filtering, sorting and going functions on key columns by right-click functionality:

Invoice Report

Thank you for your order. Please print this page for your records.

Invoice NO.: 3033

Date: 4/14/2014

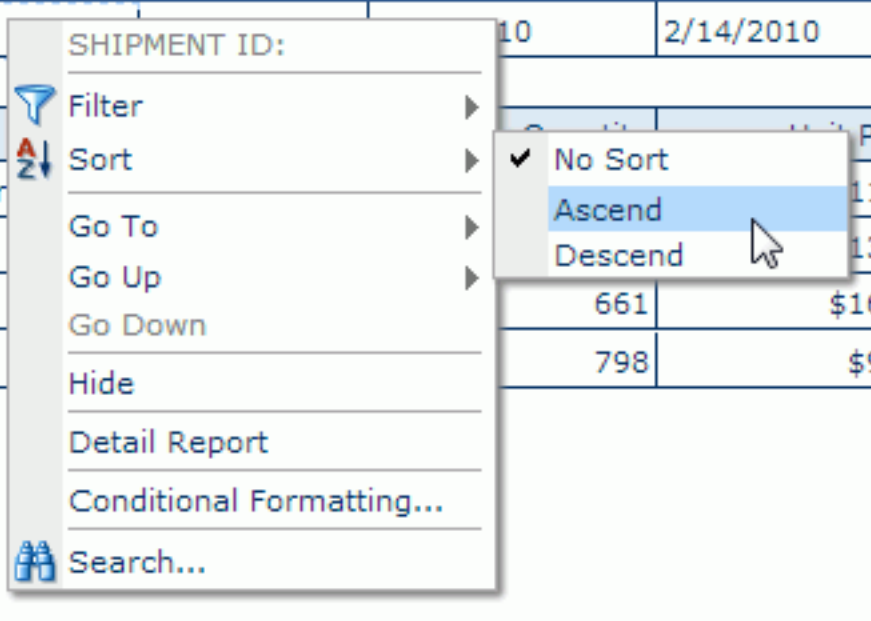
Bill to: Fatima Anderson, Cafe Brazil - Sydney, A.M.P. Centre, Sydney, New South Wales 2000, Australia, +61 (2) 9225 99999

Ship to: Fatima Anderson, Cafe Brazil - Sydney, A.M.P. Centre, Sydney, New South Wales 2000, Australia, +61 (2) 9225 99999

Payment method:
Credit Card

Order Information:

Order #	Shipment #	Order Date	Required Date	Ship Date	Ship Via	Terms
3033	8169		10	2/14/2010	On-Board Transport	30 days



Item		Price	Discount%	Total
Tanzania Blackbur		11.45	15%	$7,786.00
Mexico Organic		13.45	20%	$11,739.16
Rift Valley Blend	661	$16.95	10%	$10,083.55
Espresso Roast	798	$9.90	10%	$7,110.18
			Sub-total:	**$36,718.90**
			Shipping:	**$600.00**
			Tax:	**$1,835.94**
			Invoice Total:	**$39,154.84**

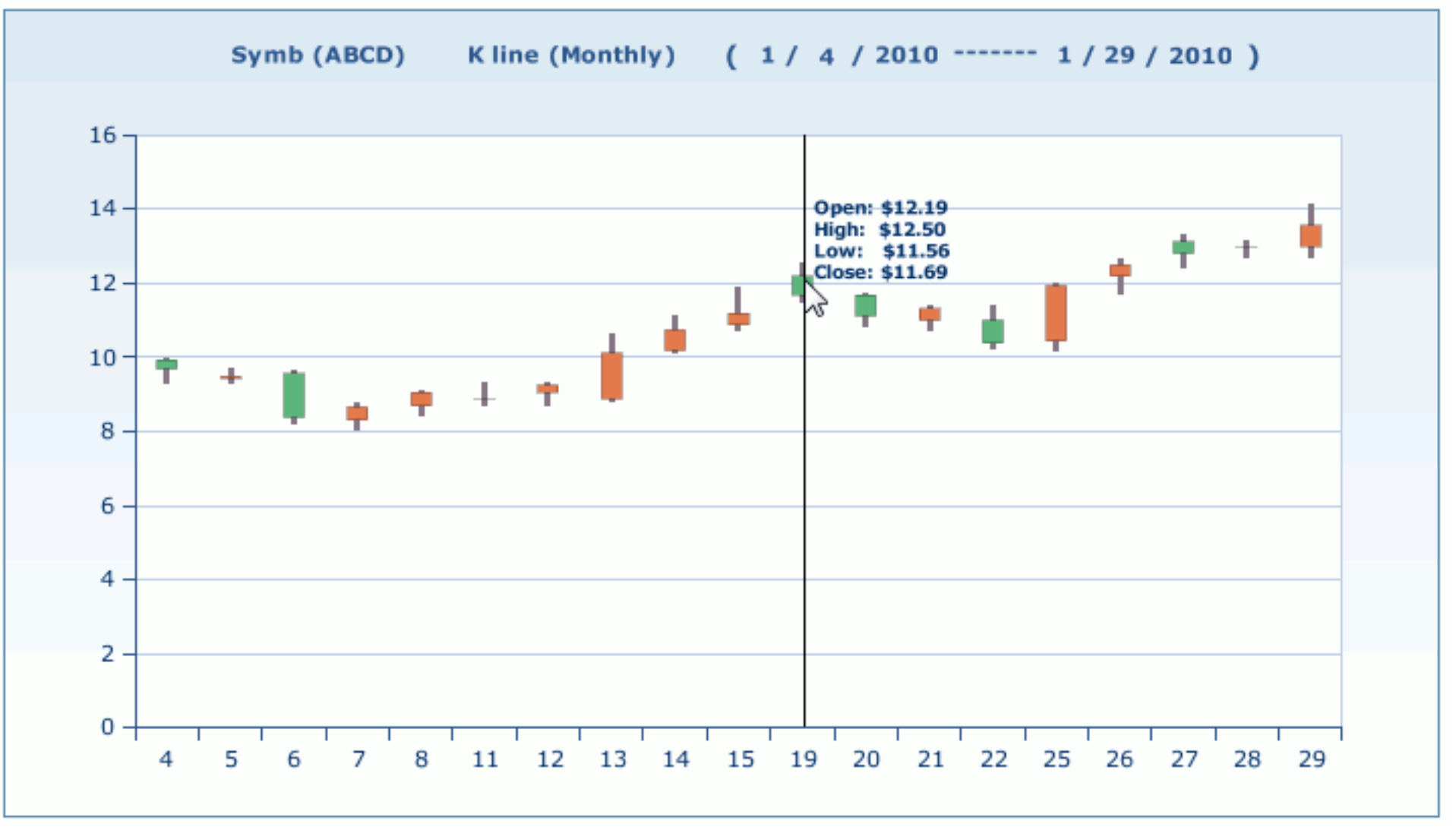
Jinfonet Gourmet Java
88 Main Street, Suite 123 Potomac, MD 98765
Phone: (301) 555-4567
Fax: (301) 555-4321

Report sample 2: Stock chart

End users can use this report to track performance indices over time. By using the mouse over, end users can instantly view specific metrics for a designated point in time:

Stock Market

Date: 4/14/2014



Report sample 3: Shipment details report

By using web controls in the report, end users can dynamically change the results of the report.

Shipment Details by Customer

Order ID: No Sort [Submit]
Territory: Africa [Submit]

Date: 4/14/2014

Territory: Africa

Order ID	Order Date	Ship Date	Ship Via	Shipped	Payment Received	Shipping Cost
Customer Name:	**Java Blend**					
3007	1/12/2010	1/13/2010	United Packing	No	Yes	$100.00
3365	9/2/2011	9/3/2011	Federal Shipping	No	Yes	$100.00
3382	10/4/2011	10/5/2011	Federal Shipping	No	No	$100.00
3400	11/2/2011	11/3/2011	Federal Shipping	Yes	No	$100.00
3417	12/4/2011	12/5/2011	Federal Shipping	Yes	Yes	$100.00
Customer Name:	**The Java Roaster**					
3298	5/20/2011	5/21/2011	Express Delivery	No	Yes	$150.00
3315	6/6/2011	6/7/2011	Express Delivery	No	No	$200.00
3333	7/8/2011	7/9/2011	Express Delivery	No	Yes	$150.00
3350	8/6/2011	8/7/2011	Express Delivery	No	Yes	$200.00
3368	9/8/2011	9/9/2011	Express Delivery	No	No	$150.00
Total Shipping Cost						**$1,350.00**

To see more sample reports, visit http://www.jinfonet.com/products/demos/overview.htm.

Setting up JReport on your system

The *JReport Tutorial* assumes that you have downloaded and installed both JReport Designer and JReport Server on a Windows 7 machine.

The *JReport Tutorial* also assumes that you are performing the lessons on a Windows 7 machine and therefore operating system specific commands and screen shots reflect this. However JReport can run on multiple platforms.

JReport Designer has the following system requirements:

JReport Designer System Requirements		
	Recommended Requirements	**Minimum Requirements**
OS:	Windows x64, Unix x64 and Linux x64	Windows, Unix, Linux
CPU:	Intel Core 2	Intel P4 Xeon 3.0GHz
Free Memory:	4 GB	2 GB
Free Disk:	2 GB	1 GB
JDK:	6 or above	6 or above
Recommended Browsers		
Internet Explorer 9 and higher versions, Firefox 20 and higher versions, Google Chrome 23 and higher versions.		

JReport Server has the following system requirements:

JReport Server System Requirements		
	Recommended Requirements	**Minimum Requirements**
OS:	Windows x64, Unix x64, Linux x64, z/Linux64	Windows, Unix, Linux, z/Linux
CPU:	Intel Core 2	Intel P4 Xeon 3.0GHz
Free Memory:	8 GB	2 GB
Free Disk:	10 GB	1 GB
JDK:	6 or above	6 or above
Recommended Browsers		
Internet Explorer 9 and higher versions, Firefox 20 and higher versions, Google Chrome 23 and higher versions.		

You can download a JDK version at http://www.oracle.com/technetwork/java/javase/downloads/index.html.

Notes:

- JReport supports Java VMs from Sun and IBM. You can try using other Java VMs, but their compatibility cannot be guaranteed. Jinfonet technical support welcomes any feedback on the use of other Java VMs.
- If you want to use JReport Server on a z/Linux system, you must download the JDK specially used for IBM from http://www-03.ibm.com/servers/eserver/zseries/software/java/ and the version should be at least V6.
- You are not recommended to run JReport in the Internet Explorer Compatibility View mode.

Part II: Quick Start

In this part, we will introduce the most powerful yet easy-to-use web tools of JReport.

JDashboard enables displaying multiple predefined library components created from different data sources in one dashboard freely and conveniently. Web report components and Visual Analysis components can also be inserted in dashboards.

Visual Analysis is simply used for analyzing data step by step visually by combining text or graphics display type and color/size/pie slice/shape customization.

Web Report Studio aims at faster and simpler design and creation of multi-component web reports, easier customization, and better presentation style.

Jump to the corresponding tracks to view detailed lessons:

- Track 1: Self-service dashboard with JReport
- Track 2: Performing visual analysis
- Track 3: Creating and analyzing ad hoc reports

Before starting this part, make sure your JReport Server is launched and the Server Console page is opened (for details, refer to Starting JReport Server).

Click the right arrow (⇨) below to begin reading the part.

Track 1: Self-service dashboard with JReport

JDashboard is a new way of information delivery, using a user portal user interface rather than a web report or page report. Users can create, edit and browse dashboards from the JReport Console using JDashboard. With pre-built library components, 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, etc. A dashboard can hold multiple library components so that when browsing the dashboard users are able to see multiple data aspects. Within a dashboard, library 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.

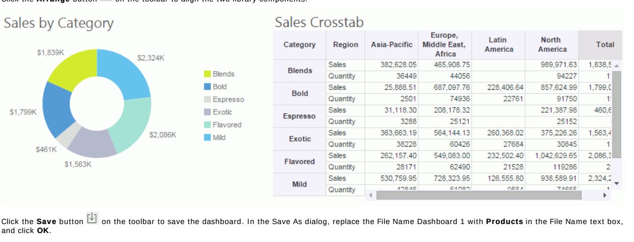
- Task 1: Create a dashboard and insert library components
- Task 2: Clicking on a pie chart legend will automatically update the crosstab (delivering messages between components)
- Task 3: Insert a report component into dashboard
- Task 4: Use a slider to filter on quantity
- Task 5: Insert a third party gadget (stock widget)
- Task 6: Export the library components
- Task 7: Use the configuration panel to change parameters
- Task 8: Share parameters among components
- Task 9: Insert a Visual Analysis component into dashboard
- Task 10: Run a report in JDashboard
- Task 11: Set JDashboard as the server home page

Note: A JDashboard license for JReport Server is required in order to perform this track. If you do not have the licenses, please contact your Jinfonet Software account manager to obtain it first.

Task 1: Create a dashboard and insert library components

1. On the JReport Console > Resources page, click **New** > **Dashboard** on the task bar. JDashboard window will be displayed and a blank dashboard created.

2. Click on the toolbar to display the Resources panel which lists all the data resources and components that can be inserted in dashboards, then go to **Component Library** > **Public Components** > **SampleReports** and drag **Sales by Category.lc** to the dashboard body.
3. Drag **Crosstab.lc** in the same folder to the right of the first component.
4. Click on the top right of the Resources panel to hide the panel.
5. Click the **Arrange** button on the toolbar to align the two library components.

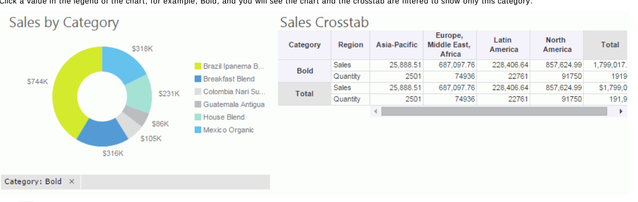


6. Click the **Save** button on the toolbar to save the dashboard. In the Save As dialog, replace the File Name Dashboard 1 with **Products** in the File Name text box, and click **OK**.

Task 2: Clicking on a pie chart legend will automatically update the crosstab (delivering messages between components)

This task shows the case of sending a filter message from one library component to another.

1. Click a value in the legend of the chart, for example, Bold, and you will see the chart and the crosstab are filtered to show only this category.

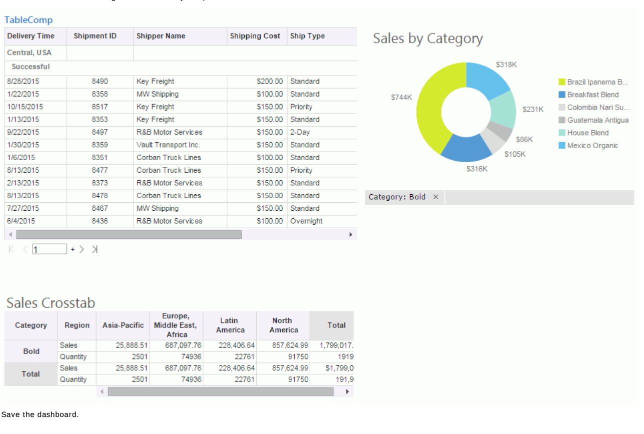


2. Click on the toolbar to save the dashboard.

Task 3: Insert a report component into dashboard

Report components created from business views can be inserted in dashboards directly.

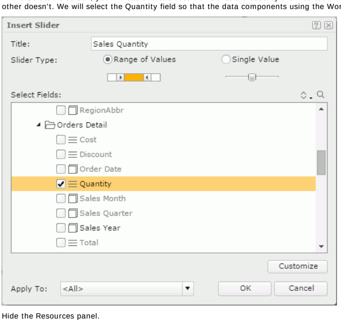
1. Click on the toolbar to display the Resources panel, then go to **Reports** > **Public Reports** > **SampleReports**, expand **ShipmentStatus.wls** and drag **TableComp** to the dashboard body below the existing components. Hide the Resources panel.
2. Click on the toolbar to align the three library components.



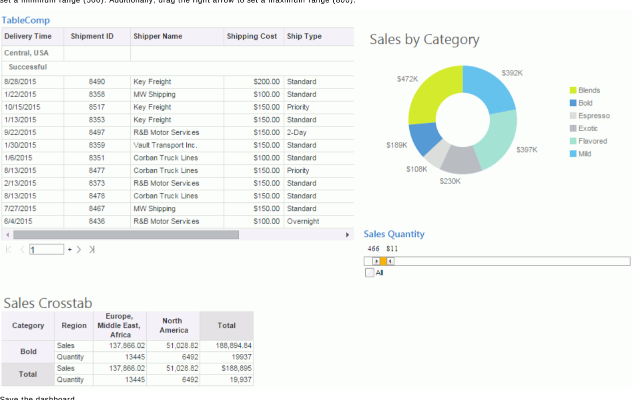
3. Save the dashboard.

Task 4: Use a slider to filter on quantity

1. Click on the toolbar to display the Resources panel. From the Toolbox node, drag **Slider** to the dashboard body below the chart.
2. In the Insert Slider dialog, input **Sales Quantity** in the Title text box. Choose **Range of Values** as the slider type.
3. In the Select Fields box, you will see two business views that are used by the inserted three data components. The WorldWideSalesBV has the Quantity field while the other doesn't. We will select the Quantity field so that the data components using the WorldWideSalesBV will be able to respond to the slider. Then click **OK**.



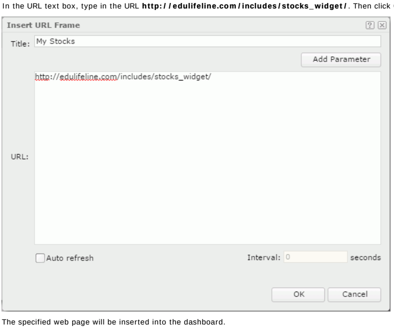
4. Hide the Resources panel.
5. In order to filter a range of records using a slider component (500-800), on the slider component, remove the checkmark from the **All** checkbox. Drag the left arrow to set a minimum range (500). Additionally, drag the right arrow to set a maximum range (800).



6. Save the dashboard.

Task 5: Insert a third party gadget (stock widget)

1. Click on the toolbar to display the Resources panel, scroll down and drag a **URL Frame** component from Toolbox to the dashboard body below the slider.
2. The Insert URL Frame dialog is displayed. In the Title text field, input **My Stocks**.
3. In the URL text box, type in the URL **http://edulifeline.com/includes/stocks_widget/**. Then click **OK**.



4. The specified web page will be inserted into the dashboard.

5. Hide the Resources panel and save the dashboard.

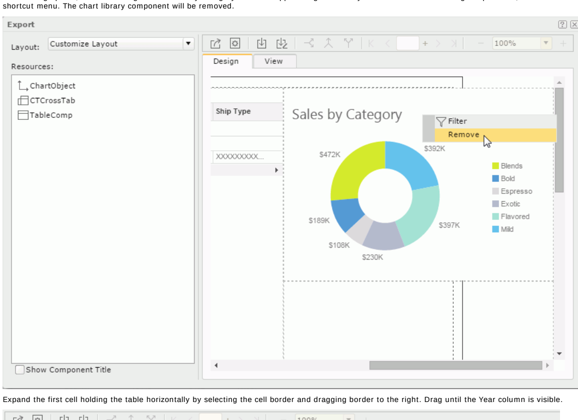
Task 6: Export the library components

1. Click the **Export** button on the toolbar to access the Export dialog. Then select **Customize Layout** from the Layout drop-down list.

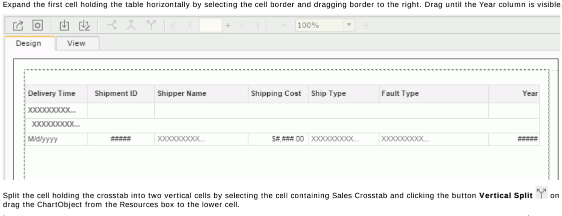
   By default, all the exportable library components are arranged using a tabular style according to their positions in the dashboard in the design panel (right hand side). Each tabular cell can hold no more than one component. There are two blank cells to the right of the crosstab where the slider and the gadget reside. Sliders and gadgets cannot be exported so they are not available here.

   We will remove the chart on the right of the table and add it below the crosstab.

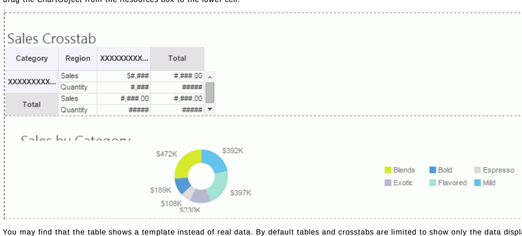
2. In the design panel, scroll to the right. The Sales by Category chart will appear. Right-click anywhere in the cell containing the pie chart, select **Remove** from the shortcut menu. The chart library component will be removed.



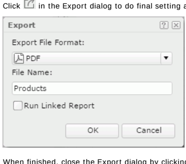
3. Expand the first cell holding the table horizontally by selecting the cell border and dragging border to the right. Drag until the Year column is visible.



4. Split the cell holding the crosstab into two vertical cells by selecting the cell containing Sales Crosstab and clicking the button **Vertical Split** on the toolbar, then drag the ChartObject from the Resources box to the lower cell.



5. You may find that the table shows a template instead of real data. By default tables and crosstabs are limited to show only the data displayed in the dashboard. If you want the full table data to be exported, right-click in the cell holding the table, select **Filter** from the shortcut menu, click the **All** filter, and click **OK**.
6. Click the **View** tab to preview the layout.
7. You can save the layout if you are going to reuse it. Click on the toolbar, in the Save As dialog, type **Layout1** in the Name text box and click **OK**. You will see that Layout1 is added in the Layout drop-down list and selected as the value.
8. Click in the Export dialog to do final setting and then start exporting. In the Export dialog, click **OK**. The exporting process will begin.

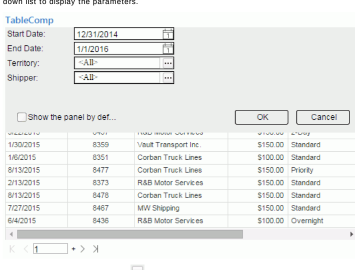


9. When finished, close the Export dialog by clicking in the upper right corner.

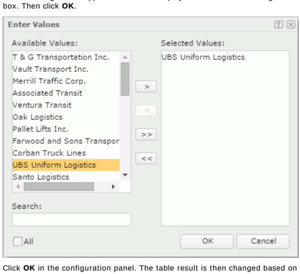
Task 7: Use the configuration panel to change parameters

When a component uses parameters, you can make use of the configuration panel to change the parameter values.

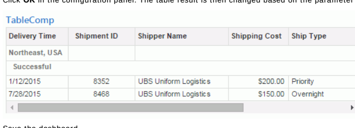
1. On the table, TableComp, place your mouse anywhere on the title bar, click the **Options** button that appears on the title bar, select **Edit Setting** from the drop-down list to display the parameters.



2. We will change the Shipper. Click to display the Enter Values dialog. Uncheck the All option. Select **UBS Uniform Logistics** and click to add it to the right box. Then click **OK**.



3. Click **OK** in the configuration panel. The table result is then changed based on the parameter value.



4. Save the dashboard.

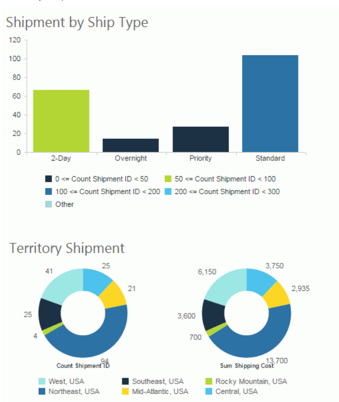
Task 8: Share parameters among components

When two or more library components in a dashboard contain parameters that meet the following cases, the parameters can be shared among components. After sharing parameters, we just need to provide values to one group of the parameters and all related components will be able to receive them.

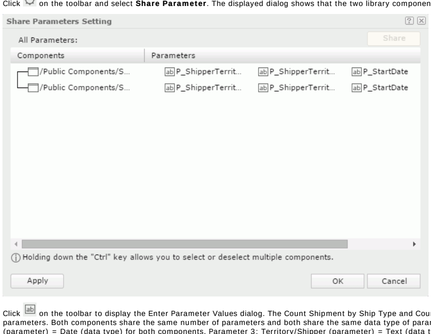
- The numbers of the parameters in each library component are the same.
- According to the parameter order in each library component, the orders of the parameter data types are the same. For example, the parameter data types in a component are String, Number, and Boolean. If there is another component in which the parameter data types are also String, Number, and Boolean, the two components fulfill the condition of the same parameter data types.
- It is up to users to make sure the to-be-shared parameters contain some common values.

The following example illustrates two components sharing similar parameters.

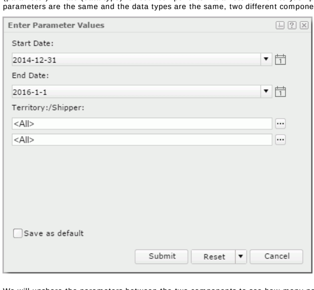
1. Click on the dashboard title bar to add a new dashboard. A new tab will be created, labeled **Dashboard 1**.
2. Click on the toolbar to display the Resources panel, expand **Component Library** > **Public Components** > **SampleReports** and drag **Count Shipment by Ship Type.lc** and **Count Shipment by Territory.lc** one by one to the dashboard body. Hide the Resources panel. Click the **Arrange** button on the toolbar to align the two library components.



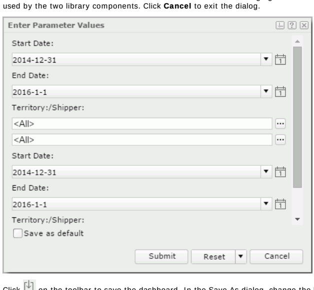
3. Click on the toolbar and select **Share Parameter**. The displayed dialog shows that the two library components have shared parameters. Then click **OK**.



4. Click on the toolbar to display the Enter Parameter Values dialog. The Count Shipment by Ship Type and Count Shipment by Territory components match parameters. Both components share the same number of parameters and both share the same data type of parameters. Parameter 1 and 2: Start Date/End Date (parameter) = Date (data type) for both components. Parameter 3: Territory/Shipper (parameter) = Text (data type) for both components. If the number of parameters are the same and the data types are the same, two different components can share the same parameters. Exit the Enter Parameter Values dialog.



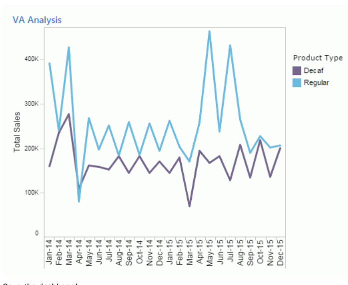
5. We will unshare the parameters between the two components to see how many parameters we need to specify. Click on the toolbar and select **Share Parameter**. In the displayed dialog, select either component and click the **Cancel Share** button. Then click **OK**.
6. Click on the toolbar to access the Enter Parameter Values dialog again. Now it lists separate pairs of the Start Date, End Date, and Territory/Shipper parameters used by the two library components. Click **Cancel** to exit the dialog.



7. Click on the toolbar to save the dashboard. In the Save As dialog, change the File Name from Dashboard 1 to **Shipment** and click **OK**.

Task 9: Insert a Visual Analysis component into dashboard

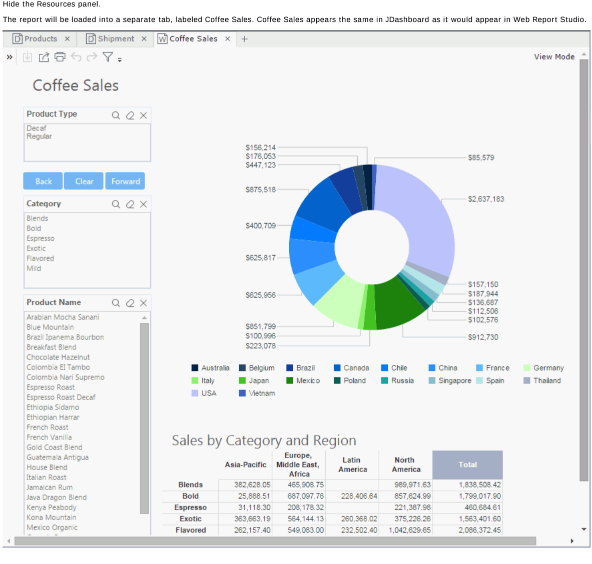
1. Keep the Shipment dashboard active.
2. Click on the toolbar to display the Resources panel, then go to **Reports** > **Public Reports** > **SampleReports**, expand **VA Analysis.va** and you will find a VCTObject under it. Drag the VCTObject into the dashboard body.



3. Save the dashboard.

Task 10: Run a report in JDashboard

1. Keep the Shipment dashboard active.
2. Click on the toolbar to display the Resources panel, then go to **Reports** > **Public Reports** > **SampleReports** and drag **Coffee Sales.wls** to the dashboard body. Hide the Resources panel.
3. The report will be loaded into a separate tab, labeled Coffee Sales. Coffee Sales appears the same in JDashboard as it would appear in Web Report Studio.



Task 11: Set JDashboard as the server home page

Keep the JDashboard window open and we will continue from the task 10. There is currently a browser session open from JReport Server console and a popup window containing JDashboard.

1. In your browser, change from the JDashboard popup or floating window back to the JReport Server Console (ALT + tab in your browser window).
2. On the JReport Console page, click **Profile** on the system toolbar, then **Customize Server Preferences** on the task bar. In the General tab check the option **Use JDashboard as Server User Console Home Page**, then click **OK**. Click **OK** in the prompt message.
3. Click **Resources** on the system toolbar. In the Name column, click **My Reports**. Click **Shipment.dsh** in the folder to open the dashboard which will be loaded into a new JDashboard window so that the setting in step 1 can take effect in JDashboard.
4. In the new JDashboard window, click on the toolbar and you will see Set as Server Home is enabled on the menu list. Click the option to set JDashboard with the Shipment dashboard active as the server home page.
5. Refresh the JReport Console page and a Home tab will be available beside the Resources tab on the system toolbar. Click the Home tab and you can access the Shipment dashboard immediately.

Track 1 summary

In this track, we first created a dashboard by inserting two library components, a report component, a slider, a third party gadget, and a Visual Analysis component, and performed operations in the dashboard such as delivering the filtering message predefined between two library components, filtering the data using the slider, exporting the library components, changing parameter values via the configuration panel, sharing component parameters, setting JDashboard as the server home for easier access, and running a report in JDashboard.

Track 2: Performing visual analysis

Visual Analytics is a WYSIWYG product to visualize the result of every step of your work. Simply by dragging and dropping data fields onto a layout module, you are able to visually create crosstabs and charts step by step.

Business views are the data sources used in Visual Analytics.

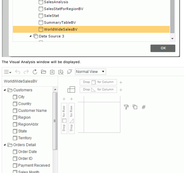
- Task 1.: Select a business view.
- Task 2.: Adding data fields.
- Task 3.: Saving the template.
- Task 4.: Filtering the data.

Note: A Visual Analytics license is required in order to perform this track. If you do not have the license, please contact your Jinfonet Software account manager to obtain one first.

Task 1: Select a business view

1. On the **Jinfonet > Resource page**, click **New > Analysis** on the task bar.
2. The Select Data Source dialog lists all catalogs in the current folder and the business views in the catalogs. Click **Public Reports > SampleReport.cat > Data Source 3 > WorldWideDatabase01** then click **OK**.

3. The Visual Analysis window will be displayed.

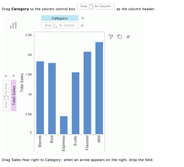


Task 2: Adding data fields

This section demonstrates how to drag data fields to the data presentation area and how to use the legend buttons.

1. In the Visual Analysis, click the **Display Type** button and then select Bar icon from the drop-down list.
2. Drag **Total Sales** from the resources panel on the left to the row control box. Drop Here Item The Total Sales field is used to draw the axis in the row header.

3. Drag **Category** in the column control box Drop Here Item as the column header.

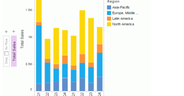


4. Drag Sales Year right to Category: when an arrow appears on the right, drop the field:

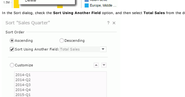
5. Drag Sales Year to Category: when an arrow appears on the right, drop the field.

You can also make use of the button to add a field from its drop-down resource list.

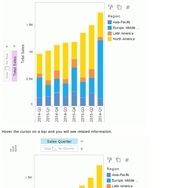
6. To replace Sales Year with Sales Quarter, drag Sales Quarter over Sales Year. An arrow will appear above Sales Year (in the center), then drop Sales Quarter.



6. To Remove Category, click on it and select **Delete** from the drop-down menu. Or you can simply drag the Category label out of its position to remove it.



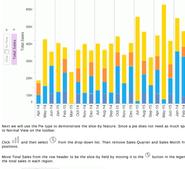
7. To mark different regions by color, drag **Region** to the Color button in the legend section.



8. We will sort sales quarters. Click Sales Quarter and select **Sort**.

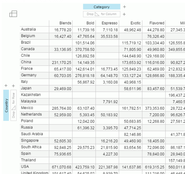
9. In the Sort dialog, check the **Sort Using Another Field** option, and then select **Total Sales** from the drop-down list. Click **OK**.

10. Here is the sort result.

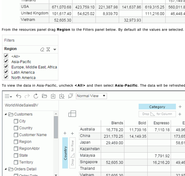


11. Hover the cursor on a bar and you will see related information.

12. Drag Sales Month after Sales Quarter.



13. The scrollbar shows and we cannot see the whole chart. Widen the presentation area and select **Fit Visible** from the drop-down list of Normal View on the toolbar so that the bar is fully displayed according to the current space.



14. Next we will use the Pie type to demonstrate the slice-by feature. Since a pie does not need as much space as the above bar, we will change Fit Visible back to Normal View on the toolbar.

15. Click and then select from the drop-down list. Then remove Sales Quarter and Sales Month from the column header by dragging them out of their positions.

16. Move Total Sales from the row header to the slice-by field by moving it to the Slice-by button in the legend section on the right. Slice by decides the angle for the total sales in each region.

Task 3: Saving the template

The current data source will be saved as a template. Click on the toolbar. In the Save As dialog, leave the save location be the My Reports folder, type in **Productsales** in the File Name text box, and click **OK**. Next time we will able to open the template in the My Reports folder in the server resource tree.

Task 4: Filtering the data

We will use the Pivot display type to show the filter feature.

1. Click and then select **New** from the drop-down menu to start a new Visual Analysis session.
2. In the Select Data Source dialog, select **WorldWideDatabaseBIr** and click **OK**.
3. From the resources panel, drag **Country** to the row header. Drop Here Item **Category** to the column header. Drop Here Item, and **Total Sales** to the button in the legend section.

4. Filter the resource panel drag **Region** to the Filters panel below. By default all the values are selected.

5. To view the data in Asia-Pacific, uncheck **<All>** and then select **Asia-Pacific**. The data will be refreshed to show the countries in Asia-Pacific.

6. Save the template as **Sales-test** in the My Reports folder.

Track 2 summary

In this track, we used Visual Analysis to add data dimensions step by step visually, saved the current data status as a template and filtered the data.

Track 3: Creating and analyzing ad hoc reports

JReport features a powerful ad hoc reporting capability, which allows end users and business analysts to create and manipulate ad hoc reports in the JReport Server-powered runtime environment.

There are two types of ad hoc reports in JReport, Web Report and Page Report. Both are created from a business view in a catalog. Business views have the capability of providing multiple hierarchies allowing for automatic drill down and drill up for detailed analysis and slicing and dicing data which is especially powerful using a crosstab component.

- Using Web Report Wizard the user can easily in a single pass create a multi-component web report which is viewed in Web Report Studio. Web Report Studio provides the user a very high quality Web 2.0 Rich Internet Application (RIA) view of the report with component level navigation commands. For example, a single pass through the wizard can create a tabular style report with a chart and crosstab followed by the detail information in a table.
- Using Page Report Wizard the user can create a single component page report which is viewed in Page Report Studio. Page Report Studio provides the user a page mode view of the report very similar to a printed report.
- Lesson 1: Creating and working with web reports
- Lesson 2: Creating and performing data analysis on page reports

Note: A JReport Live license for JReport Server is required in order to perform this track. If you do not have the license, please contact your Jinfonet Software account manager to obtain one first.

Click the right arrow ( ) below to begin reading this track.

Lesson 1: Creating and working with web reports

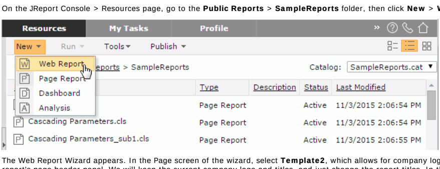
End users can create web reports on JReport Server if a business view is available. In this lesson, we will create a web report as an end user. The report should contain a table, a crosstab, and a chart.

This lesson contains the following tasks:

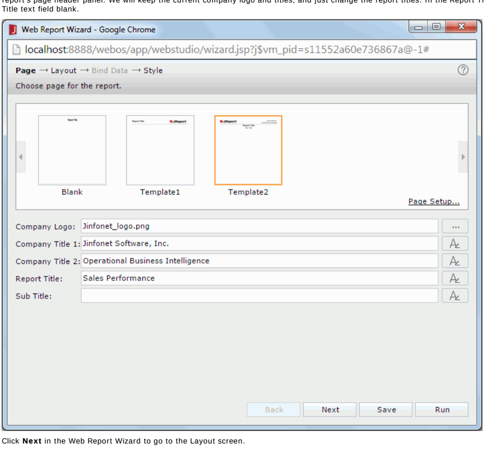
- Task 1: Create a web report
- Task 2: Operate on the crosstab
- Task 3: Perform on the chart
- Task 4: Work with the table
- Task 5: Apply a filter

Task 1: Create a web report

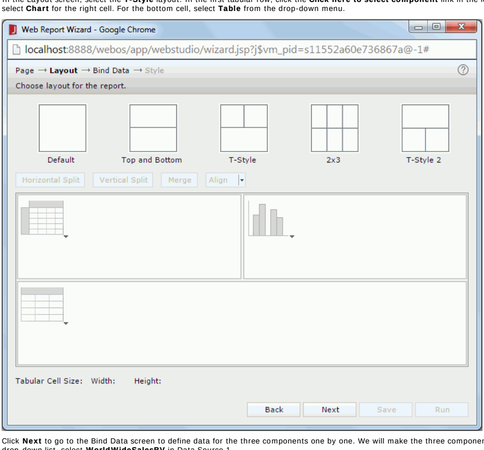
1. On the JReport Console > Resources page, go to the **Public Reports** > **SampleReports** folder, then click **New** > **Web Report** on the task bar.



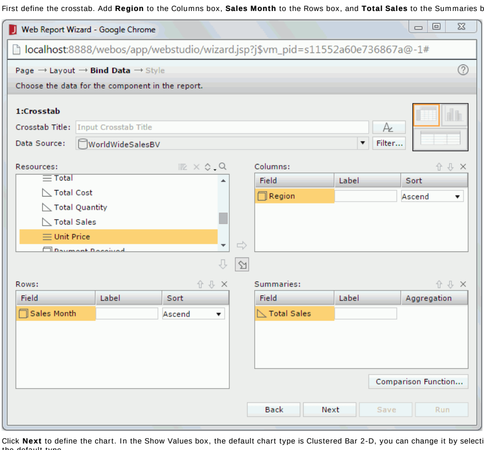
2. The Web Report Wizard appears. In the Page screen of the wizard, select **Template2**, which allows for company log, company name, report titles to be defined and added in the report's page header panel. We will keep the current company logo and titles, and just change the report titles. In the Report Title text field, input **Sales Performance**. Leave the Sub Title text field blank.



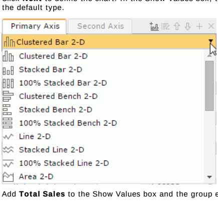
3. Click **Next** in the Web Report Wizard to go to the Layout screen.
4. In the Layout screen, select the **T-Style** layout. In the first tabular row, click the **Click here to select component** link in the left cell, select **Crosstab** from the drop-down menu, select **Chart** for the right cell. For the bottom cell, select **Table** from the drop-down menu.



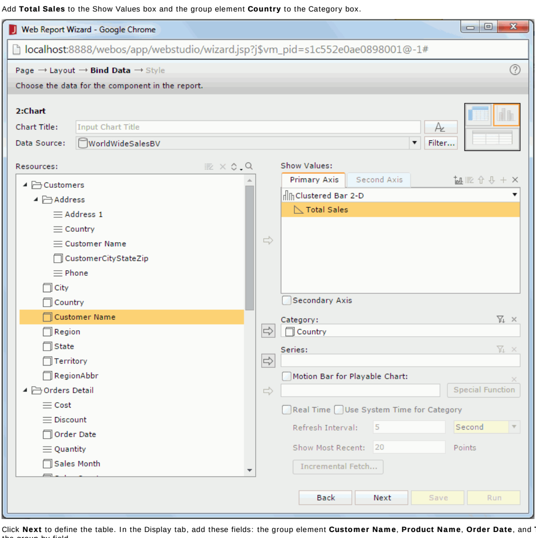
5. Click **Next** to go to the Bind Data screen to define data for the three components one by one. We will make the three components use the same business view. From the Data Source drop-down list, select **WorldWideSalesBV** in Data Source 1.
6. First define the crosstab. Add **Region** to the Columns box, **Sales Month** to the Rows box, and **Total Sales** to the Summaries box.



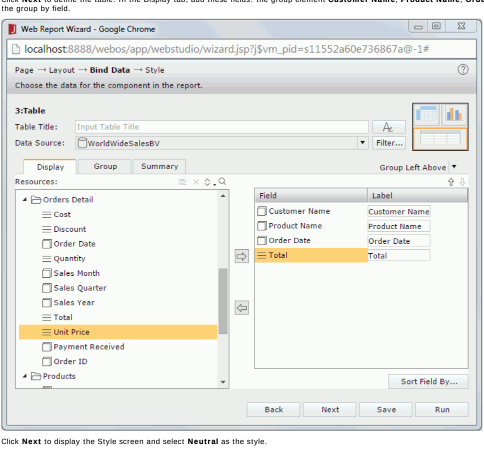
7. Click **Next** to define the chart. In the Show Values box, the default chart type is Clustered Bar 2-D, you can change it by selecting another type from the drop-down list. Here we use the default type.



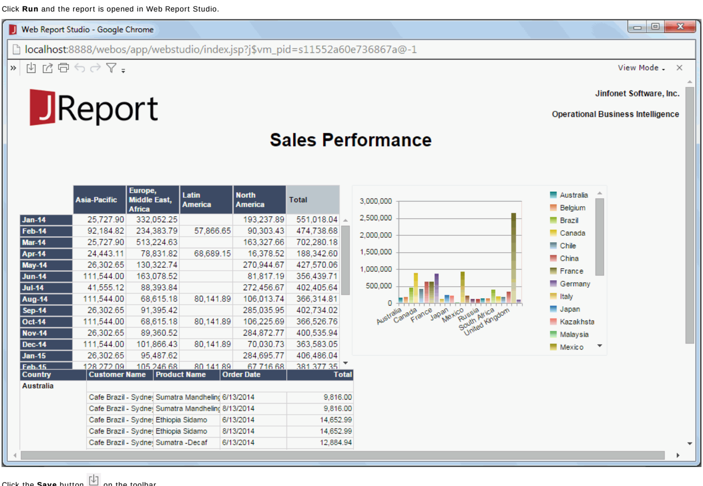
Add **Total Sales** to the Show Values box and the group element **Country** to the Category box.



8. Click **Next** to define the table. In the Display tab, add these fields: the group element **Customer Name**, **Product Name**, **Order Date**, and **Total**. In the Group tab, add **Country** as the group by field.



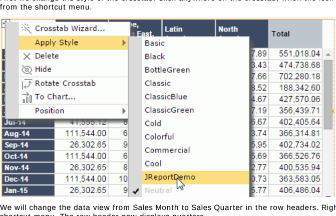
9. Click **Next** to display the Style screen and select **Neutral** as the style.
10. Click **Run** and the report is opened in Web Report Studio.



11. Click the **Save** button on the toolbar.
12. In the Save As dialog, type **Sales Performance** in the File Name text field, then click **OK** to save the web report. In the prompt message window, click **OK**.

Task 2: Operate on the crosstab

1. By default it is view mode for the report. To analyze the report data, select **Edit Mode** from the mode drop-down menu on the toolbar.
2. First we change the style of the crosstab. Click anywhere on the crosstab, when the icon appears at its upper top left, right-click the icon, then select **Apply Style** > **JReportDemo** from the shortcut menu.



3. We will change the data view from Sales Month to Sales Quarter in the row headers. Right-click the row header which displays months, select **Switch Row** > **Sales Quarter** from the shortcut menu. The row header now displays quarters.

	Asia-Pacific	Europe, Middle East, Africa	Latin America	North America	Total
2014-Q1	143,640.62	1,079,660.66	57,866.65	446,868.98	1,728,036.90
2014-Q2	162,289.76	372,233.09	68,689.15	369,140.38	972,352.37
2014-Q3	179,401.76	248,404.44	80,141.89	663,506.36	1,171,454.46
2014-Q4	249,390.64	259,842.12	160,283.79	461,129.19	1,130,645.74
2015-Q1	180,877.39	222,638.52	80,141.89	542,771.12	1,026,428.93
2015-Q2	249,390.64	281,783.59	160,283.79	808,605.86	1,500,063.88
2015-Q3	161,424.84	291,597.76	80,141.89	821,341.28	1,354,505.78
2015-Q4	269,799.74	446,575.72	160,283.79	312,067.26	1,188,726.51
Total	1,596,215.39	3,202,735.92	847,832.86	4,425,430.42	10,072,214.59

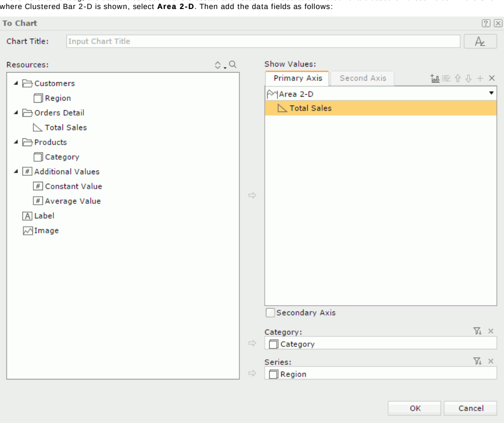
4. To further view information about the product category in the first quarter of 2014, right-click **2014-Q1** in the row header, select **Go to by Value** > **Category** from the shortcut menu. The crosstab now shows the total sales of the product categories in the first quarter of 2014 in different regions.

	Asia-Pacific	Europe, Middle East, Africa	Latin America	North America	Total
Blends	51,455.80	112,572.60		127,182.87	291,211.28
Bold	25,888.51	285,134.36		67,353.14	378,376.00
Espresso	7,110.18	76,762.47		37,562.40	121,435.06
Exotic	33,989.72	118,181.03		30,141.52	182,312.28
Flavored	25,196.40	165,596.40		109,302.40	300,095.20
Mild		321,413.80	57,866.65	75,326.64	454,607.09
Total	143,640.62	1,079,660.66	57,866.65	446,868.98	1,728,036.90

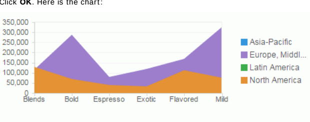
5. The row and column headers can be switched. Click anywhere on the crosstab, when the icon appears at its upper top left, right-click the icon, then select **Rotate Crosstab** from the shortcut menu.

	Blends	Bold	Espresso	Exotic	Flavored	Mi
Asia-Pacific	51,455.80	25,888.51	7,110.18	33,989.72	25,196.40	
Europe, Middle East, Africa	112,572.60	285,134.36	76,762.47	118,181.03	165,596.40	
Latin America						
North America	127,182.87	67,353.14	37,562.40	30,141.52	109,302.40	
Total	291,211.28	378,376.00	121,435.06	182,312.28	300,095.20	

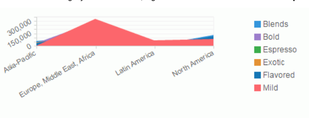
6. Click on the toolbar to undo the rotation.
7. We will convert the crosstab to a chart. Click anywhere on the crosstab, when the icon appears at its upper top left, right-click the icon, then select **To Chart** from the shortcut menu.
8. In the To Chart dialog, the data fields used in the crosstab are listed. We need to define our chart based on these fields. In the Show Values box, from the chart type drop-down list where Clustered Bar 2-D is shown, select **Area 2-D**. Then add the data fields as follows:



9. Click **OK**. Here is the chart:



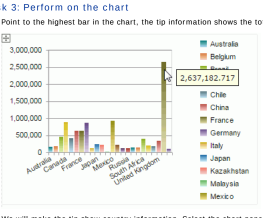
10. To switch the category and the series, right-click on the chart and select **Swap Chart Groups**.



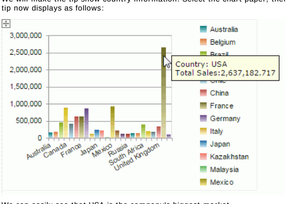
11. Click on the toolbar to save the report.

Task 3: Perform on the chart

1. Point to the highest bar in the chart, the tip information shows the total sales. But it does not tell which country it is since it is difficult to figure out simply by the color.



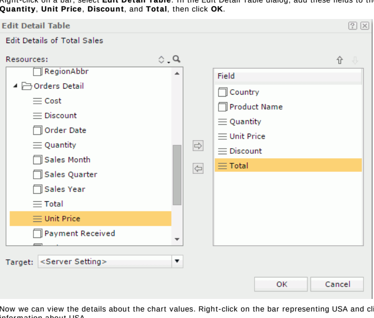
2. We will make the tip show country information. Select the chart paper, then go to the Inspector panel, set the property **Show Category and Series** to **true** in the Hint category. The tip now displays as follows:



We can easily see that USA is the company's biggest market.

We will link the values to a table which contains detailed information about the values.

6. Right-click on a bar, select **Edit Detail Table**. In the Edit Detail Table dialog, add these fields to the right panel concerning the total sales in the countries: **Country**, **Product Name**, **Quantity**, **Unit Price**, **Discount**, and **Total**, then click **OK**.



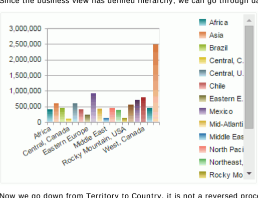
7. Now we can view the details about the chart values. Right-click on the bar representing USA and click **Go to Detail** on the shortcut menu. The following table will be displayed showing information about USA.

Detail Table - Total Sales group by Country (USA)

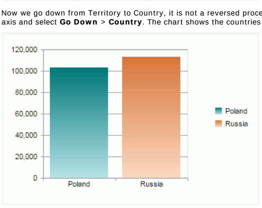
Country	Product Name	Quantity	Unit Price	Discount	Total
USA	Java Dragon Blend	375	13.45	.00	5,043.75
USA	Java Dragon Blend	375	13.45	.00	5,043.75
USA	Sulawesi	210	12.15	.00	2,551.50
USA	Sulawesi	210	12.15	.00	2,551.50
USA	Chocolate Hazelnut	157	9.00	.00	1,413.00
USA	Chocolate Hazelnut	157	9.00	.00	1,413.00
USA	Java Dragon Blend	457	13.45	.00	6,146.65
USA	Chocolate Hazelnut	192	9.00	10.00	1,555.20
USA	Blue Mountain	192	9.95	10.00	1,719.36
USA	Jamaican Rum	256	9.00	10.00	2,073.60
USA	Gold Coast Blend	213	11.45	10.00	2,194.96
USA	Java Dragon Blend	375	13.45	.00	5,043.75
USA	Java Dragon Blend	375	13.45	.00	5,043.75
USA	Java Dragon Blend	375	13.45	.00	5,043.75
USA	Sulawesi	210	12.15	.00	2,551.50
USA	Sulawesi	210	12.15	.00	2,551.50
USA	Sulawesi	210	12.15	.00	2,551.50
USA	Chocolate Hazelnut	157	9.00	15.00	1,201.05
USA	Chocolate Hazelnut	157	9.00	.00	1,413.00
USA	Java Dragon Blend	375	13.45	.00	5,043.75
USA	Java Dragon Blend	375	13.45	.00	5,043.75
USA	Java Dragon Blend	375	13.45	.00	5,043.75
USA	Sulawesi	210	12.15	.00	2,551.50
USA	Sulawesi	210	12.15	.00	2,551.50
USA	Sulawesi	210	12.15	.00	2,551.50
USA	Chocolate Hazelnut	157	9.00	.00	1,413.00
USA	Chocolate Hazelnut	157	9.00	.00	1,413.00
USA	Chocolate Hazelnut	157	9.00	.00	1,413.00
USA	Java Dragon Blend	375	13.45	.00	5,043.75
USA	Sulawesi	210	12.15	.00	2,551.50
USA	Chocolate Hazelnut	157	9.00	.00	1,413.00
USA	Torrefazione Italia	533	12.00	20.00	5,116.80
USA	Blue Mountain	1280	9.95	.00	12,736.00

1 28

8. Click on the toolbar to return to the main report.
9. Since the business view has defined hierarchy, we can go through data upward and downward directly. Right-click any country name on the X axis and select **Go Up** > **Territory**.



10. Now we go down from Territory to Country, it is not a reversed process of going up from Country to Territory, but applying an additional filter. Right-click **Eastern Europe** on the X axis and select **Go Down** > **Country**. The chart shows the countries in the Eastern Europe instead of all the countries.



11. Click on the toolbar to save the report.

Task 4: Work with the table

1. We first sort the Total column in descending order. Right-click any value in the Total column and then select **Sort** > **Descend** from the shortcut menu.

Country	Customer Name	Product Name	Order Date	Total
Australia				
	Cafe Brazil - Sydne	Ethiopia Sidamo	6/13/2014	14,652.99
	Cafe Brazil - Sydne	Ethiopia Sidamo	8/13/2014	14,652.99
	Cafe Brazil - Sydne	Sumatra -Decaf	6/13/2014	12,884.94
	Cafe Brazil - Sydne	Sumatra -Decaf	8/13/2014	12,884.94
	Cafe Brazil - Sydne	Mexico Organic	2/13/2014	11,739.16
	Cafe Brazil - Sydne	Torrefazione Italia	6/13/2014	11,048.40
	Cafe Brazil - Sydne	Torrefazione Italia	8/13/2014	11,048.40
	Cafe Brazil - Sydne	Rift Valley Blend	2/13/2014	10,083.55
	Cafe Brazil - Sydne	Sumatra Mandheling	6/13/2014	9,816.00
	Cafe Brazil - Sydne	Sumatra Mandheling	8/13/2014	9,816.00
	Cafe Brazil - Sydne	Tanzania Blackburn	2/13/2014	7,786.00
	Cafe Brazil - Sydne	Italian Roast	6/13/2014	7,367.25

2. Next we summarize the total sales for each country group and for all the countries.

Right-click any value in the Total column, then select **Aggregate On** from the shortcut menu. In the Aggregate On dialog, set Function to **Sum**, then click **OK**. You will see the total sales for each country group is added at the end of each group and the total sales for all the countries is added at the end of the table.

This operation also creates a dynamic aggregation which is given a default name Sum_Total. You can find it in the Dynamic Resource > Aggregations list in the Resources panel and you can use it again in the report.

The sum of total for Australia:

Country	Customer Name	Product Name	Order Date	Total
Australia				
	Cafe Brazil - Sydne	Italian Roast	8/13/2014	7,367.25
	Cafe Brazil - Sydne	Espresso Roast	2/13/2014	7,110.18
	Cafe Brazil - Sydne	Ethiopia Sidamo	4/13/2014	1,786.95
	Cafe Brazil - Sydne	Sumatra -Decaf	4/13/2014	1,575.50
	Cafe Brazil - Sydne	Torrefazione Italia	4/13/2014	1,350.00
	Cafe Brazil - Sydne	Sumatra Mandheling	4/13/2014	1,200.00
	Cafe Brazil - Sydne	Gold Coast Blend	4/13/2014	1,145.00
	Cafe Brazil - Sydne	Italian Roast	4/13/2014	898.70
				156,214.20
Belgium				
	Coffee of Belgium	Mexico Organic	7/21/2015	10,006.80
	Coffee of Belgium	Mexico Organic	5/21/2015	10,006.80

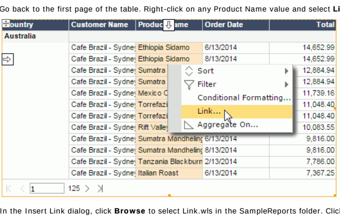
The sum of total for all the countries:

Country	Customer Name	Product Name	Order Date	Total
USA				
	Brazilean's Best	Arabian Mocha San	4/9/2014	133.24
	Kona Coffee House	Arabian Mocha San	4/5/2014	103.63
	International Coffee	Brazil Ipanema Bou	4/4/2014	58.28
				2,637,182.72
Vietnam				
	Java Brewers - Har	French Roast	5/27/2014	26,302.65
	Java Brewers - Har	French Roast	7/27/2014	26,302.65
	Java Brewers - Har	Ethiopia Sidamo	6/23/2014	16,486.96
	Java Brewers - Har	Ethiopia Sidamo	8/23/2014	16,486.96
				85,579.23
				10,072,214.59

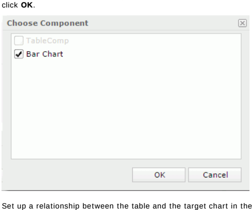
125 125

We will link the table to another report.

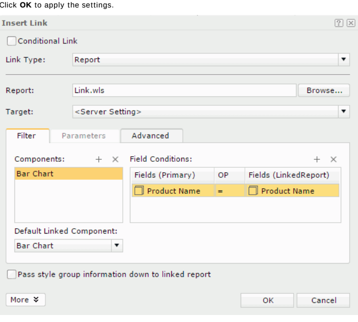
3. Go back to the first page of the table. Right-click on any Product Name value and select **Link** from the shortcut menu.



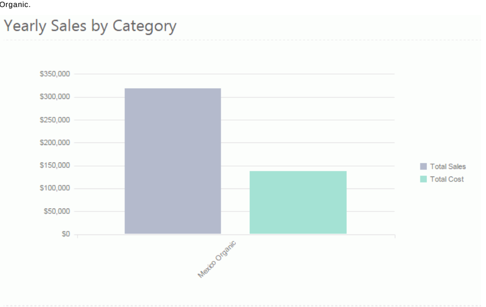
4. In the Insert Link dialog, click **Browse** to select Link.wls in the SampleReports folder. Click **OK**.
5. Click **More** to show more settings. In the Filter tab, click + above the Components box. In the Choose Component dialog, select **Bar Chart** in the linked report as the link target and click **OK**.



6. Set up a relationship between the table and the target chart in the Field Conditions box. Click the button + above the Field Conditions box to add a new condition row. Set the condition as Product Name = Product Name.
7. Click **OK** to apply the settings.



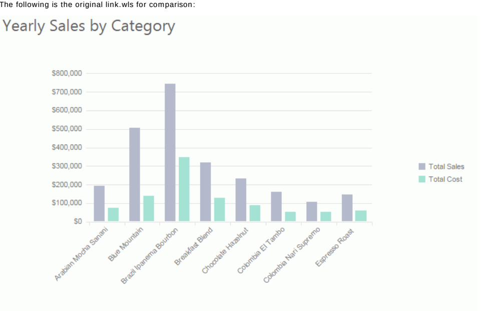
8. Switch to the View Mode. Click **Mexico Organic** in the Product Name column of the table. The link.wls report will be displayed with the chart filtered to show only data about Mexico Organic.



Order Detail

Product Name	Total Sales	Total Cost	Total Quantity
2014			
Arabian Mocha Sanani	$113,041.11	$44,096.00	8320
Blue Mountain	$119,258.71	$32,413.50	12005
Brazil Ipanema Bourbon	$554,319.57	$260,296.76	60817
Breakfast Blend	$131,432.86	$53,167.00	13460
Chocolate Hazelnut	$133,520.85	$50,542.96	16904
Colombia El Tambo	$87,924.00	$27,696.06	7327
Colombia Nari Supremo	$65,414.20	$31,650.85	8221
Espresso Roast	$84,989.52	$38,318.00	11270
Espresso Roast Decaf	$84,121.18	$47,949.20	9221

The following is the original link.wls for comparison:



Order Detail

Product Name	Total Sales	Total Cost	Total Quantity
2014			
Arabian Mocha Sanani	$113,041.11	$44,096.00	8320
Blue Mountain	$119,258.71	$32,413.50	12005
Brazil Ipanema Bourbon	$554,319.57	$260,296.76	60817
Breakfast Blend	$131,432.86	$53,167.00	13460
Chocolate Hazelnut	$133,520.85	$50,542.96	16904
Colombia El Tambo	$87,924.00	$27,696.06	7327
Colombia Nari Supremo	$65,414.20	$31,650.85	8221
Espresso Roast	$84,989.52	$38,318.00	11270
Espresso Roast Decaf	$84,121.18	$47,949.20	9221

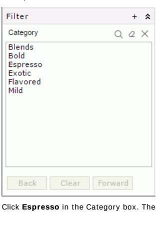
9. Click on the toolbar to return to the main report. Then click on the toolbar to save the report.

Task 5: Apply a filter

The three data components in the report use the same business view, therefore we can apply filters to them at the same time. There are two tools to use: the Filter panel and filter controls. For the usage of filter controls, you can learn from Creating a web report using JReport Designer. Here we focus on using the Filter panel to dynamically filter report data.

We would like to filter data with product categories, so we need to add the Category field to the Filter panel.

1. Switch to the Edit Mode. Click **+** on the title bar of the Filter panel on the left, then in the Select Field dialog, select **Category** and click **OK**. The Category field will be added in the Filter panel with its values.



2. Click **Espresso** in the Category box. The report comes out as follows:

	Employee Name	Payment Date	Amount
Nicolas	Lucas Nguyen	01/12/2023	11,000.00
	Lucas Nguyen	01/01/2024	10,000.00
Sophia	Sofia Alvarez	Organic Bananas	1,000.00
	Sofia Alvarez	Organic Bananas	1,000.00
	Sofia Alvarez	Organic Bananas	1,000.00
	Sofia Alvarez	Organic Bananas	1,000.00
	Sofia Alvarez	Organic Bananas	1,000.00
Nicolas	Tyler Jones	10/12/2023	1,000.00
	Tyler Jones	10/01/2024	1,000.00
	Tyler Jones	10/02/2024	1,000.00
	Tyler Jones	10/03/2024	1,000.00
	Tyler Jones	10/04/2024	1,000.00

1 - 15 of 100 Rows

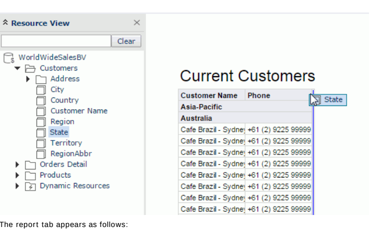
Show 15 rows per page to view the report.

LESSON 3: SUPPLIES

Lesson 3 covers the Lucas Nguyen supply chain terminology, costs, or overhead, which stems from the cost of all parts used in Mission Server, architectural blueprint services to the data processing, and other supplies. Second Row, the Lucas Nguyen Supply Chain Management Report, includes the Lucas Nguyen supply chain and the payment to Mission Company for Lucas Nguyen processing of Lucas Nguyen.

Lesson 2: Creating and performing data analysis on page objects

Data analysis is the process of examining data in order to find meaningful information. Data analysis can be performed on page objects. The Data Analysis pane is where you can create, edit, and run data analysis on page objects. Data analysis can be performed on page objects that are based on a table, query, or SQL statement. Data analysis can be performed on page objects that are based on a table, query, or SQL statement. Data analysis can be performed on page objects that are based on a table, query, or SQL statement

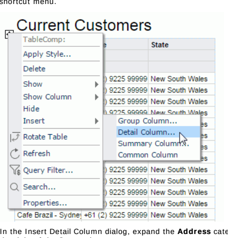


The report tab appears as follows:

Current Customers

Customer Name	Phone	State
Asia-Pacific		
Australia		
Cafe Brazil - Sydne	+61 (2) 9225 99999	New South Wales
Cafe Brazil - Sydne	+61 (2) 9225 99999	New South Wales
Cafe Brazil - Sydne	+61 (2) 9225 99999	New South Wales
Cafe Brazil - Sydne	+61 (2) 9225 99999	New South Wales
Cafe Brazil - Sydne	+61 (2) 9225 99999	New South Wales
Cafe Brazil - Sydne	+61 (2) 9225 99999	New South Wales
Cafe Brazil - Sydne	+61 (2) 9225 99999	New South Wales
Cafe Brazil - Sydne	+61 (2) 9225 99999	New South Wales
Cafe Brazil - Sydne	+61 (2) 9225 99999	New South Wales
Cafe Brazil - Sydne	+61 (2) 9225 99999	New South Wales
Cafe Brazil - Sydne	+61 (2) 9225 99999	New South Wales
Cafe Brazil - Sydne	+61 (2) 9225 99999	New South Wales
Cafe Brazil - Sydne	+61 (2) 9225 99999	New South Wales
Cafe Brazil - Sydne	+61 (2) 9225 99999	New South Wales
Cafe Brazil - Sydne	+61 (2) 9225 99999	New South Wales
Cafe Brazil - Sydne	+61 (2) 9225 99999	New South Wales
Cafe Brazil - Sydne	+61 (2) 9225 99999	New South Wales
Cafe Brazil - Sydne	+61 (2) 9225 99999	New South Wales
Cafe Brazil - Sydne	+61 (2) 9225 99999	New South Wales
Cafe Brazil - Sydne	+61 (2) 9225 99999	New South Wales
China		
Cafe Colombia	+886 (2) 771 99999	Taipei
Cafe Colombia	+886 (2) 771 99999	Taipei
Cafe Colombia	+886 (2) 771 99999	Taipei

5. Click anywhere on the table, when the icon appears at its upper left corner, right-click the icon and select **Insert** > **Detail Column** from the shortcut menu.



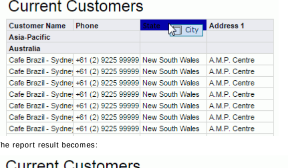
6. In the Insert Detail Column dialog, expand the **Address** category, then select **Address 1** and click **OK**. A detail column, Address 1, will be added to the right of the State column.

Current Customers

Customer Name	Phone	State	Address 1
Asia-Pacific			
Australia			
Cafe Brazil - Sydne	+61 (2) 9225 99999	New South Wales	A.M.P. Centre
Cafe Brazil - Sydne	+61 (2) 9225 99999	New South Wales	A.M.P. Centre
Cafe Brazil - Sydne	+61 (2) 9225 99999	New South Wales	A.M.P. Centre
Cafe Brazil - Sydne	+61 (2) 9225 99999	New South Wales	A.M.P. Centre
Cafe Brazil - Sydne	+61 (2) 9225 99999	New South Wales	A.M.P. Centre
Cafe Brazil - Sydne	+61 (2) 9225 99999	New South Wales	A.M.P. Centre
Cafe Brazil - Sydne	+61 (2) 9225 99999	New South Wales	A.M.P. Centre
Cafe Brazil - Sydne	+61 (2) 9225 99999	New South Wales	A.M.P. Centre
Cafe Brazil - Sydne	+61 (2) 9225 99999	New South Wales	A.M.P. Centre
Cafe Brazil - Sydne	+61 (2) 9225 99999	New South Wales	A.M.P. Centre
Cafe Brazil - Sydne	+61 (2) 9225 99999	New South Wales	A.M.P. Centre
Cafe Brazil - Sydne	+61 (2) 9225 99999	New South Wales	A.M.P. Centre
Cafe Brazil - Sydne	+61 (2) 9225 99999	New South Wales	A.M.P. Centre
Cafe Brazil - Sydne	+61 (2) 9225 99999	New South Wales	A.M.P. Centre
Cafe Brazil - Sydne	+61 (2) 9225 99999	New South Wales	A.M.P. Centre
Cafe Brazil - Sydne	+61 (2) 9225 99999	New South Wales	A.M.P. Centre
Cafe Brazil - Sydne	+61 (2) 9225 99999	New South Wales	A.M.P. Centre
Cafe Brazil - Sydne	+61 (2) 9225 99999	New South Wales	A.M.P. Centre
Cafe Brazil - Sydne	+61 (2) 9225 99999	New South Wales	A.M.P. Centre
Cafe Brazil - Sydne	+61 (2) 9225 99999	New South Wales	A.M.P. Centre
China			
Cafe Colombia	+886 (2) 771 99999	Taipei	2 Tun Hwa South R
Cafe Colombia	+886 (2) 771 99999	Taipei	2 Tun Hwa South R
Cafe Colombia	+886 (2) 771 99999	Taipei	2 Tun Hwa South R

However, we now consider the state information is not very useful because we will care more about the city information. So we can overwrite State with City.

7. Drag **City** in the Customers category from the Resource View panel to the State column label and drop it:

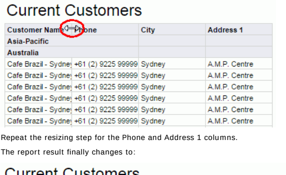


The report result becomes:

Current Customers

Customer Name	Phone	City	Address 1
Asia-Pacific			
Australia			
Cafe Brazil - Sydne	+61 (2) 9225 99999	Sydney	A.M.P. Centre
Cafe Brazil - Sydne	+61 (2) 9225 99999	Sydney	A.M.P. Centre
Cafe Brazil - Sydne	+61 (2) 9225 99999	Sydney	A.M.P. Centre
Cafe Brazil - Sydne	+61 (2) 9225 99999	Sydney	A.M.P. Centre
Cafe Brazil - Sydne	+61 (2) 9225 99999	Sydney	A.M.P. Centre
Cafe Brazil - Sydne	+61 (2) 9225 99999	Sydney	A.M.P. Centre
Cafe Brazil - Sydne	+61 (2) 9225 99999	Sydney	A.M.P. Centre
Cafe Brazil - Sydne	+61 (2) 9225 99999	Sydney	A.M.P. Centre
Cafe Brazil - Sydne	+61 (2) 9225 99999	Sydney	A.M.P. Centre
Cafe Brazil - Sydne	+61 (2) 9225 99999	Sydney	A.M.P. Centre
Cafe Brazil - Sydne	+61 (2) 9225 99999	Sydney	A.M.P. Centre
Cafe Brazil - Sydne	+61 (2) 9225 99999	Sydney	A.M.P. Centre
Cafe Brazil - Sydne	+61 (2) 9225 99999	Sydney	A.M.P. Centre
Cafe Brazil - Sydne	+61 (2) 9225 99999	Sydney	A.M.P. Centre
Cafe Brazil - Sydne	+61 (2) 9225 99999	Sydney	A.M.P. Centre
Cafe Brazil - Sydne	+61 (2) 9225 99999	Sydney	A.M.P. Centre
Cafe Brazil - Sydne	+61 (2) 9225 99999	Sydney	A.M.P. Centre
Cafe Brazil - Sydne	+61 (2) 9225 99999	Sydney	A.M.P. Centre
Cafe Brazil - Sydne	+61 (2) 9225 99999	Sydney	A.M.P. Centre
Cafe Brazil - Sydne	+61 (2) 9225 99999	Sydney	A.M.P. Centre

8. Resize the Customer Name column by dragging the boundary to the right of the column to make all the customer names in the column displayed entirely.



9. Repeat the resizing step for the Phone and Address 1 columns.

The report result finally changes to:

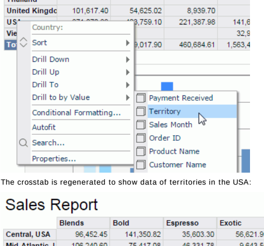
Current Customers

Customer Name	Phone	City	Address 1
Asia-Pacific			
Australia			
Cafe Brazil - Sydney	+61 (2) 9225 99999	Sydney	A.M.P. Centre
Cafe Brazil - Sydney	+61 (2) 9225 99999	Sydney	A.M.P. Centre
Cafe Brazil - Sydney	+61 (2) 9225 99999	Sydney	A.M.P. Centre
Cafe Brazil - Sydney	+61 (2) 9225 99999	Sydney	A.M.P. Centre
Cafe Brazil - Sydney	+61 (2) 9225 99999	Sydney	A.M.P. Centre
Cafe Brazil - Sydney	+61 (2) 9225 99999	Sydney	A.M.P. Centre
Cafe Brazil - Sydney	+61 (2) 9225 99999	Sydney	A.M.P. Centre
Cafe Brazil - Sydney	+61 (2) 9225 99999	Sydney	A.M.P. Centre
Cafe Brazil - Sydney	+61 (2) 9225 99999	Sydney	A.M.P. Centre
Cafe Brazil - Sydney	+61 (2) 9225 99999	Sydney	A.M.P. Centre
Cafe Brazil - Sydney	+61 (2) 9225 99999	Sydney	A.M.P. Centre
Cafe Brazil - Sydney	+61 (2) 9225 99999	Sydney	A.M.P. Centre
Cafe Brazil - Sydney	+61 (2) 9225 99999	Sydney	A.M.P. Centre
Cafe Brazil - Sydney	+61 (2) 9225 99999	Sydney	A.M.P. Centre
Cafe Brazil - Sydney	+61 (2) 9225 99999	Sydney	A.M.P. Centre
Cafe Brazil - Sydney	+61 (2) 9225 99999	Sydney	A.M.P. Centre
Cafe Brazil - Sydney	+61 (2) 9225 99999	Sydney	A.M.P. Centre
Cafe Brazil - Sydney	+61 (2) 9225 99999	Sydney	A.M.P. Centre
Cafe Brazil - Sydney	+61 (2) 9225 99999	Sydney	A.M.P. Centre
Cafe Brazil - Sydney	+61 (2) 9225 99999	Sydney	A.M.P. Centre
China			
Cafe Colombia	+886 (2) 771 99999	Taipei	2 Tun Hwa South Road, Section 2
Cafe Colombia	+886 (2) 771 99999	Taipei	2 Tun Hwa South Road, Section 2
Cafe Colombia	+886 (2) 771 99999	Taipei	2 Tun Hwa South Road, Section 2
Cafe Colombia	+886 (2) 771 99999	Taipei	2 Tun Hwa South Road, Section 2

10. Click the **Save** button on the toolbar. In the Save Report Template dialog, click **Yes** to save the report, then click **OK** in the prompt message window.

Task 7: Analyze data of a crosstab

1. Click **Sales** from the Go To drop-down list on the toolbar to switch to the report Sales, which contains a crosstab and a chart.
2. Right-click **USA** in the row header, and select **Drill to by Value** > **Territory** to obtain a more detailed view of USA information.



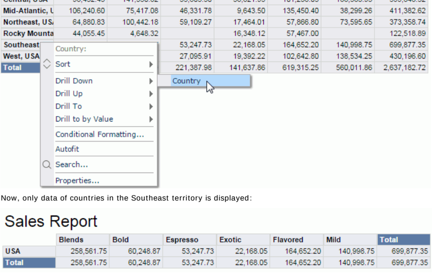
The crosstab is regenerated to show data of territories in the USA:

Sales Report

	Blends	Bold	Espresso	Exotic	Flavored	Mild	Total
Central, USA	96,452.45	141,350.82	35,603.30	56,621.95	101,236.05	168,583.95	599,848.52
Mid-Atlantic, U	106,240.60	75,417.08	46,331.78	9,643.50	135,450.40	38,299.26	411,382.62
Northeast, US	64,880.83	100,442.18	59,109.27	17,464.01	57,866.80	73,595.65	373,358.74
Rocky Mounta	44,055.45	4,648.32		16,348.12	57,467.00		122,518.89
Southeast, US	258,561.75	60,248.87	53,247.73	22,168.05	164,652.20	140,998.75	699,877.35
West, USA	100,879.60	41,651.81	27,095.91	19,392.22	102,642.80	138,534.25	430,196.60
Total	671,070.68	423,759.10	221,387.98	141,637.86	619,315.25	560,011.86	2,637,182.72

Next, we want to further view the details about the territory Southeast of USA. Because Territory is a level predefined in the hierarchy Geography of WorldWideSalesBV, we can drill it down to the Country level.

3. Right-click **Southeast, USA** in the row header, and select **Drill Down** > **Country** from the shortcut menu.



Now, only data of countries in the Southeast territory is displayed:

Sales Report

	Blends	Bold	Espresso	Exotic	Flavored	Mild	Total
USA	258,561.75	60,248.87	53,247.73	22,168.05	164,652.20	140,998.75	699,877.35
Total	258,561.75	60,248.87	53,247.73	22,168.05	164,652.20	140,998.75	699,877.35

We can further drill Country down to State in this way if necessary. Here, let's leave the crosstab as it is.

Next, we want to pivot the crosstab, namely interchange the rows and columns.

4. Click any number value in the crosstab, then click **Menu** > **Report** > **Rotate Crosstab**, and the crosstab changes to:

Sales Report

	USA	Total
Blends	258,561.75	258,561.75
Bold	60,248.87	60,248.87
Espresso	53,247.73	53,247.73
Exotic	22,168.05	22,168.05
Flavored	164,652.20	164,652.20
Mild	140,998.75	140,998.75
Total	699,877.35	699,877.35

5. Right-click any value and select **Properties** from the shortcut menu.
6. In the General tab of the Data Field Properties dialog, set the Format property to **$#,###.##**, then click **OK** to confirm.
7. The report result finally changes to:

Sales Report

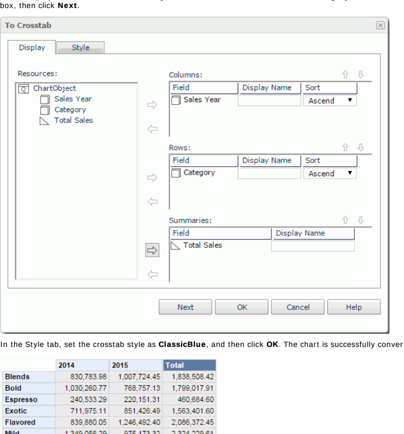
	USA	Total
Blends	$258,561.75	$258,561.75
Bold	$60,248.87	$60,248.87
Espresso	$53,247.73	$53,247.73
Exotic	$22,168.05	$22,168.05
Flavored	$164,652.2	$164,652.2
Mild	$140,998.75	$140,998.75
Total	$699,877.35	$699,877.35

8. Click the **Save** button on the toolbar. In the Save Report Template dialog, click **Yes** to save the report, then click **OK** in the prompt message window.

Task 8: Analyze data of a chart

Now let's focus on the chart at the lower part of the Sales report. In JReport, a chart can be converted to a crosstab, and vice versa. So we will convert the chart to a crosstab first.

1. Right-click anywhere on the chart and select **To Crosstab** from the shortcut menu.
2. In the Display tab of the To Crosstab dialog, add **Sales Year** to the Columns box, **Category** to the Rows box, and **Total Sales** to the Summaries box, then click **Next**.



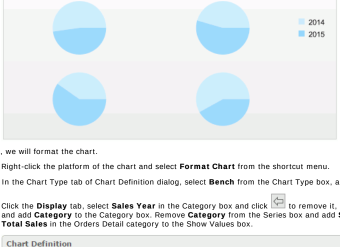
3. In the Style tab, set the crosstab style as **ClassicBlue**, and then click **OK**. The chart is successfully converted to a crosstab.

	2014	2015	Total
Blends	830,783.98	1,007,724.45	1,838,508.42
Bold	1,030,260.77	768,757.13	1,799,017.91
Espresso	240,533.29	220,151.31	460,684.60
Exotic	711,975.11	851,426.49	1,563,401.60
Flavored	839,880.05	1,246,492.40	2,086,372.45
Mild	1,349,056.29	975,173.32	2,324,229.61
Total	5,002,489.49	5,069,725.10	10,072,214.59

Now, we want to return the crosstab to a chart, and change the chart type to pie chart.

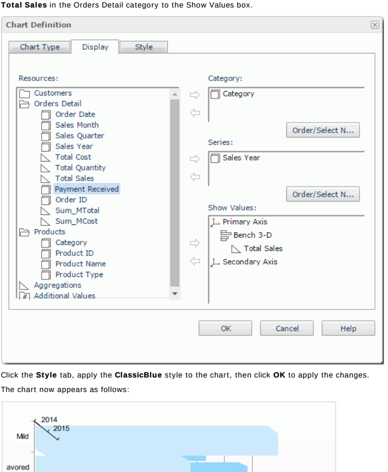
4. Click the menu command **Menu** > **Edit** > **Undo** to cancel the operation of converting chart to crosstab.

   We can also select any cell of the crosstab and click **Menu** > **Report** > **To Chart** to convert the crosstab to chart.
5. Right-click anywhere on the chart, and on the shortcut menu, click **Chart Type** > **Pie** > **Clustered Pie**. The chart appears as follows:



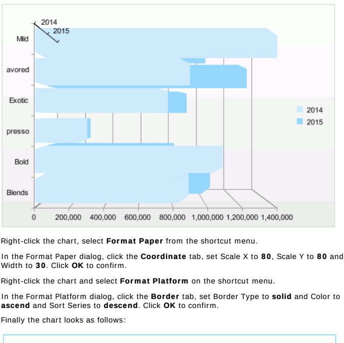
Next, we will format the chart.

6. Right-click the platform of the chart and select **Format Chart** from the shortcut menu.
7. In the Chart Type tab of Chart Definition dialog, select **Bench** from the Chart Type box, and select the **Bench 3-D** (the last one) thumbnail.
8. Click the **Display** tab, select **Sales Year** in the Category box and click to remove it, then expand the **Products** category in the Resources box and add **Category** to the Category box. Remove **Category** from the Series box and add **Sales Year** in the Orders Detail category to the Show Values box. Add **Total Sales** in the Orders Detail category to the Show Values box.



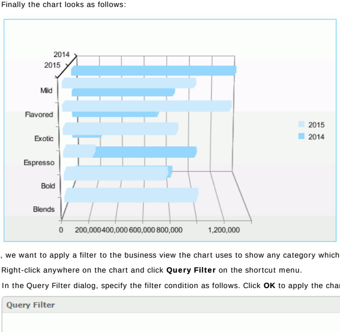
9. Click the **Style** tab, apply the **ClassicBlue** style to the chart, then click **OK** to apply the changes.

   The chart now appears as follows:



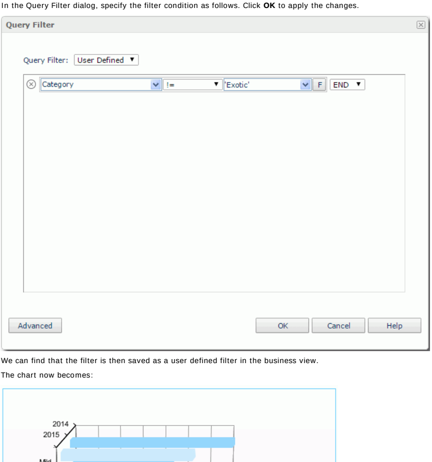
10. Right-click the chart, select **Format Paper** from the shortcut menu.
11. In the Format Paper dialog, click the **Coordinate** tab, set Scale X to **80**, Scale Y to **80** and Angle Y to **12**, then switch to the **Graph** tab and set Bar Width to **30**. Click **OK** to confirm.
12. Right-click the chart and select **Format Platform** on the shortcut menu.
13. In the Format Platform dialog, click the **Border** tab, set Border Type to **solid** and Color to **#66ccff**, then click the **Data** tab, set Sort Category to **ascend** and Sort Series to **descend**. Click **OK** to confirm.

    Finally the chart looks as follows:



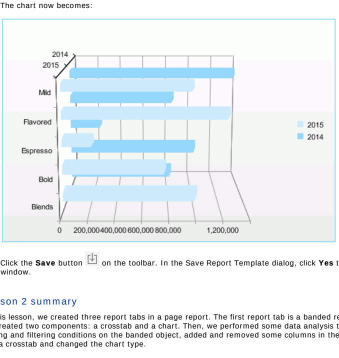
Next, we want to apply a filter to the business view the chart uses to show any category which is not Exotic.

14. Right-click anywhere on the chart and click **Query Filter** on the shortcut menu.
15. In the Query Filter dialog, specify the filter condition as follows. Click **OK** to apply the changes.



    We can find that the filter is then saved as a user defined filter in the business view.

    The chart now becomes:



16. Click the **Save** button on the toolbar. In the Save Report Template dialog, click **Yes** to save the report, then click **OK** in the prompt message window.

Lesson 2 summary

In this lesson, we created three report tabs in a page report. The first report tab is a banded report, the second contains a table. In the third report tab, we created two components: a crosstab and a chart. Then, we performed some data analysis tasks on the four data components. We applied some sorting and filtering conditions on the banded object, added and removed some columns in the table, drilled around the crosstab, and converted the chart into a crosstab and changed the chart type.

Part III: Advanced Reporting

In this part, we will focus on some advanced reporting features of JReport: how to use JReport Designer to create data resources and use the resources to build reports and library components; how the resources are published to JReport Server and managed on the server, etc..

The reports we build in this part are related to a fictional, mid-size retail coffee chain named Jinfonet Gourmet Java. All documents needed to run the business and drive decision-making are reports generated by JReport.

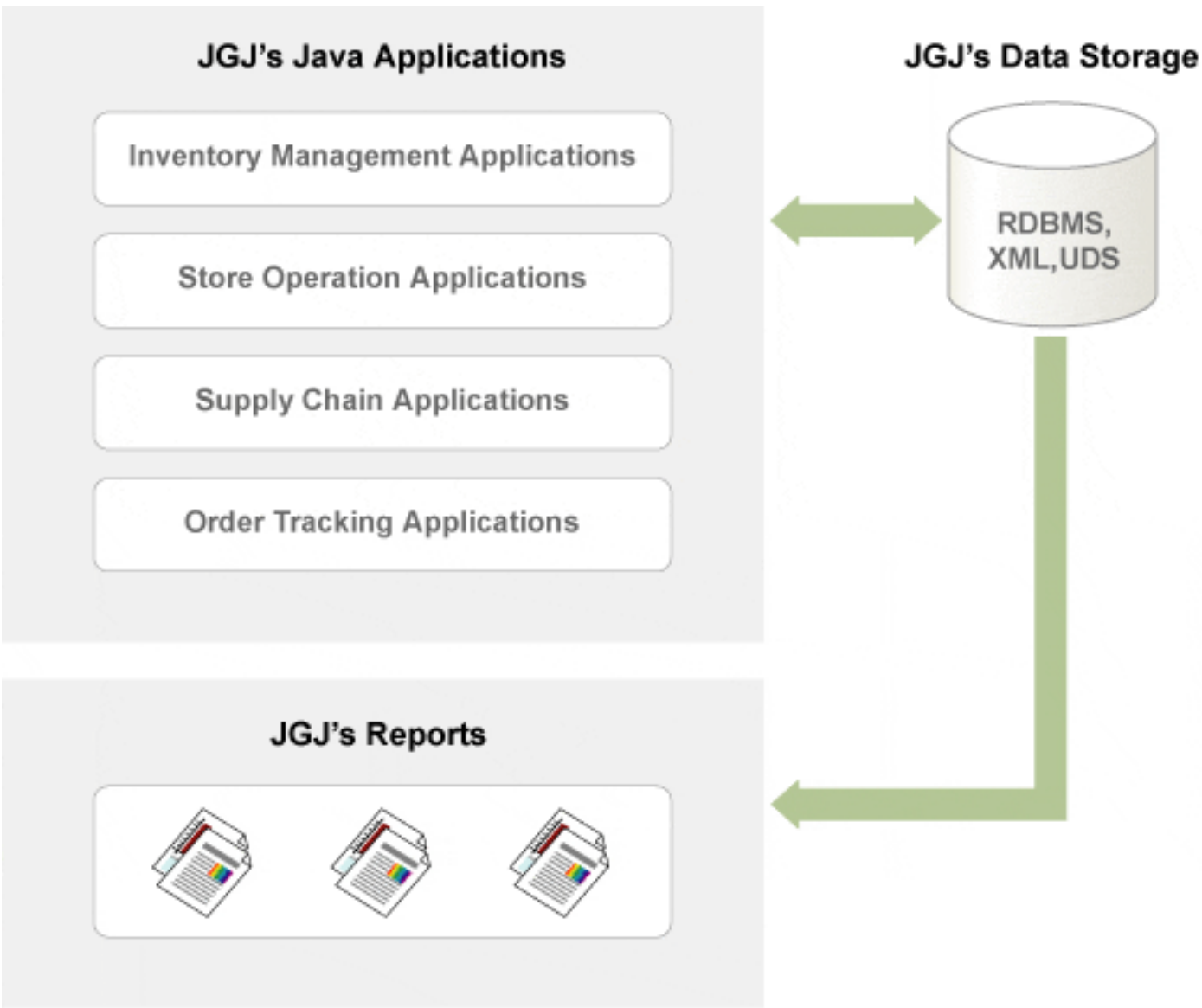
The business questions that our reports can answer include:

- What were the Sales totals last month? Last year?
- Which regions had the highest sales? Which sales representative?
- Which product was the best selling last month? Last year?

Reports deliver real-time data

By integrating sophisticated reporting functionality in enterprise applications, organizations can access, analyze and deliver real-time data as well as unlock critical business data hidden within applications.

The reports access the same data storage as the applications of Jinfonet Gourmet Java:



Data sources used in the tracks

The data store for the company is an RDBMS and XML files. The applications of the company—inventory management system, store operations, supply chain management, and order tracking—access and update the data store.

For the reports that you will be creating and managing in this part, the data sources are predefined in a built-in HyperSonic DBMS database and an XML data source. The table below shows the tables in the DBMS and XML data sources. The DBMS Name/XML Name is the name of the table as it is known in the DBMS or XML data itself; the Table Name is the table name that is shown in the JReport Designer:

DBMS database		
DBMS Name	**Table Name**	**Table Description**
ACCOUNTMANAGERS	Account Managers	Jinfonet Gourmet Coffee employees
CONTACTS	Contacts	Customer contact information
CUSTOMERS	Customers	Customer name, contact, and address
ORDERS	Orders	Customer orders
ORDERSDETAIL	Orders Detail	Orders line items
PRODUCTS	Products	Product information
STOCKMARKET	Stock Market	Stock information
XML data source		
XML Name	**Table Name**	**Table Description**
_Jinfonet	Jinfonet	
_Jinfonet_Employee	Employee	Jinfonet Gourmet Coffee employees
_Jinfonet_CompanyInfo	CompanyInfo	Company information
_Jinfonet_StockMarket	StockMarket	Stock information
_Jinfonet_Customer	Customer	Customer name, contact, and address
_Jinfonet_Customer_Order	Order	Customer orders
_Jinfonet_Customer_Order_OrderDetail	OrdersDetail	Orders line items
_Jinfonet_Catalog	Catalog	Product number and description for catalog
_Jinfonet_Catalog_Product	Product	Product information

By mapping the actual data source names to JReport Designer names, report designers can work with more intuitive or customized names.

This part contains the following tracks:

- Track 1: Creating reports in JReport Designer
- Track 2: Creating business views and web reports
- Track 3: Creating library components
- Track 4: Publishing, running and administering resources
- Track 5: Integrating reports with JSP

Click the right arrow ( ) below to begin reading the part.

Track 1: Creating reports in JReport Designer

In this track, assume that you are a report developer working for the Jinfonet Gourmet Java (JGJ) company. You have been asked to write a set of reports based on the data in the company's existing database. For a description of the tables in this database, see Data sources used in the tracks.

JReport reports can pull data from any type of JDBC-accessible database, including Oracle and SQL Server, as well as XML data sources. To simplify the lessons, we'll use the built-in Hypersonic database and XML files which contain the sample data we need.

This track introduces JReport Designer, a Swing-based Integrated Development Environment (IDE), for designing and testing of reports.

This track contains a lesson for each type of predefined report:

- Lesson 1: Creating a standard banded report
- Lesson 2: Creating a horizontal banded report
- Lesson 3: Creating a mailing label report
- Lesson 4: Creating a chart report
- Lesson 5: Creating a table report
- Lesson 6: Creating a crosstab report
- Lesson 7: Creating a tabular report
- Lesson 8: Creating a report that contains a subreport
- Lesson 9: Creating a parameter-based report

You will be developing a set of new reports based on the procedures in this track. JReport provides finished versions of these reports if you would like to reference them. See the note at the end of each lesson for information about how to access the JReport-provided tutorial reports.

Click the right arrow (⇨) below to begin reading the track.

Lesson 1: Creating a standard banded report

A report of monthly sales from 2004 to 2005 has been requested by the Vice President of Sales at Jimbom Sportswear Inc. In this lesson, you will create the report, which will be grouped by Year and Month, and generated for a great deal of all sales. Jimbom's a draft of the report is shown on the next page to you.

```
                    Order List by Date
Year 1 Month 1
  Order Id   Date        Product ID   Product Name     Category    Sales
  1          1-1-2004    1            Product 1        Category    Sales
  2          1-2-2004    2            Product 2        Category    Sales
  3          1-3-2004    3            Product 3        Category    Sales
                                      Total sales for Month
Year 1 Month 2
  Order Id   Date        Product ID   Product Name     Category    Sales
  1          2-1-2004    1            Product 1        Category    Sales
  2          2-2-2004    2            Product 2        Category    Sales
  3          2-3-2004    3            Product 3        Category    Sales
                                      Total sales for Month
Year 1 Month 3
  Order Id   Date        Product ID   Product Name     Category    Sales
  1          3-1-2004    1            Product 1        Category    Sales
  2          3-2-2004    2            Product 2        Category    Sales
  3          3-3-2004    3            Product 3        Category    Sales
                                      Total sales for Month
                                      Total sales for Year
```

You immediately recognize the repetitive, columnar data in the center of the report and associated subtotals as a candidate for a standard banded report. This type of report has functional "bands" or assets that are geared for similar data and subtotals. Jimbom's report will be a good candidate for the use of the Report Wizard and Report Designer.

In this lesson, we perform the following tasks:

- Task 1. Create the initial report and query in preparation 4
- Task 2. Add summaries and a pivot table to the report
- Task 3. Fine tune the report layout

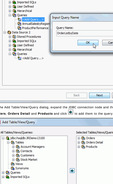
Task 1. Create the initial report and query in preparation 4

In this task, the report wizard creates the necessary information and then creates the standard banded report as well as the Report Designer.

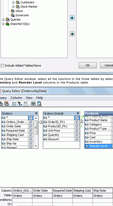
1. Click **Start > All Programs > (Report 11.5 Update) > Report Designer** to start Report Designer. The Project Designer appears. Create a new project by clicking **File > New > Project**. Name the project **Jimbom**.
2. In the Project Designer menu bar, select **File > Opening Workspace > Open Catalog** to bring up the Open Catalog dialog box.
3. Browse to select the **JimbomOrderSalesDate.cat** catalog file in **<installed>\Samples\Reports\JimbomOrderSalesDate.cat**, click the **Open** button, and then click the **Open** button in the Open Catalog dialog box.
4. Choose **File > New > Page Report** on the menu bar, or click the **New Page Report** button on the Standard toolbar.
5. In the New Page Report dialog box, select **Use the Report Wizard**, then select **Banded** from the report, then click **OK**. In all lessons in this crack, we will create the report inform of other ways, so in the New Page Report dialog, we need to close the report into the wizard. Alternate a default report into will be used.

A page report is a composer which holds multiple report data. Report tabs in the same report can share the same parameters and data. Report Designer provides a wizard and a blank report template. Report Designer also provides a blank report template and a wizard that together. A page report has the e.rdl extension.

6. In the Data screen of the Report Wizard, click **Add > Existing Report...** in the Queries node of Data Source 1, enter **OrderListByDate** in the Query Name and Data Set Name and click **OK**.



7. In the Add Existing Report dialog, expand the Report Queries node and then the **Tables** node, then select the tables **Orders**, **Orders Detail**, **Products** and **Order**, and click **Open**. Click **OK** to close the dialog.



8. In the Query Editor window, select all the columns in the three tables by selecting the * checkbox, then uncheck the remaining and **Reorder Level** columns in the Products table.

By default, the wizard uses options for queries and related joins we find the options on the wizard Data > Arrow data to the Data Editors, which are automatically created and saved in the job. Click **OK**. The Data Editor pop-up. Select the Data Editor, and the Data Editor pop-up. Select the Data Editor, and the Data Editor pop-up. Select the Data Editor, and the

Lesson 2: Creating a horizontal banded report

Now your reporting assignment is to create a series of summaries of each Jotmart Gourmet Java employee. The report will also need to communicate the ranking of the employees' sales, that is, whether it is in the upper or lower end, or somewhere in between, relative to all employees. The following prototype of the report has been given to you:

This lesson contains the following tasks:

- Task 1: Create the initial report
- Task 2: Adjust the layout of the report and add the Photo DBField
- Task 3: Rank the employees
- Task 4: Fine tune the report layout

Task 1: Create the initial report

Again the report has repeated rows of information, but this time they are repeated left to right instead of top down. A horizontal banded report is different to support this format.

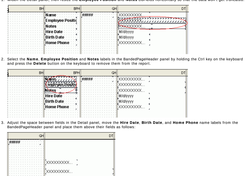
To create the initial report with a horizontal banded report, follow these steps:

1. From the *Report Designer* toolbar, click the **New Page Report** button.
2. In the New Page Report dialog, clear the text in the Report Title text box, select **Horizontal Banded** from the layout box, then click **OK**.

   Be sure that JotmartGourmetJava can's specified as the current catalog because it is the catalog we use in this track. For information about specifying this catalog, see Task 1, Step 2 and 3 of Lesson 1.
3. In the Data screen of the Horizontal Banded Wizard, click **+Add Query...>** in the Queries node of Data Source 1, enter EmployeesPhotos as the Query Name, and click **Text Command**.
4. In the Add Text Command Query dialog, paste the DBF connection code and then the **Tables** node, then select the table **Account Managers** and click **>>** to add it to the query. Click **OK** to close the dialog.
5. In the Query Editor, select **?** in the Account Managers table to add all the columns in it to the query and then click **OK** on the bottom right to close the Query Editor.

   Click **Next** in the Horizontal Banded Wizard to go to the Display screen.
6. From the Resources tree, select the following DBFields in the Account Managers table and click **>>** button to add them in the right-hand box one by one: **Name**, **Employee Position**, **Notes**, **HireDate**, **Birthday** and **Home Phone**. Add the display names HireDate and Birthday to **Hire Date** and **Birth Date**, then click **Next**.

8. In the Group screen, select **Account Manager ID** and click **>>** to add it as the group by field.
9. Switch to the **Style** screen and select **Simple** as the report style. Click **Finish** to create the report and the report appears as follows:



Task 2: Adjust the layout of the report and add the Photo DBField

To improve the appearance of the report, we will do some adjustments to the report's layout. In addition, we will add a Photo field to the report. Since it is a BLOB field, we will use a Picture Box control.

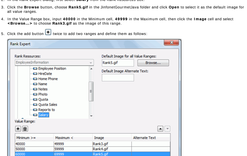
1. Widen the Detail panel, then resize the **Employee Position** and **Notes** DBFields horizontally so that data won't get truncated.

2. Select the **Name**, **Employee Position** and **Notes** labels in the BandedPageHeader panel by holding the Ctrl key on the keyboard and press the **Delete** button on the keyboard to remove them from the report.

3. Adjust the space between fields in the Detail panel, move the **Hire Date**, **Birth Date**, and **Home Phone** name labels from the BandedPageHeader panel and place them above their fields as follows:

4. Hide the BandedHeader, BandedPageHeader, GroupHeader, BandedPageFooter, BandedFooter panels that don't hold any data by right-clicking their panel and select **Hide** from the shortcut menu.
5. Resize the GroupHeader panel, click **Insert > Label** to add a label before the group field in the panel, edit its text as **'Employee ID:'**, resize the group field and the label, and adjust the position like the following:

6. From the Resource View panel, drag the **Photo** DBField in the Account Managers table to the GroupHeader panel.
7. Remove the name label of the Photo DBField by right-clicking it and selecting **Delete** from the shortcut menu, then resize the field as follows:



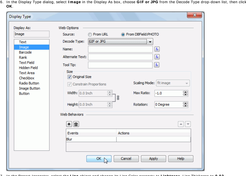
8. Click **Insert > Drawing Object** to insert a Line object above the Employee Position DBField, and insert a Box object in the same way to enclose all the objects in the reports:

Task 3: Rank the employees

As required at the beginning of the report design, the manager wants to rank the employees by their salaries, so we will add a rank object to the report.

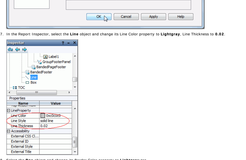
1. Select the Detail panel of the report and click **Insert > Rank** on the menu to display the Rank Expert dialog.
2. In the Rank Expert dialog, first select **Salary** from the Rank Resources box.
3. Click the **Browse** button, choose **Rank5.gif** in the JotmartGourmetJava folder and click **Open** to select a default image for the rank object.
4. In the Value Range box, input **45000** in the Minimum edit, **99999** in the Maximum edit, then click the **Image** cell and select **<Browse...>** to choose **Rank3.gif** as the image of this range.
5. Click the edit button (**...**) twice to add two ranges and define them as follows:

6. Click the **Insert** button to insert the rank to the Detail panel, below the Home Phone field.
7. Move the name label of the rank from the bottom of GroupHeader panel and place it above the rank, then edit its text to **Salary Ranking** and resize the label to full display the text.



Task 4: Fine tune the report layout

1. Select the **Detail** panel, **Ctrl-Shift-Left** the group by field in the GroupHeader panel by holding the Ctrl key on the keyboard, then change their Font Face property to **Arial**, and Font Size property to **8** in the Report Inspector.
2. Select all the objects except the Line and Rank objects in the Detail panel by holding the Ctrl key on the keyboard and change their Font Face property to **Arial** and Font Size to **8**.
3. Select the **Salary Ranking** label and the rank by holding the Ctrl key on the keyboard, then change their Horizontal Alignment property to **left** so as to make them aligned the same with other objects in the detail panel.
4. Select the **Name** DBField, change its properties in the Report Inspector as follows, then resize it to make sure the names can be fully displayed:
   - Font Size: 12
   - Bold: true
   - Foreground: Gray0000C0
5. Right-click the **Photo** DBField and select **Display Type** from the shortcut menu.
6. In the Display Type dialog, select **Image** in the Display As box, choose **GIF or JPG** from the Decode Type drop-down list, then click **OK**.



7. In the Report Inspector, select the **Line** object and change its Line Color property to **LightGray**, Line Thickness to **0.02**.

8. Select the **Box** object and change its Border Color property to **LightGray** too.

Next, we will add a title to the report:

1. Click **View > Page Header** to display the page header.
10. In the Report Inspector, select the **PageHeaderPanel** node and change its Height property to **0.6**.
11. Drag **Label** from the Toolbox panel and drop it in the PageHeader panel.
12. Resize the label, double-click it to modify its text as **Employee Information List**, and change its Font Face property to **Arial**, Font Size to **24**, Foreground to **DarkBlue** in the Report Inspector.

    After editing, the report looks somewhat like below in design view:

13. On the report tab bar, right-click the report tab and select **Rename** from the shortcut menu, then enter **EmployeeDetails** in the Report Name Text frame dialog and click **OK**.
14. Click **File > Save** to save the report as **EmployeeInformation.cls**.
15. Click the **View** tab to preview the report. The report appears similar to the below one:

   Note: If the report does not look correct, you can compare it to the final version of the report provided by JReport. To do so, you will need to save and close this catalog and then open the JotmartGourmetJava cat catalog file located at <install_root>/Samples/Reports/Catalogs/Basic.

Lesson 2 summary

In this lesson, we created a horizontal banded report to present a summary of each employee. We added drawing objects to improve the readability, and a rank graphic to communicate the relative salaries of the employees.

Lesson 3: Creating a mailing label report

In this lesson, we create a Customer Contact Card report using the mailing label layout. This report can be used to print rolodex cards as well as mailing labels. On each card, the following information needs to be shown:

Customer Name	Position
Company Name	
Country	
Phone:	Fax:

The data source for this report is our built-in XML data, predefined as Data Source 2 in the catalog.

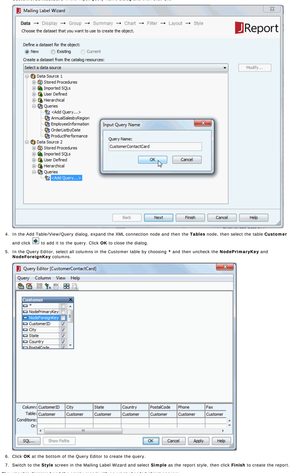
This lesson contains the following tasks:

- Task 1. Define the query that the report uses
- Task 2. Add objects to the report
- Task 3. Edit object properties and print the report

Task 1: Define the query that the report uses

1. From the JReport Designer toolbar, click the **New Page Report** button Del.
2. In the New Page Report dialog, select the text in the Report Title text box, select **Mailing Label** from the layout box, then click **OK**.

   Be sure that JrptlandSourceJava.cat is specified as the current catalog because it is the catalog we use in this track. For information about specifying this catalog, see Task 1. Step 4 and 5 of Lesson 1.
3. In the Data screen of the Mailing Label Wizard, click **<Add Query...>** in the Queries node of Data Source 2, enter **CustomerContactCard** in the Input Query Name dialog and then click **OK**.



4. In the Add Table/View/Query dialog, expand the XML connection node and then the **Tables** node, then select the table **Customer** and click to add it to the query. Click **OK** to close the dialog.
5. In the Query Editor, select all columns in the Customer table by choosing * and then uncheck the **NodePrimaryKey** and **NodeForeignKey** columns.

6. Click **OK** at the bottom of the Query Editor to create the query.
7. Switch to the **Style** screen in the Mailing Label Wizard and select **Simple** as the report style, then click **Finish** to create the report.

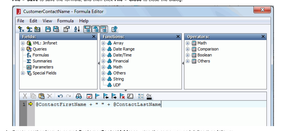
   The wizard is dismissed and the empty report with an empty banded object appears.

Task 2: Add objects to the report

1. Select and resize the Detail panel of the banded object, to make it similar to the one below:

We will use a simple formula to create a single value out of the customer's last and first names, and similarly to the address.

2. In the Resource View panel, select **<Add Formula...>** in the Formulas node.



3. Enter the formula name as **CustomerContactName** in the Input Formula Name dialog, click **OK**, define the formula in the Formula Editor as **@InitialFirstName + " " + @ContactLastName** (you can copy the formula to the formula editing area directly), click **File > Save** to save the formula, and then click **File > Close** to close the dialog.

4. Create another formula named **CustomerContactAddress** using the same way and define it as follows:

   @address1 + ", " + @City + ", " + @State + " " + @PostalCode

Now we will use one of the JReport Designer's customizations that can simplify the creation of this report.

5. Click **File > Options** on the menu bar.
6. In the Options dialog, select **Component**, remove the checkmark from the **Insert field name label with field** checkbox and click **OK**.

We don't want to show the name labels of the DBFields in this report. Now we can add our formulas and DBFields to the report.

7. From the Resource View panel, drag the **CustomerContactName** formula in the Formulas node to the Detail panel of the banded object, and then the **CustomerPhone** DBField in the Customer table to the right of it.

8. Use the same way to add the following fields:

9. Add two labels to identify the Phone and Fax DBFields by selecting the detail panel and clicking **Insert > Label**, resize the labels and edit their text to "**Phone:**" and "**Fax:**" respectively, select the two labels by holding the Ctrl key on the keyboard and change their Bold property to **false** in the Report Inspector, then the report displays as follows:

10. Insert a Line drawing object just above the CustomerName DBField by clicking **Insert > Drawing Objects > Line**.

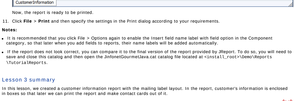
11. Select the Detail panel of the banded object and add a Box drawing object to enclose all the objects in the panel by clicking **Insert > Drawing Objects > Box**.

12. Resize the BandedPageHeader panel, add a label to it by clicking **Insert > Label**, then resize the label and edit its text to **Customer Contact Card** as the title of the report.

Task 3: Edit object properties and print the report

1. Select the label in the BandedPageHeader panel and change its properties in the Report Inspector as follows:
   - Bold: false
   - Font Face: Arial
   - Font Size: 14
   - Foreground: Red
2. Select all the objects in the Detail panel except the Line and the Box drawing objects by holding the Ctrl key on the keyboard, change their Font Face property to **Arial** and Font Size to **10** in the Report Inspector.
3. Select the **CustomerContactName** formula and set its Bold property to **true** in the Report Inspector.
4. In the Report Inspector, select the **Box** object and change its Border Color property to **Lightgray**.

5. Select the **Line** object and edit its Line Color property to **Lightgray** and Line Thickness to **0.02**.
6. Resize the fields horizontally if data is truncated.



7. Hide the BandedHeader, BandedPageFooter and BandedFooter panels that don't hold any data by right-clicking the panel and select **Hide** from the shortcut menu.
8. On the report tab bar, right-click the report tab and select **Rename** from the shortcut menu, then enter **CustomerInformation** in the Input Report File Name dialog and click **OK**.
9. Click **File > Save** to save the report as **CustomerContactCard.cls**.
10. Click the **View** tab to preview the report. The report appears similar as follows:

Now, the report is ready to be printed.

11. Click **File > Print** and then specify the settings in the Print dialog according to your requirements.

Notes:

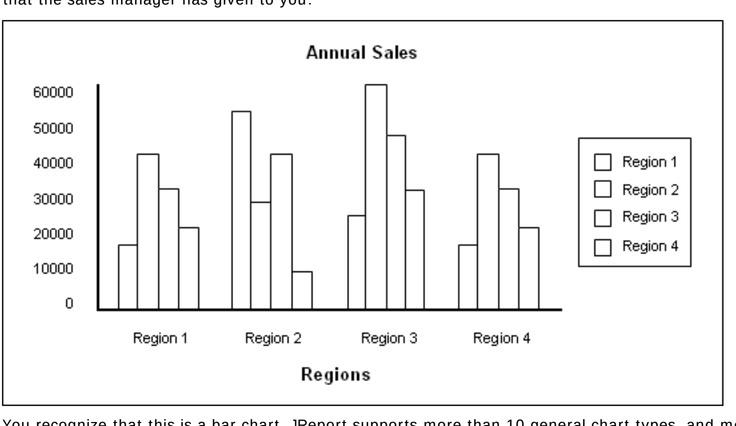
- It is recommended that you click File > Options again to enable the insert field name label with field option in the Component category, so that later when you add fields to reports, their name labels will be added automatically.
- If the report does not look correct, you can compare it to the final version of the report provided by JReport. To do so, you will need to save and close this sample and then open the JrptlandSourceJava.cat catalog file located at `<install_root>/demo/Reports/Future Samples`.

Lesson 3 summary

In this lesson, we created a customer information report with the mailing label layout. In the report, customer's information is enclosed in boxes so that later we can print the report and make contact cards out of it.

Lesson 4: Creating a chart report

The fourth quarter of the year has just ended, and you have been asked to produce a chart that shows product annual sales for each sales region. The sales manager wants to compare the historical sales data with the current data. Here's a sketch of the report that the sales manager has given to you:



You recognize that this is a bar chart. JReport supports more than 10 general chart types, and most of them have many sub-types or variations. After showing the sales manager the available bar chart types, he chooses the 2-dimensional clustered bar chart.

Additionally, the sales manager requests that the report output format should be Microsoft Excel. This is not a problem, because all pre-defined reports in JReport can be exported to HTML, PDF, Excel, RTF, XML, Text, Postscript, Mail and Fax.

This lesson contains the following tasks:

- Task 1: Create the chart
- Task 2: Format the chart
- Task 3: Export the report to an Excel file

Task 1: Create the chart

1. From the JReport Designer toolbar, click the New Page Report button.
2. In the New Page Report dialog, clear the text in the Report Title text box, select **Chart** from the layout box, then click **OK**.

   Be sure that JinfonetGourmetJava.cat is specified as the current catalog because it is the catalog we use in this track. For information about specifying this catalog, see Task 1, Step 2 and 3 of Lesson 1.
3. In the Data screen of the Chart Wizard, select the query **AnnualSalesbyRegion** from the Queries node of Data Source 1 and then click **Next**.
4. In the Type screen, check the **Single chart** radio button, select **Bar** as the chart type and **Clustered Bar 2-D** as the sub type, then click **Next**.

In this chart, we want to display the region name in the Category(X) axis, and annual sales of each region in the Value(Y) axis. However, as the region names are too long to be displayed completely in the Category(X) axis, we will create a formula here to just get the abbreviations of the region names.

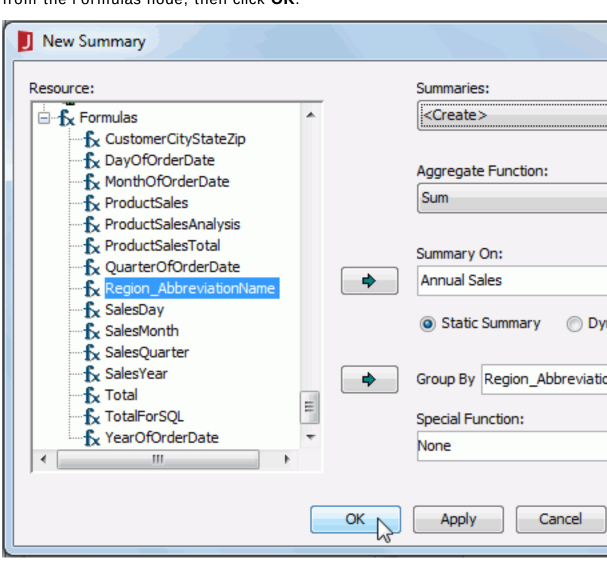
5. In the Display screen, scroll down to the **Formulas** node in the Resources panel, and then select **<Add Formula...>**.
6. Enter the formula name **Region_AbbreviationName** in the Input Formula Name dialog, click **OK**, define the formula in the Formula Editor window as follows (you can copy the formula to the formula editing area directly), click **File** > **Save** to save the formula, and then click **File** > **Close** to close the dialog and finish editing.

```
if (@Customers_REGION == "Asia-Pacific (APAC)")
        "APAC"
else if (@Customers_REGION == "Europe, Middle East, Africa (EMEA)")
        "EMEA"
else if (@Customers_REGION == "Latin America (LATAM)")
        "LATAM"
else if (@Customers_REGION == "North America (NA)")
        "NA"
```

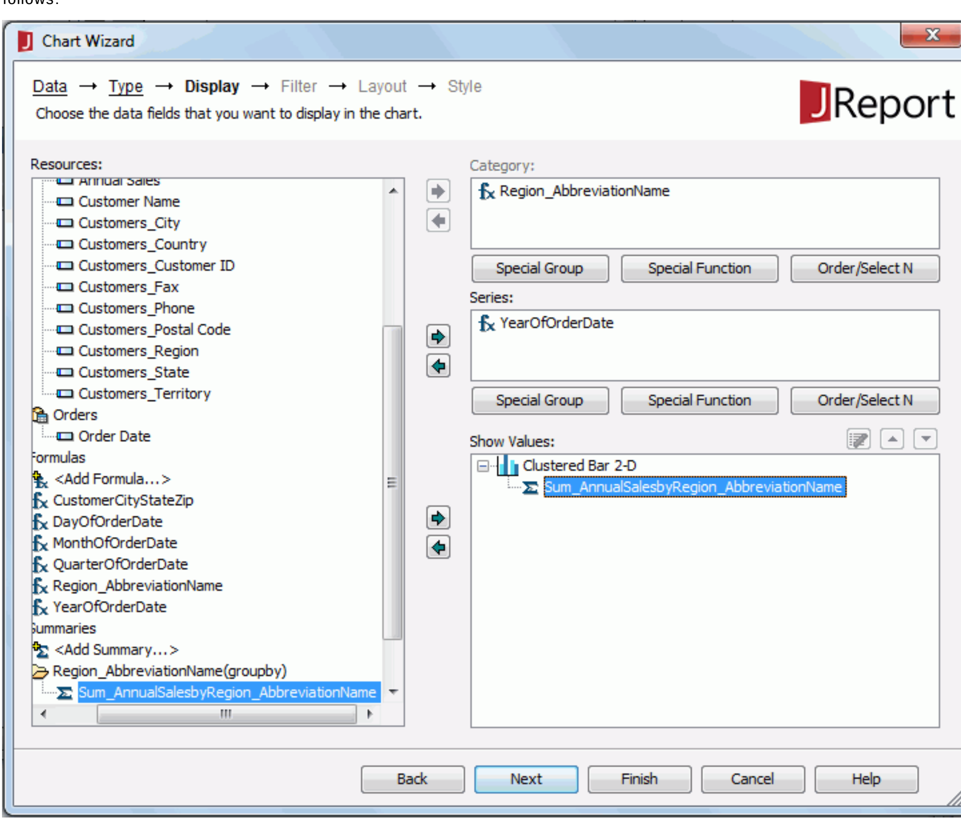
7. Add the formula **Region_AbbreviationName** to the Category box, and **YearofOrderDate** to the Series box by selecting them and clicking button one by one. In the Show Values box, a numerical value is required.

In this lesson, we need to show the annual sales of each region, so a summary which is created on the column Annual Sales, and grouped by the formula Region_AbbreviationName is needed.

8. Scroll down to the **Summaries** node in the Resources panel of the Display screen and then select **<Add Summary...>**.
9. In the New Summary dialog, specify the aggregate function as **Sum**, add the field **Annual Sales** from the Customers table to the Summary On field, check the **Static Summary** radio button, specify the Group By field as **Region_AbbreviationName** from the Formulas node, then click **OK**.

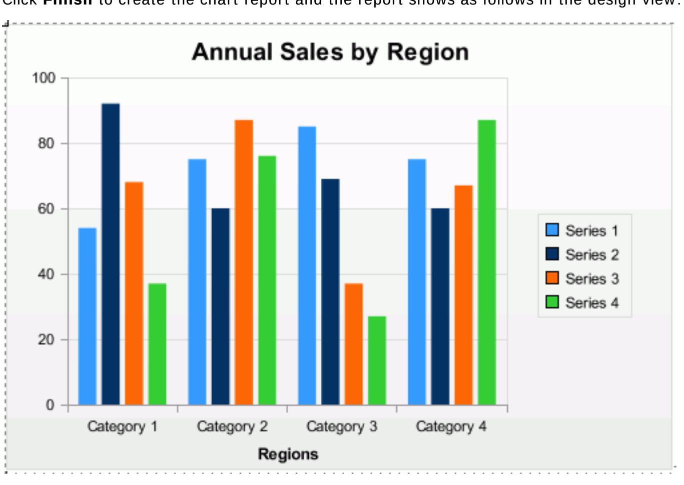


10. Enter **Sum_AnnualSalesbyRegion_AbbreviationName** in the Summary Name dialog and click **OK** to create the summary.
11. Add the newly-created summary to the Show Values box by selecting it and clicking button. The Display screen appears as follows:

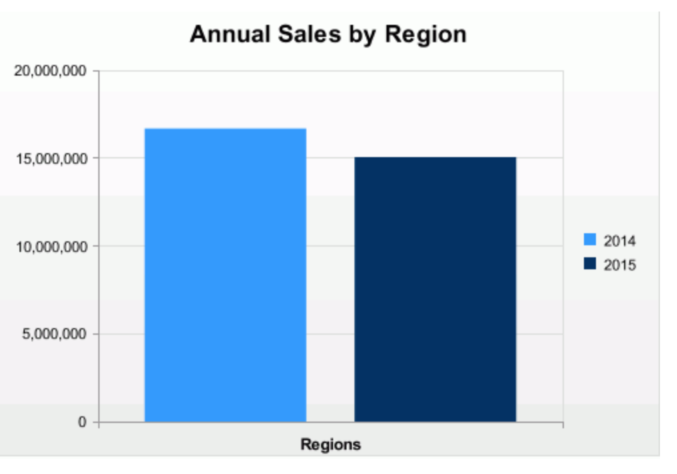


12. Click **Layout** to switch to the **Layout** screen, select **Title** from the Options box, and then enter **Annual Sales by Region** as the Chart Title and **Regions** as Category(X) Axis Title.

    JReport provides a set of CSS styles that can be applied to reports to easily change the format and appearance of the report. We will apply the Classic style to the chart.
13. Click **Next** to go to the **Style** screen and select **Classic** from the style list.
14. Click **Finish** to create the chart report and the report shows as follows in the design view:



15. Click the **View** tab to view this chart, and it appears as follows:

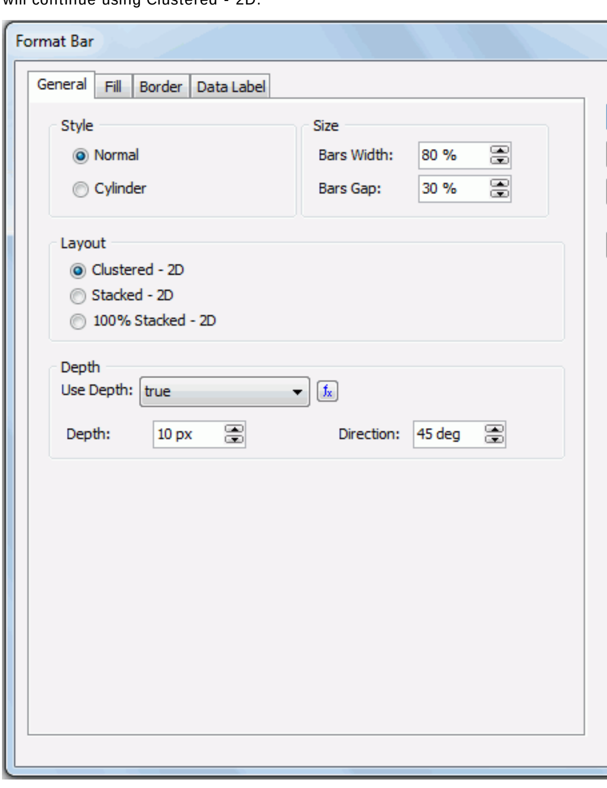


Task 2: Format the chart

The chart is accurate, but a little simple. We can add some polish to it by setting some of the chart properties.

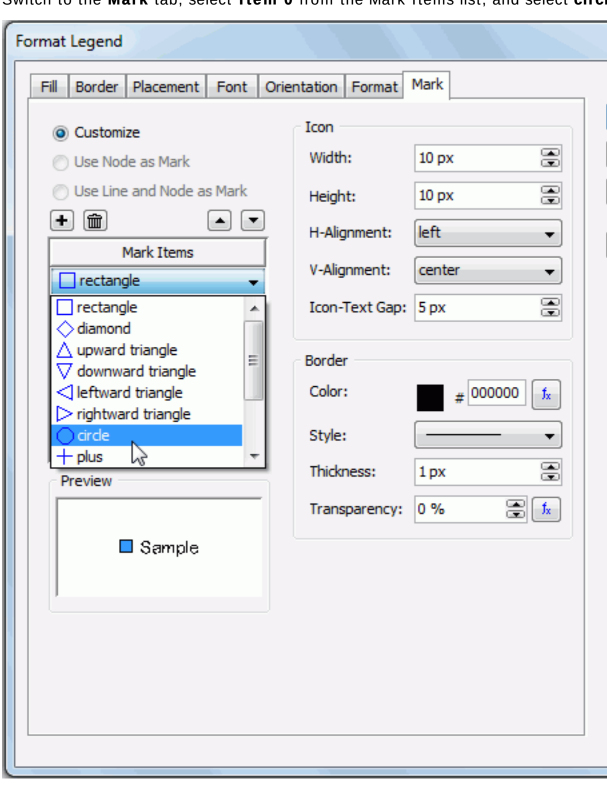
1. Click the **Design** tab to return to the design mode to do the adjustments.
2. Double-click a bar of the chart, and the Format Bar dialog appears.
3. In the General tab of the Format Bar dialog, set the **Use Depth** option to **true**, and specify Depth as **10px** and Direction as **45deg**.

   In the Format Bar dialog, you could also choose another sub-type for the chart in the Layout box if desired. In this lesson, we will continue using Clustered - 2D.

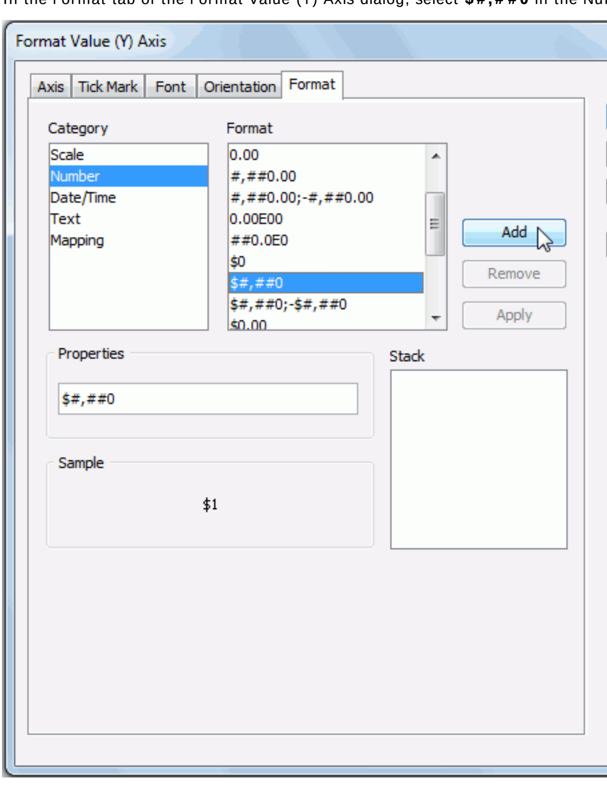


4. Switch to the **Data Label** tab, specify Font as **Arial** and Font Size to **10 pt**.

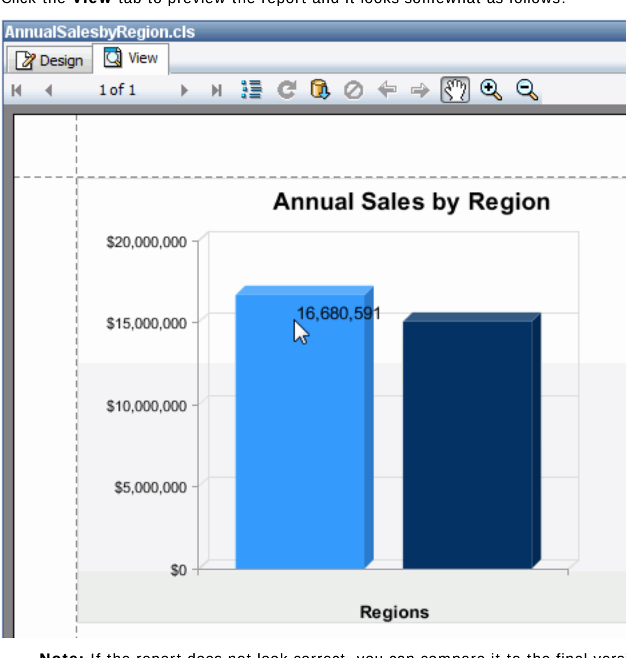
   The Data labels that charts contain can be either static or dynamic. You can check the Show Static Data Label option and specify the Position for data labels according to your requirements, so that the labels are displayed statically in the chart. But in this lesson, we will just use the labels as dynamic ones, which will appear when the cursor is placed on bars.
5. Click **OK** to apply these property settings to bars of the chart.
6. Right-click the legend and then select **Format Legend** from the shortcut menu.
7. In the Format Legend dialog, click the **Font** tab, set Font as **Arial** and Font Size to **10 pt**.
8. Switch to the **Mark** tab, select **Item 0** from the Mark Items list, and select **circle** from the mark drop-down list.



9. Select **Item 1**, and apply **upward triangle** to it in the same way.
10. Click **OK** in the Format Legend dialog to apply these changes.
11. Right-click the chart and then select **Format Axes** > **Format Value (Y) Axis** from the shortcut menu.
12. In the Format tab of the Format Value (Y) Axis dialog, select **$#,##0** in the Number category, and then click **Add**.



13. Click **OK** in the Format Value (Y) Axis dialog.
14. Double-click the **Regions** label. In the Format Label dialog, switch to the **Font** tab, specify Font as **Arial** and Font Size to **10 pt** , and then click **OK**.
15. On the report tab bar, right-click the report tab and select **Rename** from the shortcut menu, then enter **AnnualSales** in the Input Report Tab Name dialog and click **OK**.
16. Click **File** > **Save** to save the report as **AnnualSalesbyRegion.cls**.
17. Click the **View** tab to preview the report and it looks somewhat as follows:

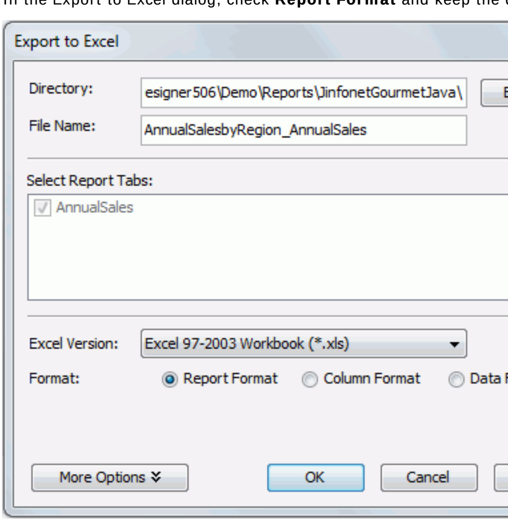


Note: If the report does not look correct, you can compare it to the final version of the report provided by JReport. To do so, you will need to save and close this catalog and then open the JinfonetGourmetJava.cat catalog file located at `<install_root>\Demo\Reports\TutorialReports`.

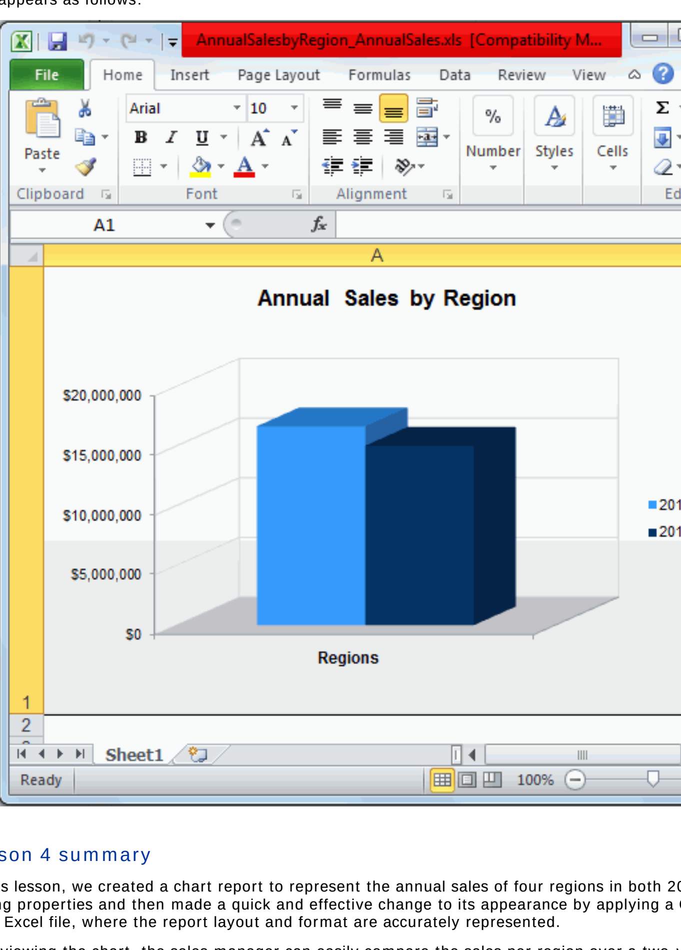
Task 3: Export the report to an Excel file

Now we can export the report to different formats as required. In this lesson, we will export it to an Excel file.

1. Click **File** > **Export To** > **Excel** on the menu bar.
2. In the Export to Excel dialog, check **Report Format** and keep the default settings of other options. Then click **OK**.



3. Open the file **AnnualSalesbyRegion_AnnualSales.xls** saved in `<install_root>\Demo\Reports\JinfonetGourmetJava`. It appears as follows:



Lesson 4 summary

In this lesson, we created a chart report to represent the annual sales of four regions in both 2013 and 2014, formatted the chart by setting properties and then made a quick and effective change to its appearance by applying a CSS. Finally, we exported the chart to an Excel file, where the report layout and format are accurately represented.

By reviewing the chart, the sales manager can easily compare the sales per region over a two-year period.

Lesson 5: Creating a table report

The Jetform Solution Suite allows you to create reports from customers in different territories around the world, and by reviewing this shipping information the Shipping department can receive bids from various shipping vendors. You are assigned the task of creating a shipment information report that lists the customer information (customer ID, order date, ship date, shipping cost, whether the payment has been received, and the shipping vendor) required to be reported by territory.

The following prototype of the report has been given to you:

```
Order ID Order Date Ship Date Ship Via Payment Received Shipping Cost
Territory :
Customer Name
XXXX  XXXXX  XXXXX  XXXX  Yes/No  XXXXX
XXXX  XXXXX  XXXXX  XXXX  Yes/No  XXXXX
XXXX  XXXXX  XXXXX  XXXX  Yes/No  XXXXX
Customer Name :
XXXX  XXXXX  XXXXX  XXXX  Yes/No  XXXXX
XXXX  XXXXX  XXXXX  XXXX  Yes/No  XXXXX
XXXX  XXXXX  XXXXX  XXXX  Yes/No  XXXXX
Territory :
Customer Name
XXXX  XXXXX  XXXXX  XXXX  Yes/No  XXXXX
XXXX  XXXXX  XXXXX  XXXX  Yes/No  XXXXX
XXXX  XXXXX  XXXXX  XXXX  Yes/No  XXXXX
```

This report will be run online, so an interactive report will work well. Interactive reports allow the end user to change the view of the report. They can filter the data on the report or change the sort order, as well as search the report.

This lesson contains the following tasks:

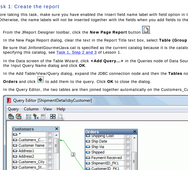
- Task 1: Create the table
- Task 2: Add a web control to the report
- Task 3: Add objects and edit their properties

Task 1: Create the report

Before taking this task, make sure you have enabled the Insert field name label with field option in the Options dialog as noted at the end of Lesson 2. Otherwise, the report labels will not be inserted together with the fields when you add fields to the table.

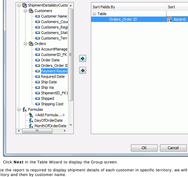
1. From the *Report Designer* toolbar, click the **New Page Report** button.
2. In the **New Page Report** dialog, clear the **Orbeon** in the Report Title text box, select **Table (Group Above)** from the layout box, then click **OK**.

   Be sure that *JetformCustomerCare* is specified as the current catalog because it is the catalog we use in this track. For information about specifying this setting, see Task 1, Step 2 and 3 of Lesson 1.
3. In the Data screen of the Table Wizard, click **+Add Query...** in the Queries node of Data Source 1, enter **ShipmentDetailsByCustomer** in the Input Query Name dialog and then click **OK**.
4. In the Add Table/View/Query dialog, expand the *JDBC* connection node and then the **Tables** node, then select the tables **Customers** and **Orders** and click **>** to add them to the query. Click **OK** to close the dialog.
5. In the Query Editor, the two tables are then joined together automatically on the Customers_Customer ID and Customer ID_FK1 columns.

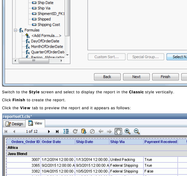


6. Check all the columns in the Orders table by selecting the **+** checkbox. For the Customers table, check the following columns: **Customer Name**, **Address1**, **Address2**, **Customers_Country**, **Customers_Territory** and **Customers_Region**.
7. Click **OK** at the bottom of the Query Editor to create the query. Then click **Next** in the Table Wizard.
8. In the Display screen, add the fields **Order_ID**, **Order_Date**, **Ship Date**, **Ship Via**, **Payment Received** and **Shipping Cost** to the table by selecting them and clicking button one by one.

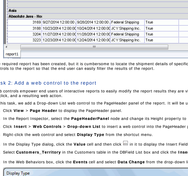
   By default, the records in a table are displayed randomly; they are displayed in the order they are returned from the fetch operation. You can specify the order in which the records are to be displayed by sorting the groups in table if any, according to your requirement. In this lesson, we will make the records in the table sorted by order ID ascending.
9. Click **Sort Fields By** button at the right bottom corner of the Table Wizard. In the Sort Fields By dialog, add the DBfield **Orders_Order ID** as the sort field by selecting it and clicking button, then click **OK** to return back to the table wizard.



10. Click **Next** in the Table Wizard to display the Group screen.
11. Select the report to organize the shipping shipment details of each customer in specific territory, we will add two groups to it: first group the report by territory and then by customer name.
12. In the Group screen, add the DBfield **Customers_Territory** as the first group by field and **Customer Name** the second one by selecting them and clicking button one by one.



13. Switch to the **Style** screen and select to display the report in the **Classic** style vertically.
14. Click **Finish** to create the report.
15. Click the **View** tab to preview the report and it appears as follows:



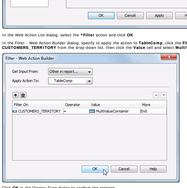
The required report has been created, but it is cumbersome to locate the shipment details of specific territories. So our next task is to add web controls to the report so that the end user can easily filter the results of the report.

Task 2: Add a web control to the report

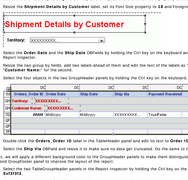
Web controls empower end users of interactive reports to easily modify the report results they are viewing, and are defined by a trigger event, such as click, and a resulting web action.

In this task, we add a drop-down List web control to the Pagemaker panel of the report. It will be used to filter the report records by territory.

1. Click **View** > **Page Header** to display the Pagemaker panel.
2. In the Report Inspector, select the **ShipmentDetailsByCustomer** node and change its height property to **1.25**.
3. Click **Insert** > **Web Controls** > **Drop-down List** to insert a web control into the Pagemaker panel.
4. Right click the web control and select **Display Type** from the shortcut menu.
5. In the Display Type dialog, click the **Value** cell and then click FM in it to display the Insert Fields dialog.
6. Select **Customers_Territory** in the Customers node in the DBfield List box and click the **Insert** button to add a value of the drop-down list.
7. In the Web Behaviors box, click the **Events** cell and select **Data Change** from the drop-down list, then click in the Actions cell.



8. In the Web Action List dialog, select the **Filter** action and click **OK**.
9. In the Filter - Web Action Builder dialog, specify to apply the action to **TableComp**, click the **Filter On** cell and select **CUSTOMERS_TERRITORY** from the drop-down list, then click the **Value** cell and select **MultiValueContainer**. Click **OK** to apply the settings.



10. Click **OK** in the Display Type dialog to confirm the settings.
11. Resize the drop-down list horizontally to make sure the territory name can be displayed completely.

By now, we have finished defining values of the web control. Next we will add a label ahead of it for identification, and a report title to the report.

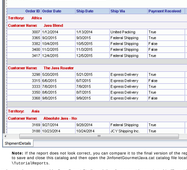
Task 3: Add objects and edit their properties

1. Drag and drop two Label objects from the Toolbox panel to the Pagemaker panel. Resize them and edit their text as follows:

   Shipment Details by Customer

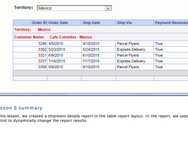
   Territory: XXXXXXXXX
2. Select the two labels by holding the Ctrl key on the keyboard and set their Bold property to **true** in the Report Inspector.
3. Resize the **Shipment Details by Customer** label, set its Font Size property to **18** and Foreground property to **Red** in the Report Inspector.

   Shipment Details by Customer
4. Select the **Order Date** and the **Ship Date** DBfields by holding the Ctrl key on the keyboard and set their Format property to **M/d/yyyy** in the Report Inspector.
5. Resize the two group by fields, add two labels ahead of them and add the text of the labels as "**Territory:**" for the first group by field and "**Customer Name:**" for the second one.
6. Select the four objects in the two GroupHeader panels by holding the Ctrl key on the keyboard, set their Foreground property to **Red0000**.



7. Double click the **Orders_Order ID** label in the Tablemaster panel and edit its text to **Order ID**.
8. Select the **Ship Via** DBfield and resize it to make sure no data get truncated. Do the same to the Payment Received label.

   Next, we will apply a different background color to the GroupHeader panels to make them distinguished from the Tablemaster panel and hide the second GroupHeader panel to improve the layout of the report.
9. Select the two TableGroupHeader panels in the Report Inspector by holding the Ctrl key on the keyboard and set their Background property to **Darkgray**.
10. Right-click on the second GroupHeader panel header and select **Hide** from the shortcut menu to hide the panel.



11. On the report tab bar, right-click the report tab and select **Rename** from the shortcut menu, then enter **ShipmentDetails** in the Input Report Tab Name dialog and click **OK**.
12. Click File > **Save** to save the report as **ShipmentDetailsByCustomer.cls**.
13. Click the **View** tab to preview the report and it appears as follows:

   Note: If the report does not look correct, you can compare it to the final version of the report provided by *iReport*. To do so, you will need to copy the *iReport* sample files and then open the *JetformCustomerCare.cat* catalog file located at *<install_root>/webapps/ROOT/iReport/*.

The web controls are powered by Page Report Studio and therefore cannot be previewed in *iReport Designer*. We can use the Preview as Page Report feature of *iReport Designer* to preview this report. However, this command is enabled only if the option Server for Previewing Reports was specified when installing *iReport Designer*.

Click **Preview As > Page Report Server**. The report will then be opened in a web browser. We can see select values from the drop-down list to dynamically change the report results. In the report below, we choose to view records in the Mexico territory.

Lesson 5 summary

In this lesson, we created a shipment details report in the table report layout. In the report, we used a table container to format the data, and a web control to dynamically change the report results.

Lesson 6: Creating a crosstab report

A report with quantities of different products that are sold in different regions and their sales totals is needed. You are required to create a report that can clearly represent the necessary information, which allows the sales manager to easily compare the sales of different products in different regions.

Here is the sketch that is given to you:

Product Sales Analysis

	Category 1	Category 2	Category 3	Category 4	Total
Country 1	xxx $XXXX.00	xxx $XXXX.00	xxx $XXXX.00	xxx $XXXX.00	xxx $XXXX.00
Country 2	xxx $XXXX.00	xxx $XXXX.00	xxx $XXXX.00	xxx $XXXX.00	xxx $XXXX.00
Country 3	xxx $XXXX.00	xxx $XXXX.00	xxx $XXXX.00	xxx $XXXX.00	xxx $XXXX.00
Country 4	xxx $XXXX.00	xxx $XXXX.00	xxx $XXXX.00	xxx $XXXX.00	xxx $XXXX.00
Total	xxx $XXXX.00	xxx $XXXX.00	xxx $XXXX.00	xxx $XXXX.00	xxx $XXXX.00

In this lesson, we learn to create a crosstab report from the following tasks:

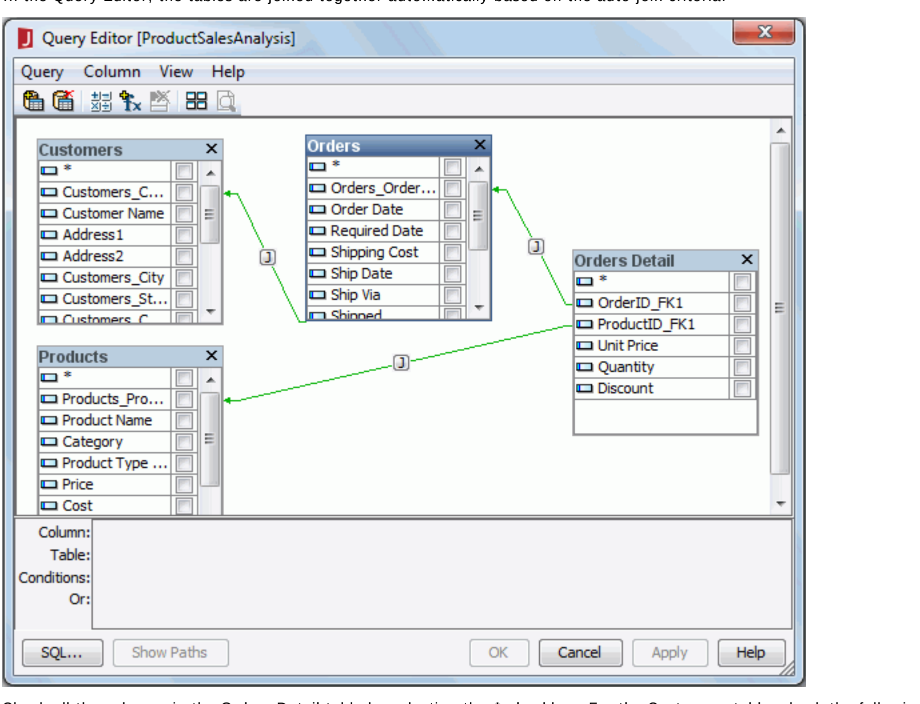
- Task 1: Create the crosstab
- Task 2: Format the crosstab
- Task 3: Change the page layout of the report
- Task 4: Save the report style as a CSS file

Task 1: Create the crosstab

Before taking this task, make sure you have enabled the Insert field name label with field option in the Options dialog as noted at the end of Lesson 3. Otherwise, the name labels will not be inserted together with the fields when you add fields to the crosstab.

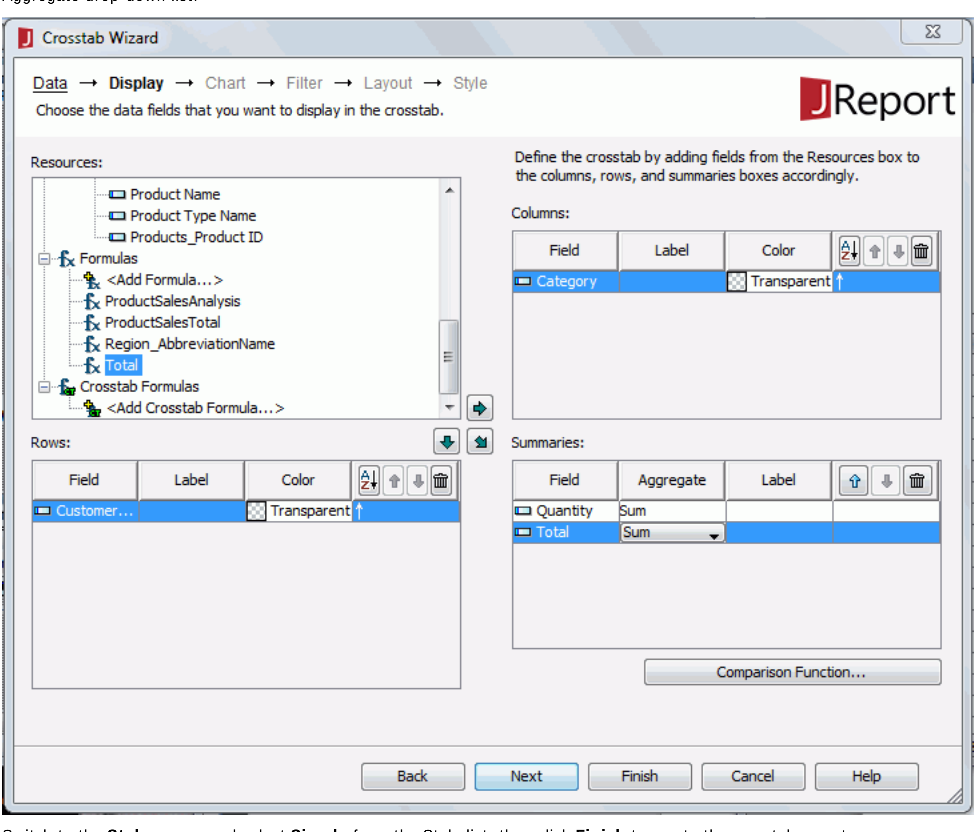
1. From the JReport Designer toolbar, click the New Page Report button.
2. In the New Page Report dialog, clear the text in the Report Title text box, select **Crosstab** from the layout box, then click **OK**.

   Be sure that JinfonetGourmetJava.cat is specified as the current catalog because it is the catalog we use in this track. For information about specifying this catalog, see Task 1, Step 2 and 3 of Lesson 1.
3. In the Data screen of the Crosstab Wizard, click **<Add Query...>** in the Queries node of Data Source 1, enter **ProductSalesAnalysis** in the Input Query Name dialog and click **OK**.
4. In the Add Table/View/Query dialog, expand the JDBC connection node and then the **Tables** node, then select the **Customers**, **Orders**, **Orders Detail** and **Products** tables and click to add them to the query. Click **OK** to close the dialog.
5. In the Query Editor, the tables are joined together automatically based on the auto join criteria.



6. Check all the columns in the Orders Detail table by selecting the * checkbox. For the Customers table, check the following columns: **Customer Name**, **Customers_City**, **Customers_State**, **Customers_Country**, **Customers_Territory** and **Customers_Region**, and for the Products table, the following columns: **Products_Product ID**, **Product Name**, **Category**, **Product Type Name** and **Price**.

   You may notice that here we do not check any columns in the Orders table, that is because in this report, columns in this table are not needed but we need this table to create joins between tables in the query.
7. Click **OK** at the bottom of the Query Editor to create the query.
8. In the Display screen, add the DBField **Category** to the Columns box by selecting it and clicking, add **Customers_Country** to the Rows box by selecting it and clicking, add the **Quantity** DBField and **Total** formula to the Summaries box by selecting them and clicking one by one, and then specify their aggregate function to **Sum** from the Aggregate drop-down list.

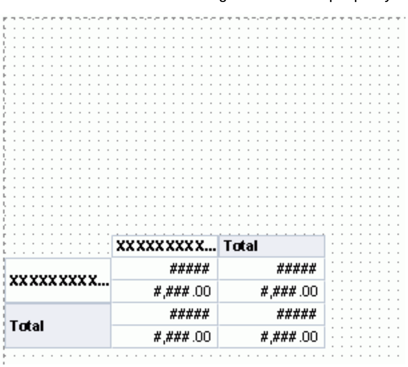


9. Switch to the **Style** screen and select **Simple** from the Style list, then click **Finish** to create the crosstab report.

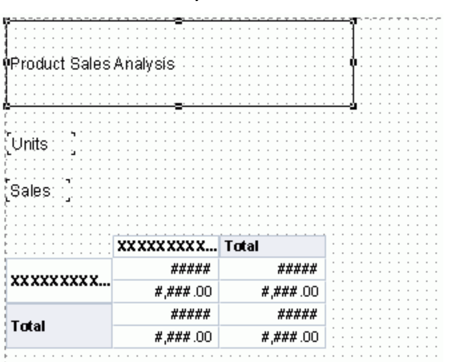
Task 2: Format the crosstab

In this task, we will format the crosstab to make it look more professional.

1. Select the crosstab and change its Position property to **absolute** in the Report Inspector, then drag it to the following position:

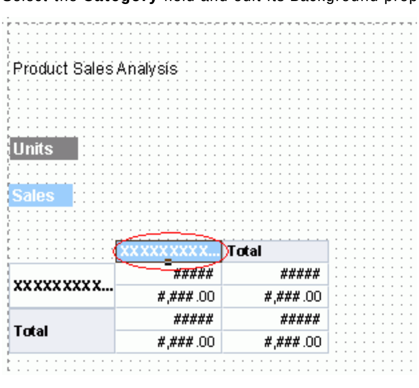


   By setting the Position property of an object to absolute, the object will be located at the position specified by dragging and dropping or by setting its X and Y coordinate property values. Meanwhile, when objects in a flow layout container have their Position property value set to absolute and are overlapped, we can set the display order of the objects by right-clicking the object and then selecting an item from the Move submenu.
2. From the Toolbox panel, drag three labels to the report, edit their text respectively to **Product Sales Analysis**, **Units** and **Sales**, then resize and adjust the crosstab and the three labels to align them as follows:

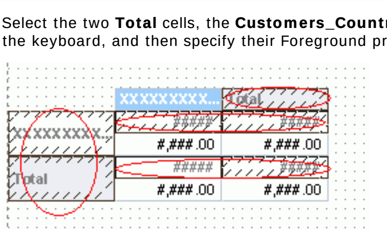


Next, we will edit properties of the crosstab report objects in the Report Inspector to improve the appearance of the report.

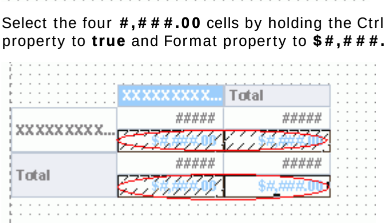
3. Select the **Units** label and edit its Bold property to **true**, Background property to **Gray** and Foreground property to **White**.
4. Select the **Sales** label and edit its Bold property to **true**, Background property to **0x99ccff** and Foreground property to **White**.
5. Select the **Category** field and edit its Background property to **0x99ccff** and Foreground property to **White**.



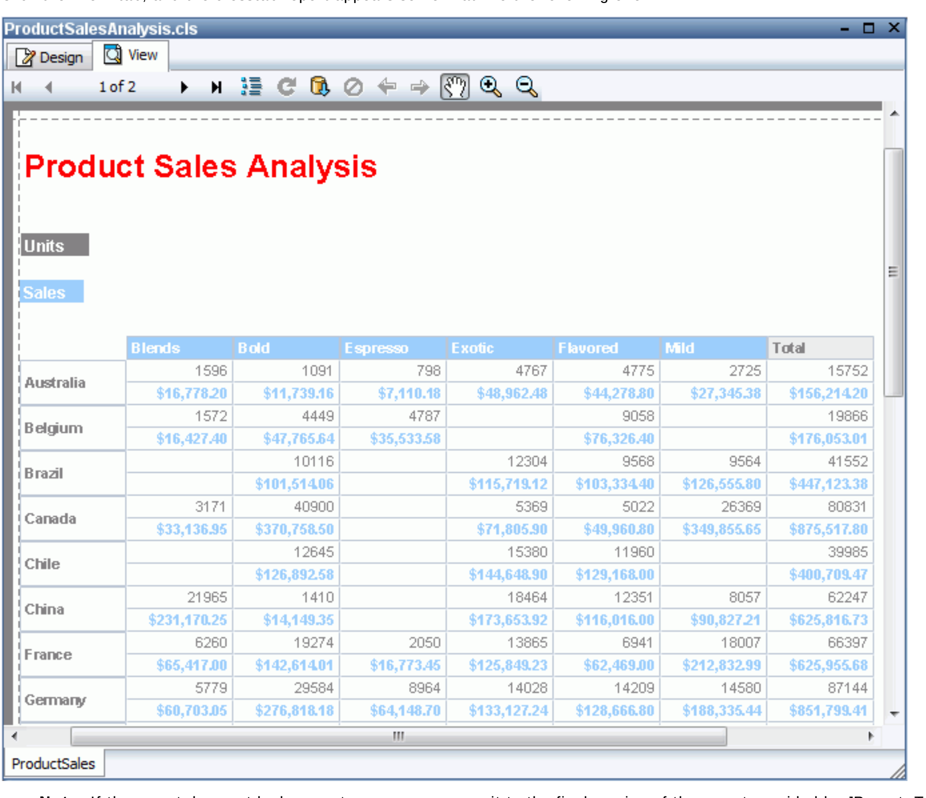
6. Select the two **Total** cells, the **Customers_Country** field and the four **#,###** cells in the crosstab by holding the Ctrl key on the keyboard, and then specify their Foreground property to **Gray**.



7. Select the four **#,###.00** cells by holding the Ctrl key on the keyboard, change their Foreground property to **0x99ccff**, Bold property to **true** and Format property to **$#,###.00**.



8. Select the **Product Sales Analysis** label, set its Bold property to **true**, Font Size property to **18**, and Foreground property to **Red**, then resize it to make its text not be truncated.
9. On the report tab bar, right-click the report tab and select **Rename** from the shortcut menu, then enter **ProductSales** in the Input Report Tab Name dialog and click **OK**.
10. Click **File** > **Save** to save the report as **ProductSalesAnalysis.cls**.
11. Click the **View** tab, and the crosstab report appears somewhat like the following one:



Note: If the report does not look correct, you can compare it to the final version of the report provided by JReport. To do so, you will need to save and close this catalog and then open the JinfonetGourmetJava.cat catalog file located at `<install_root>\Demo\Reports\TutorialReports`.

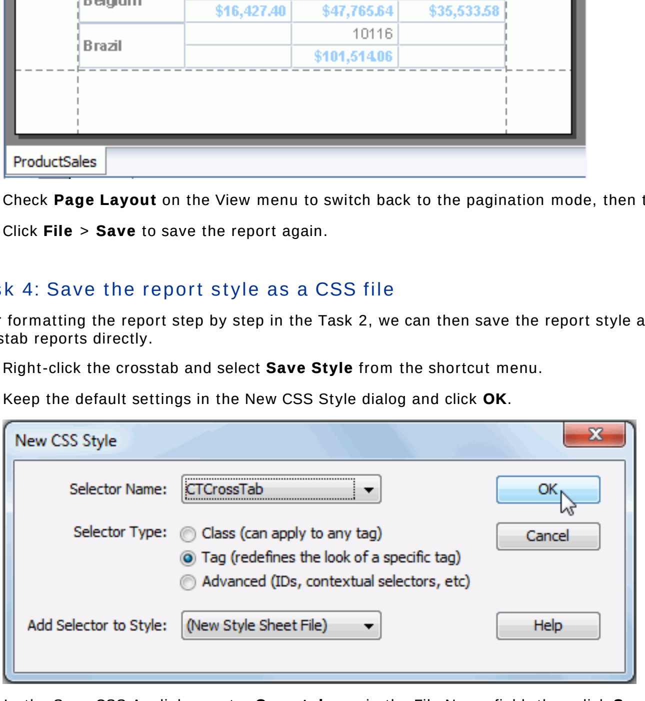
Task 3: Change the page layout of the report

When previewing the report, we can see that the crosstab is displayed in two pages. However, with JReport's page mode feature, we can decide how to view a report: in pagination mode or continuous mode. In pagination mode, the page shape and size can be specified in the page panel settings. In continuous mode, the whole report is laid out in a single page, and has no specific shape and size.

1. Uncheck **Page Layout** on the View menu.
2. Click the **View** tab to view the report again. Now, the crosstab is displayed in a single page. We can drag the scroll bar to have a complete view of the crosstab.

For a crosstab component, when it is in continuous page mode, we can further set another two properties to determine how many rows and columns we would like to view.

3. Select the crosstab, and set its Items per Row Block and Items per Column Block property values to **3** in the Report Inspector.
4. View the report. Now only three items are displayed in the row and column blocks.

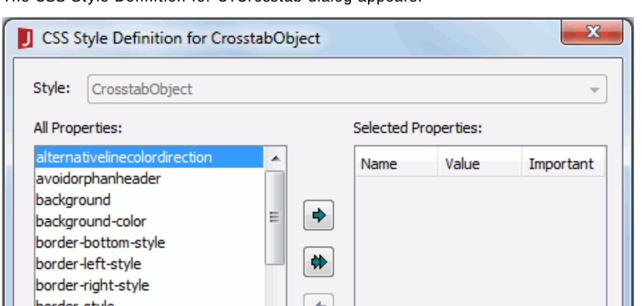


5. Check **Page Layout** on the View menu to switch back to the pagination mode, then the two properties will not take effect.
6. Click **File** > **Save** to save the report again.

Task 4: Save the report style as a CSS file

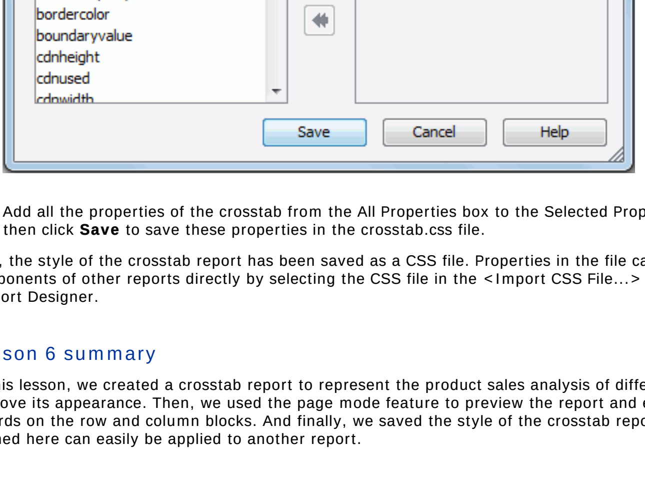
After formatting the report step by step in the Task 2, we can then save the report style as a CSS file, which can be applied to other crosstab reports directly.

1. Right-click the crosstab and select **Save Style** from the shortcut menu.
2. Keep the default settings in the New CSS Style dialog and click **OK**.



3. In the Save CSS As dialog, enter **Crosstab.css** in the File Name field, then click **Save** to save the file to the default directory `<install_root>\style`.

   The CSS Style Definition for CTCrosstab dialog appears.



4. Add all the properties of the crosstab from the All Properties box to the Selected Properties box by clicking the button, then click **Save** to save these properties in the crosstab.css file.

Now, the style of the crosstab report has been saved as a CSS file. Properties in the file can be applied to corresponding components of other reports directly by selecting the CSS file in the <Import CSS File...> drop-down list on the Format toolbar of JReport Designer.

Lesson 6 summary

In this lesson, we created a crosstab report to represent the product sales analysis of different regions, and then formatted it to improve its appearance. Then, we used the page mode feature to preview the report and edited some properties to reduce the records on the row and column blocks. And finally, we saved the style of the crosstab report as a CSS file so that the styles we defined here can easily be applied to another report.

Lesson 7: Creating a tabular report

A report about performance of products is needed, in which the product US is less than or equals to 10. The report includes annual sales of every product in every region. Each year, the quantities that are sold, and the total of their annual sales of every product. The report should contain a crosstab, which is to represent annual sales of ten products in every region, a chart object, which is to represent the quantities that are sold in every year for each product, and a tabular report, which is to represent the unit price and quantities.

Here is a sample of the report that you can refer to:

Product Performance

In this lesson, we will create a tabular report and insert crosstab object, chart object and banded object respectively into different sections of the report. We will also create a new data source based on the Sample.mdb database. In the data source, we will use the data need for creating the report from **product** in the *DWSample/Resources/Lesson* directory. Before creating the report, we will create a new folder for the report.

Follow the tasks below to create the report:

- Task 1: Create the report
- Task 2: Add components to the tabular report
- Task 3: Format the report

Task 1: Create the tabular report

1. Point to the menu in *Report Designer*, select **New > Page Report**.
2. In the New Page Report dialog, clear the text in the Report Title text box, select **Tabular** from the layout box, then click **OK**.
3. Right click the *Data Sources* folder in the Report Explorer, select **Add Data Source**, and rename it to the casing we use in this track. For example, we have data sampling stored in the folder *C:\Samples\Data\Sample.mdb*, then name it as *Sample*.
4. In the Tabular Wizard, keep the default settings and then click **Finish** to create a 2*2 tabular.

4. A message dialog appears prompting you to drag and drop data fields and components to the blank report. Click **OK** on the message dialog.

Now a blank report with a 2*2 tabular is generated. In the Report Inspector, they are respectively named Tablix1Cell and Tablix1Cell. (the name is the first line of the cell). Tablix1Cell and Tablix1Cell (the name is the second line of the cell). We can refer to the second one of the tablix1. We can then drag and drop fields and components to the cells.

The tablix1 fills the whole page panel, we can resize it by just dragging its cell borders. In this lesson, in order to make screen captures, we zoom out the tablix1 first.

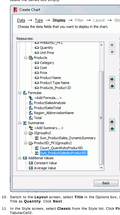
Task 2: Add components to the tabular report

Before taking this task, make sure you have enabled the Insert Ribbon tab and the Insert ribbon tab option in the Options dialog as noted at the end of Lesson 2. Otherwise, the name labels will not be inserted together with the ribbon you add fields to the report.

In this task, we will add three components - a crosstab, a chart and a banded object respectively to TabularCell1, TabularCell2 and TabularCell3 (the name is the second line of the cell) in the Report Inspector, respectively.

1. Select TabularCell and TabularCell2 (the name is the first line of the tabular) by holding the Ctrl key on the keyboard, right click and then select **Merge** from the shortcut menu.

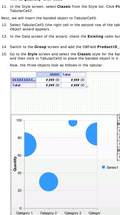
2. Select the merged tablix1 cell and click **Insert > Crosstab**. The Create Crosstab wizard appears.
3. In the Data screen of the wizard, select the query **ProductPerformance** from the Queries node of the Data Source 1, then click **Next**.
4. In the Display screen, add the **Products_Product ID** DiField to the Columns box by selecting it and clicking button, and the **Region_AddressRegion** to the Rows box by selecting it and clicking button (the information about creating the formula, see Task 1, Step 5 and 6 of Lesson 4), add the **Total** formula to the Summaries box by selecting it and clicking button, and then select the Grand Totals for Columns and Rows from the Aggregate drop-down list.



5. Search in the **Legend** screen, select **Title** in the Options box, then specify Category(X) Axis Title as **Product ID**, and Value(Y) Axis Title as **Quantity**, then click **Next**.
6. In the Style screen, select **Classic** from the Style list. Click **Finish** to close the Create Chart wizard and drop the chart in TabularCell2.

Now, we will insert the banded object to TabularCell3.

7. Select TabularCell3 (the name left on the second row of the tabular) and click **Insert > Chart**. The Create Chart wizard appears.
8. In the Data screen of the wizard, check the **Existing** radio button, select the query **ProductPerformance**, then click **Next**.
9. In the Type screen, select the **Bubble 2-D** type and click **Next**.
10. In the Series screen, add the **Quantity**, **ProductID** DiField to the Columns box by selecting it and clicking button, and the **Region_AddressRegion** to the Rows box by selecting it and clicking button (the information about creating the formula, see Task 1, Step 5 and 6 of Lesson 4), add the **Total** formula to the Summaries box by selecting it and clicking button, and then select the Grand Totals for Columns and Rows from the Aggregate drop-down list.

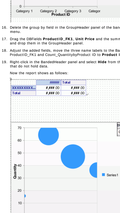


11. Search in the **Legend** screen, select **Title** in the Options box, then specify Category(X) Axis Title as **Product ID**, and Value(Y) Axis Title as **Quantity**, then click **Next**.
12. In the Style screen, select **Classic** from the Style list. Click **Finish** to close the Create Chart wizard and drop the chart in TabularCell3.

Now, we will insert the banded object to TabularCell3.

13. Select TabularCell3 (the name left on the second row of the tabular) and click **Insert > Banded Object**. The Create Banded Object wizard appears.
14. In the Data screen of the wizard, check the **Existing** radio button, select the dataset **ProductPerformance**.
15. Switch to the **Group** screen and add the DiField **ProductID_PK1** as the group by field by selecting it and clicking button.
16. Go to the **Sort** screen and select the **Count** style for the dataset object. Click **Finish** to close the Create Banded Object wizard and then click in TabularCell3 to place the banded object in it.

Now, the three objects look as follows in the tablix:



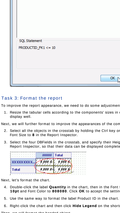
17. Delete the group by field in the Groupheader panel of the banded object by right-clicking it and selecting **Delete** from the shortcut menu.
17. Drag the DiFields **ProductID_PK1**, **Unit Price** and the summary **Count_QuantityOfProduct ID** from the Resource View panel into the Groupheader panel.
18. Adjust the added fields, move the three name labels in the BandedHeaderpanel panel, and then change the names of ProductID_PK1 to **Product ID**, Count_QuantityOfProduct ID to **Quantity**.
19. Right-click in the BandedBodyPanel panel and select **Hide** from the shortcut menu to hide it from view. Repeat this to hide all panels that do not hold data.

Now the report shows as follows:

Now, we need a report title for the tabular report.

20. Right-click the blank space of the report that holds the crosstab, and then select **Insert Row Above** from the shortcut menu, and then resize the rows to a suitable height.
21. Drag the Text box from the expanded tabular row, and enter the text **Product Performance by Product ID** as the report title, then adjust the text to a suitable size and set it as **Fast** from the Format menu.
22. In the Fast tab of the Format float dialog, set Fast to **Arial**, Fast Size to **18**, Fast Color to **Red** and Bold to **true**, then click **OK**. The title appears as follows:

Product Performance by Product ID.



In this lesson, we will use the Filter feature to limit data selected from the dataset to view the products performance from the Product ID 1 to 10. Since all the components share the same dataset, the filter will be applied to all of them.

23. Click the **Dataset Filter** button of the toolbar of the Resource View panel.
24. In the Dataset Filter dialog, click the **Add Condition** button to add a filter line, select **ProductID_PK1** from the field drop down list, **Less** from the operator drop down list, and enter **10** in the value text field to specify the filter condition as **ProductID_PK1 < 10**.

Task 3: Format the report

To improve the report appearance, we need to do some adjustments to the report, as well as the components in it:

1. Resize the tabular cells according to the components' sizes in every cell, so that they will not be truncated by the tabular and can be seen clearly.
2. We will further improve the appearances of the components. First, we will format the crosstab.
2. Select all the objects in the crosstab by holding the Ctrl key on the keyboard and then set their Fore Ground property to **Arial** and Fore Size to **8** in the Report Inspector.
3. Select the four DiFields in the crosstab, and specify their Height to **0.25**, Width property to **0.84**, and Format to **#,##.#** in the Report Inspector, so that their data can be displayed completely in the cells.

Now, we have the table:

4. Select the table **Quantity** in the chart, then in the Fore tab of the Fore Label dialog, set Fore Face to **Arial**, Fore Size to **10pt** and Fore Color to **RGB(0, 0, 255)**. Click **OK** to accept the changes.
5. Use the same way to format the label Product ID in the chart.
6. Right-click the chart and then click **Hide Legend** on the shortcut menu to hide the legend.

Then, we will format the banded object.

7. Select all the objects in the banded object by holding the Ctrl key on the keyboard and then specify their Fore Face property to **Arial** and Fore Size to **8** in the Report Inspector.
8. Select the three DiFields in the Groupheader panel by holding the Ctrl key on the keyboard, set their Foreground property to **Black** and Bold to **false** in the Report Inspector.
9. Select the **Unit Price** DiField and then specify its Format property to **#,#.#** in the Report Inspector.

After doing the adjustments, the report appears somewhat like follows in design view:

Product Performance by Product ID.

According to the sketch, a few needs to be added below the report title. We can do this just by setting tabular cell properties.

1. Select the tabular cell holding the report title, then in the Report Inspector, specify its Top Line, Left Line and Right Line properties to **none**, Border Color to **Gray** and Border Thickness to **0.01**.
2. Select the tabular cell holding the crosstab, and specify its Top Line property to the line we just customized and the other two components look too close. We need to do some adjustments, and set the above value by using the function of a tabular.
11. Right click the tabular cell holding the crosstab, and then select **Insert Row Above** from the shortcut menu. Right click the cell again and then select **Insert Row Below**.

Two rows are now added above and below the crosstab.

12. Resize the tabular rows to adjust the distances between the crosstab and the other components.
13. Open the report tab bar, right-click the report tab and select **Rename** from the shortcut menu, then enter **ProductPerformance** in the Text Report Tab Name dialog and click **OK**.
14. Click **File > Save** to save the report as **ProductPerformance.ProductID.rdl**.
15. Click the **View** tab, and the report appears as follows, depending on what formatting you have done to the components:

Product ID	Unit Price	Quantity
1	$9.90	17679
2	$11.15	13245
3	$8.00	41605
4	$10.45	19750
5	$16.45	17418
6	$10.45	19874
7	$16.95	13226
8	$11.15	33224
9	$11.45	5941
10	$13.45	28859

Note: If the report does not look correct, you can compare it to the final version of the report provided by *Report*. To do so, open the folder that contains those files catalog and then open the *DWSampleProducts.rdl* catalog file located at *<install>_samples\Reports\Tutorial\Tutorial*.

Lesson 7 summary

In this lesson, we created a tabular report that contains a crosstab object, a chart object and a banded object. In represented the unit price, quantity and annual sales, and sales total in every region about the products, of which the product US is less than or equals to 10. The category(X) axis of the chart represent the product ID, value(Y) axis showed the quantities, and the size of bubble showed the annual sales.

Lesson 8: Creating a report that contains a subreport

Here we need to create a report that contains a subreport to show the order detail information, the related customers and employees information.

This lesson shows a more complicated and another report. When adding report to mini report 8, the report is referred to as a subreport, while report 8 is considered as a primary report. We will first create a page report that contains two report tabs - one is for the order information and the other is for the customer information. Then we will create a subreport.

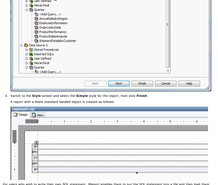
In this lesson, we have the report table set in the page report as the primary report, and the second one as the subreport. So that we can insert the second report tab to the first one to show the sales total by orders in different order ID groups.

- Task 1: Create a page report that contains two report tabs
- Task 2: Insert a subreport to the report

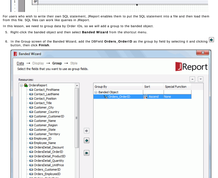
Task 1: Create a page report that contains two report tabs

Before taking this task, make sure you have enabled the Insert Report tab under Report Tab editor in the Options dialog as *noted at the end of Lesson 6*. Also make sure that you have the second chapter with the tables where you add fields to the report.

1. Click **File > New > Page Report** on the menu bar.
2. In the New Page Report dialog, clear the check in the Report Title text box, select **Banded** the layout box, then click **OK**.
   Note: The item DefaultDataSource.rsd is specified as the current catalog because this is the catalog we use in this task. For more information, see Lesson 1 Section 1 Step 1 and Lesson 1 Section 2 Step 1.
3. In the Data screen of the Banded Wizard, select **OrderReport** from the Inspector fully code of Data Source 1.



4. Switch to the **Style** screen and select the **Empty** style for the report, then click **Finish**.
   A report with a Detail banded control appears in the designer.



Put some data work to write their own SQL statement. (Report enables them to put the SQL statement into a file and then load them into the SQL editor. Here we use the SQL statement in the report.)

In this lesson, we need to group data by Order ID, so we will add a group to the banded object.

5. Right-click the banded object and select **Banded Wizard** from the shortcut menu.
6. In the Group screen of the Banded Wizard, add the DBfield **Orders_OrderID** as the group by field by selecting it and clicking **>>** button, then click **Finish**.

7. Place the DrawDetailBand panel of the banded object, drag the fields **Customer_Name**, **Order_OrderDate**, **Contact_LastName**, **Contact_FirstName**, **Employee_LastName**, **Employee_FirstName** and **Order_Total** to the Detail band.
   Rename the DBtext objects to **Customer_Name**, **Order_Date**, **Contact_LastName** and **Contact_FirstName** respectively.
   Rename the DBtext objects **Employee_LastName** and **Employee_FirstName** respectively.
8. Place the newly added column **Order_Total** to the Footer section of the Banded Wizard.

9. Rename the DrawDetailBand panel of the banded object, drag the fields **Customer_Name**, **Order_OrderDate**, **Contact_LastName**, **Contact_FirstName**, **Employee_LastName**, **Employee_FirstName** and **Order_Total** to the Detail band.
   Rename the DBtext objects to **Customer_Name**, **Order_Date**, **Employee Last Name** and **Payment Received** one by one.
10. Place the newly added column **Order_Total** to the Footer section of the Banded Wizard.

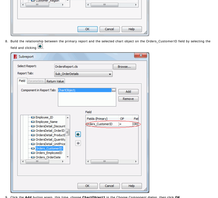
11. Place the group by field in the DrawHeader panel, click **Insert > Label** to add a label ahead of it and edit its text to **Order Id**.

12. Select the **Order Date** DBfield and set its Format property to **M/d/yyyy** in the Inspector.
13. Select the BandedPageHeader panel, click **Insert > Label** to add a label in it and edit its text to **Order Details**. The label will be used as the report header.
14. Select the Customer_Name, Order_OrderDate, BandedPageHeader and BandedHeader panels one by one by right-clicking them and selecting **Visible** from the shortcut menu.
15. Format labels and DBtext objects to improve their appearance. Here, we will not describe the formatting process in detail. Refer to Task 2 of Lesson 1 to review.
   Now the report table in the report is created to show the order detail information as follows in design mode:

   And the report looks as follows in view mode:

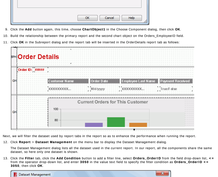
16. Select the second tabular cell and insert a pie chart in it which uses the same dataset and displays the same data in the same room with the table.
17. Right-click the bar chart and select **Add Label** from the shortcut menu. A label is then added to the chart.

18. Select the tabular cell and insert a pie chart in it which uses the same dataset and displays the same data in the same room with the table.
19. Right-click the bar chart and select **Add Label** from the shortcut menu. A label is then added to the chart.



20. Move the label in the top of the chart, then right-click it and choose **Format Label** from the shortcut menu to display the Format Label dialog box.
21. In the General tab, change the label text to **Current Orders For This Customer**.
22. Switch to the **Fill** page, fill the chart's color to **Aqua**, set the Fore Color to **LIgt**, then Color to **AliceBlue** and click **OK**.
23. Add a label to the pie chart and format it using the same way and edit its text to **Current Orders For This Employee**.
24. Select **ChartReport** and **ChartReport1** in the Report Inspector by holding the Ctrl key on the keyboard and change their Border property to **False**. Then, move the two charts to the right side of the table. Make the report table and the two charts aligned.
   Now the report table in the report is ready for the second reporting objects, that we need to add a second group header and add the subreport to show the sales total by order ID.

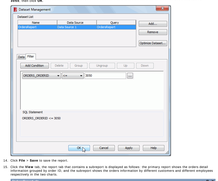
   Now the report table in the report is created to show the sales total by orders.



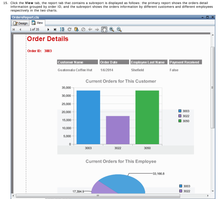
Task 2: Insert a subreport to the report

After creating the report, we can insert the subreport to the primary report by specifying relationships between them. In this lesson, we can specify different relationships between the primary report (the first one) and the sub-chart (components in the second report) by using the Subreport Wizard. The Subreport Wizard provides a convenient way to create a subreport. In this lesson, we use the Subreport Wizard to ask the corresponding customer and employee information of every order.

1. Select **OrderReport** to be at the end of the existing reports, then we need to add a second group header and add the subreport to show the sales total by order ID.
2. Click **OrderReport/Information** on the report table tab to switch to the report tab.
   OrderReport/Information is a new report tab.
3. Right-click the Subreport panel of the banded object and choose **Insert Panel After** from the shortcut menu.
4. Select the newly added panel and click **Insert**.
   A new subpanel is attached to the mouse cursor, click the newly added panel to place the subreport into it, and the Subreport dialog appears.
5. Click the **Browse** button, select **OrderReport.rdl** and then click **Open**.
6. Select **Sales Order Details** from the Report Tab drop down list.
7. Click the **Add** button beside the Component in Report Tab box in the Field tab, choose **ChartObject** in the Choose Component dialog box, then click **OK**.



8. Build the relationship between the primary report and the selected chart object on the Orders_CustomerID field by selecting the field and pressing **Add**.



9. Click the **Add** button again, this time, choose **ChartObject** in the Choose Component dialog, then click **OK**.
10. Build the relationship between the primary report and the second chart object on the Orders_EmployeeID field.
11. Click **OK** in the Subreport dialog and the report tab will be inserted in the OrderReport report tab as follows:

   Note: If the report does not load correctly, you can compare it to the final version of the report provided by *Report*. To do so, you will need to save and close this catalog and then open the *DefaultDataSource.rsd* catalog file located at *<install_root>\Samples\Reports\Tutorial\Report*.

Lesson 9 Summary

In this lesson, we created a report that contains a subreport. First we created a page report with two report tabs and the second report with two report charts of different types. Then we inserted the second report tab as a subreport, in the first report tab by setting up the relationship between the two reports. We also created a report that contains a subreport and specified the relationship between the two reports.

As you learned in this lesson, it is very convenient to build up a library of reports that act as components that you can add to various primary reports. Properly designed and used, subreports save a lot of time and effort in reports adapting your applications to changing business requirements.

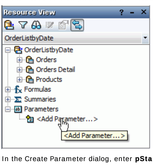
Lesson 9: Creating a parameter-based report

In Lesson 1 of this track we created a page report that contains an order list report tab to show order lists of every month from 2014 to 2015. Now, the sales manager is asking for this same report but with different orders during a specified time. Instead of building a new report about orders in each order date, we will save the page report as a new one and then modify the dataset used by its report to meet the requirements.

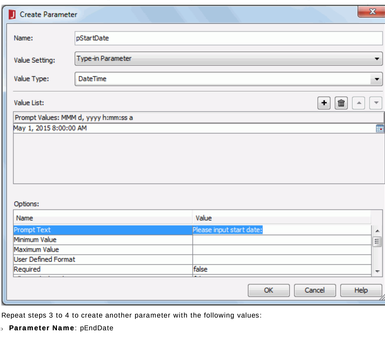
- Task 1: Create the parameters
- Task 2: Modify the dataset of the report to use the parameters
- Task 3: Preview and test the report

Task 1: Create the parameters

1. Click **File** > **Open** to open the report OrderListbyDate.cls created in Lesson 1 in the JinfonetGourmetJava.cat catalog file.
2. Click **File** > **Save As** to save the report as **OrderListbyDate_Parameter.cls**. Then we begin to modify this report instead of the former one.
3. In the Resource View panel, click the **<Add Parameter...>** item in the Parameters node.



4. In the Create Parameter dialog, enter **pStartDate** in the Parameter Name field, specify Value Setting as **Type-in Parameter** and Value Type as **DateTime**, click the **Add** button to add a Value List field, enter "**May 1, 2015 8:00:00 AM**" as the prompting value, and "**Please input start date:**" as the prompting text. Then click **OK**.

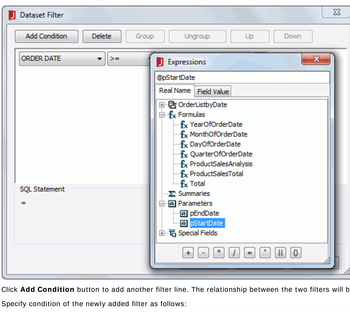


5. Repeat steps 3 to 4 to create another parameter with the following values:
   - **Parameter Name**: pEndDate
   - **Value Setting:** Type-in Parameter
   - **Value Type**: DateTime
   - **Prompting Values**: June 1, 2015 8:00:00 AM
   - **Prompting Text**: Please input end date:

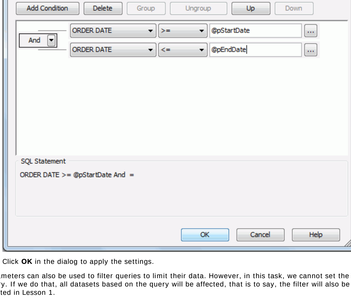
Task 2: Modify the dataset of the report to use the parameters

In this task, we will apply the parameters created above to filter the dataset which is used by the report. Then when we view the new report, a message will prompt for us to specify the parameter values. In this way, we can get data that we need.

1. On the Resource View panel toolbar, click the **Dataset Filter** button.
2. In the Dataset Filter dialog, click the **Add Condition** button to add a filter line, select **ORDER DATE** from the field drop-down list, set **>=** as the operator, then click
3. In the Expressions dialog, double-click the **pStartDate** parameter to use it as value of the filter condition. Then close the Expressions dialog.



4. Click **Add Condition** button to add another filter line. The relationship between the two filters will be And.
5. Specify condition of the newly added filter as follows:



6. Click **OK** in the dialog to apply the settings.

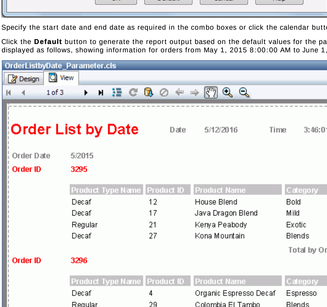
Parameters can also be used to filter queries to limit their data. However, in this task, we cannot set the filter on the report's query. If we do that, all datasets based on the query will be affected, that is to say, the filter will also be applied to the report created in Lesson 1.

Task 3: Preview and test the report

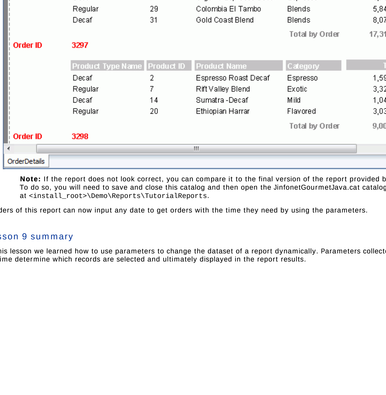
After the steps above, the parameters will take effect when viewing the report by entering values in the Enter Parameter Values dialog.

1. Click the **View** tab to preview the report.

   JReport recognizes that the query is based on the value of two parameters, and therefore prompts for them as follows:



2. Specify the start date and end date as required in the combo boxes or click the calendar button to select the date.
3. Click the **Default** button to generate the report output based on the default values for the parameters. The report is displayed as follows, showing information for orders from May 1, 2015 8:00:00 AM to June 1, 2015 8:00:00 AM only:



Note: If the report does not look correct, you can compare it to the final version of the report provided by JReport. To do so, you will need to save and close this catalog and then open the JinfonetGourmetJava.cat catalog file located at `<install_root>\Demo\Reports\TutorialReports`.

Readers of this report can now input any date to get orders with the time they need by using the parameters.

Lesson 9 summary

In this lesson we learned how to use parameters to change the dataset of a report dynamically. Parameters collected at runtime determine which records are selected and ultimately displayed in the report results.

Track 2: Creating business views and web reports

JReport provides the web reporting solution that 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. Web reports are viewed using a new interactive viewer called Web Report Studio. Web Report Studio provides a much nicer end user experience with many powerful features for interfacing with a report such as changing parameters without re-running the report.

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, report developers can create, open and edit web reports. Web reports created on JReport Server can also be downloaded to JReport Designer for further editing.

In this track we will create a business view and a web report based on the business view.

- Lesson 1: Creating a business view
- Lesson 2: Creating a web report using JReport Designer

Note: A JReport Live license for JReport Designer is required in order to perform this track. If you do not have the license, please contact your Jinfonet Software account manager to obtain it first.

Click the right arrow ( ) below to begin reading the track.

Lesson 1: Creating a business view

Business views are created and managed in *Report Designer*. A business view is built from four kinds of elements:

- **Group objects** are typical DBFields by which you want to group data. They present the availability and key performance of data, and characteristics return text data or dates, and answer the following question: who, when, what, where and which.
- **Aggregation objects** are typically numeric DBFields to which an aggregate function, such as Sum, is applied or they can be existing calculations, such as Quantity * Price.
- **Detail objects** are DBFields, typically they are values that you would want to display in the Detail section of a table. But they can be any type of DBField.
- **Categories** are simply folders for organizing the other three elements.

In this lesson, we will create a business view based on data in Dana Source of a JinformeSourceNewJava cat catalog.

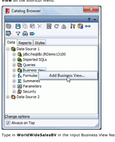
This lesson contains the following tasks:

- Task 1: Create the business view
- Task 2: Add elements to the business view
- Task 3: Define hierarchies in the business view

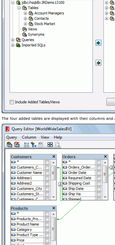
Task 1: Create the business view

In this lesson we will create a business view which provides data information about customers, orders, orders detail, and products, therefore we will make the business view contain three tables and our relationship among the tables.

1. Click **Start > All Programs > iReport 1.5 Update 2 > Report Designer** to start iReport Designer. The iReport Designer window and the Welcome to iReport Designer dialog appear.
2. In the Welcome to iReport Designer dialog, click the **Catalog** link in the Open category.
3. Browse to select the **JinformeSourceNewJava.cat** catalog file in <install_root>/demo/iReport/jinformeSourceJava, then click the **Open** button. The Catalog Browser is displayed.
4. In the Data tab of the Catalog Browser, expand the **Data Source 1** node, then right click the **Business View** node and click **Add Business View** on the shortcut menu.



5. Type in **WorldWideSalesW** in the Input Business View Name dialog and click **OK**.
6. In the Add Table/View/Query dialog, select these tables under the JDBC connection: Customers, Orders, Orders Detail and Products, click to add them to the right-hand pane, then click **OK**.



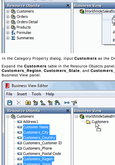
7. The four added tables are displayed with their columns and automatically joined based on the auto join criteria in the Query Editor.

8. In the Customers table, check the * checkbox to select all columns in the table, then uncheck these columns: **Address2**, **Customers_Fax**, and **Address3**. In the Orders table, select **Orders_Order ID**, **Order Date**, and **Payment Received**. In the Orders Detail table, select **Unit Price**, **Quantity**, and **Discount**. In the Products table, select all its columns first by checking *, then unselect **Inventory** and **Reorder Level**.
9. Click **OK** at the bottom of the Query Editor to create the Business View.
10. The Business View Editor is displayed.

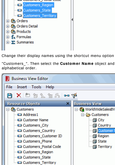
Task 2: Add elements to the business view

We will create three categories in the business view to hold customer, order and product information respectively.

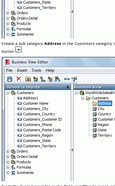
1. In the Business View panel of the Business View Editor, select **WorldWideSalesW** and right click, then select **Add Category** from the shortcut menu.



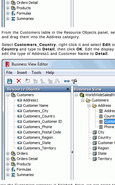
2. In the Category Property dialog, input **Customers** as the Category name, then click **OK**.
3. Expand the **Customers** table in the Resource Objects panel, select the fields **Customer Name**, **Customers_City**, **Customers_Country**, **Customers_Phone**, **Customers_State**, and **Customers_Territory** in the table, then drag them into the Customers category in the Business View panel.



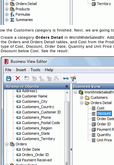
4. Change their display names using the shortcut menu option Rename to **City**, **Country**, **Region**, **State**, and **Territory**, by removing "Customers_" Then select the **Customer Name** object and click to move it below Country to make the objects displayed in alphabetical order.



5. Create a sub category **Address** in the Customers category to hold address information. Move it above the City group object by using the button.

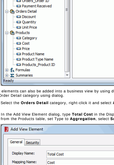


6. Move the Customers table in the Resource Objects panel, select **Address1**, **Customers_Country**, **Customer Name**, and **Customers_Phone** and drag them into the Address category.
7. Select **Customers_Country**, right click and select **Edit** on the shortcut menu. In the Edit-View-Element dialog, edit its display name to **Country** and type in **Country** then click **OK**. Edit the display name of Customers_Phone to **Phone** and type in **Detail** using the same way. Also edit the type of Address1 and Customer Name to **Detail**.



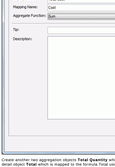
By now the Customers category is finished. Next, we are going to create the Orders Detail category.

12. Create a category **Orders Detail** in WorldWideSalesW. Add these fields from the Resource Objects panel into the category: all the fields from Orders Detail, Order Date, Payment Received, and Order ID from Orders, and Product Name, Product Type, Product ID, Cost, Unit Price, Quantity, type of Cost, Discount, Order Date, Quantity and Unit Price to **Detail**. Arrange the order of the new elements. Move Date to the top, move Discount below Cost. See the result:

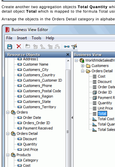


Now elements can also be added into a business view by using dialog instead of dragging and dropping. Next, we are going to add more objects to the Order Detail category using dialog.

13. In the **Orders Detail** category, right click it and select **Add View Element** from the shortcut menu.
14. In the Add View Element dialog, type **Total Cost** in the Display Name text field, click next to the Mapping Name text field to select **Cost** from the Product table, set Type to **Aggregation**, select **Sum** from the Aggregate Function drop down list, then click **OK**.

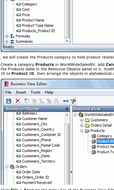


15. Create another two aggregation objects **Total Quantity** which sums on Quantity and **Total Sales** which sums on the formula Total, and a detail object **Total** which is mapped to the formula Total using the same way.
16. Arrange the objects in the Orders Detail category in alphabetical order.



Next, we will create the Products category to hold product related information.

17. Create a category **Products** in WorldWideSalesW, add **Category**, **Product Name**, **Product Type Name**, and **Products_Product ID** from Products, and **Product ID** from Product in the Resource Objects panel. Display names of Product Type Name to **Product Type** and Products_Product ID to **Product ID**. Then arrange the objects in alphabetical order.



18. Click **File > Save** on the menu bar of the Business View Editor window to save the business view.

Task 3: Define hierarchies in the business view

In this task, we define some hierarchies in the business view. Hierarchies enable drilling up and down through report data.

1. In the Business View Editor, select the root node of the business view in the Business View panel, then click **Insert > Hierarchy** on the menu bar.
2. In the Add Hierarchy dialog, input **Geography** and click **OK**. The Geography hierarchy node will be added at the bottom under the business view node.
3. In the Business View panel, drag three objects **Region**, **Territory**, **Country**, **State** and **City** from the Customers category one by one into Geography. Arrange the order of the new elements. Move Region to the top, move City to the bottom.
4. Create a new hierarchy named **Product**, then add **Product Type**, **Category** and **Product Name** to the hierarchy.

Now, two hierarchies have been defined in the business view.

5. Save the business view to save the hierarchies.

Lesson 1 summary

In this lesson, we created a business view based on four tables, add fields and define hierarchies in it.

⇦⇨

Lesson 2: Creating a web report using JReport Designer

In this lesson we will create a web report as the role of a report developer, based on the business view WorldWideSalesBV created in the previous lesson. The report will show each product's sales information, including order ID, order date, quantity, unit price, discount, and the total sales for each order. We will use a table to display the information, plus a chart to demonstrate the total sales of each product. In order for end users to see sales information within a specific range, such as region, country, category, product name, or product type, we will add filter controls for dynamically filtering the data according to their requirements.

This lesson contains the following tasks:

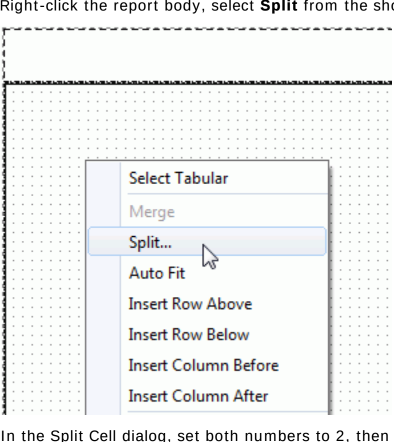
- Task 1: Create the initial report
- Task 2: Format the web report
- Task 3: Preview the web report

Task 1: Create the initial report

1. On the JReport Designer menu bar, click **File** > **New** > **Web Report**. A blank report is created.

   Be sure that JinfonetGourmetJava.cat is the current catalog because it is the catalog where WorldWideSalesBV is created. Click here for information about specifying this catalog.

2. Right-click the report body, select **Split** from the shortcut menu.



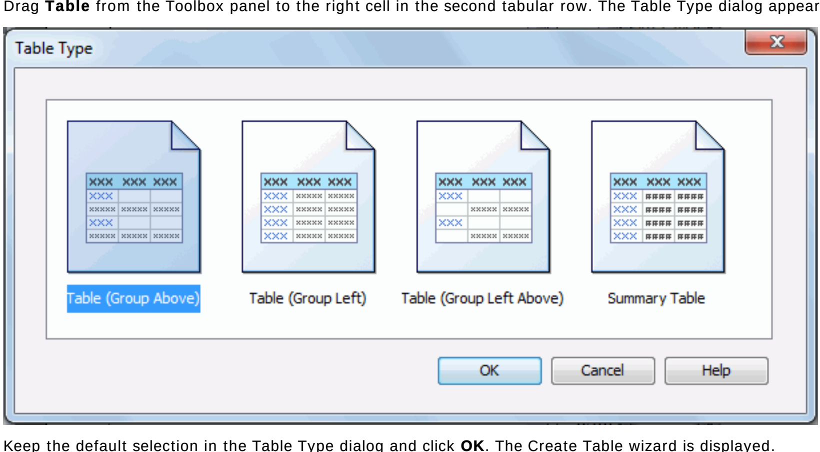
3. In the Split Cell dialog, set both numbers to 2, then click **OK**.

The report body will be hosted by four tabular cells. We will put the filter controls in the left cells, a chart and a table in the right cells. First we create a chart to demonstrate the total sales of each product.

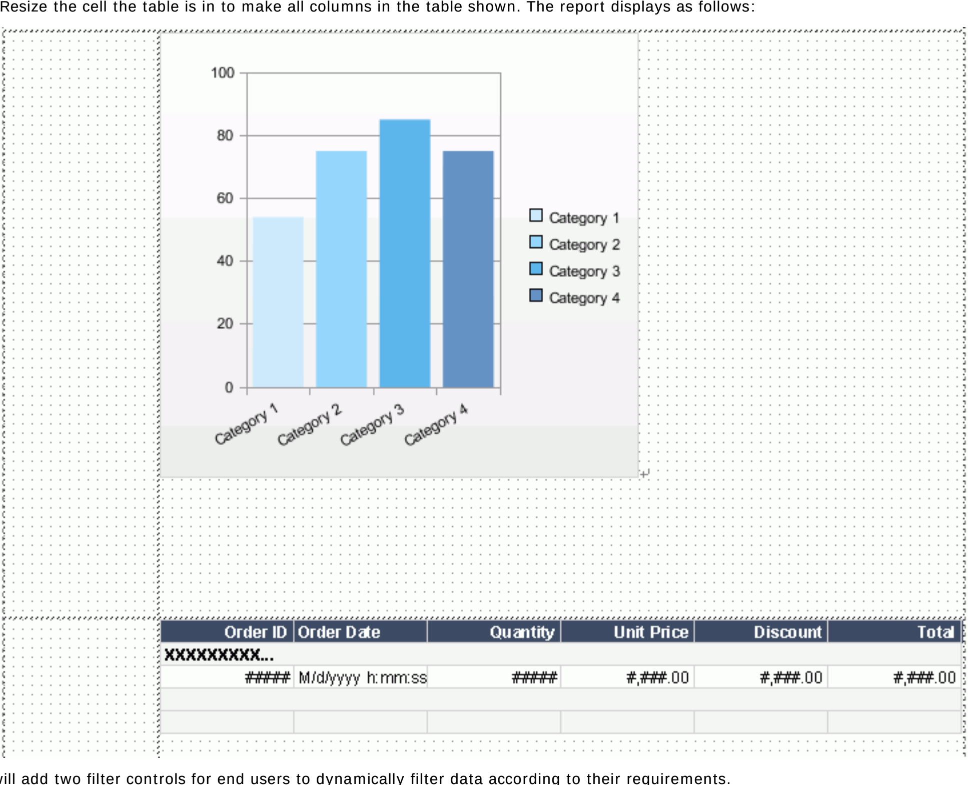
4. Drag **Chart** from the Toolbox panel to the right cell in the first tabular row. The Create Chart wizard appears.
5. In the Data screen, select **WorldWideSalesBV** in Data Source 1. Click **Next**.
6. In the Type screen, check the **Single chart** radio button, select **Bar** as the chart type and **Clustered Bar 2-D** as the sub type, then click **Next**.
7. In the Display Screen, add **Product Name** to the Category box and **Total Sales** to the Show Values box.
8. Switch to the **Style** Screen, select **ClassicBlue** as the chart style, then click **Finish** to create the chart.

Next, we will create a table to display the sales information just below the chart.

9. Drag **Table** from the Toolbox panel to the right cell in the second tabular row. The Table Type dialog appears.

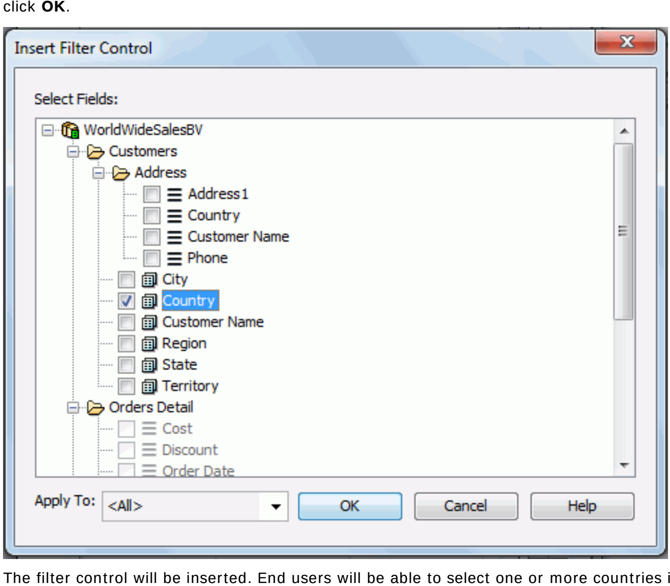


10. Keep the default selection in the Table Type dialog and click **OK**. The Create Table wizard is displayed.
11. In the Data screen of the wizard, select **WorldWideSalesBV** in Data Source 1, then click **Next**.
12. In the Display Screen, add the following fields to be displayed in the table: **Order ID**, **Order Date**, **Quantity**, **Unit Price**, **Discount**, and **Total**. Click **Next**.
13. In the Group screen, add **Product Name** as the group by field.
14. Switch to the **Style** Screen, then select **Neutral** as the table style.
15. Click **Finish** to create the table.
16. Resize the cell the table is in to make all columns in the table shown. The report displays as follows:



We will add two filter controls for end users to dynamically filter data according to their requirements.

17. Drag **Filter Control** from the Toolbox panel to the first tabular cell, then in the Insert Filter Control dialog, select **Country** and click **OK**.



   The filter control will be inserted. End users will be able to select one or more countries in it, the selected values will be applied to the previously created table and chart to show data based on the values.

18. Insert another filter control in the below tabular cell, based on the field Product Type using the same way.

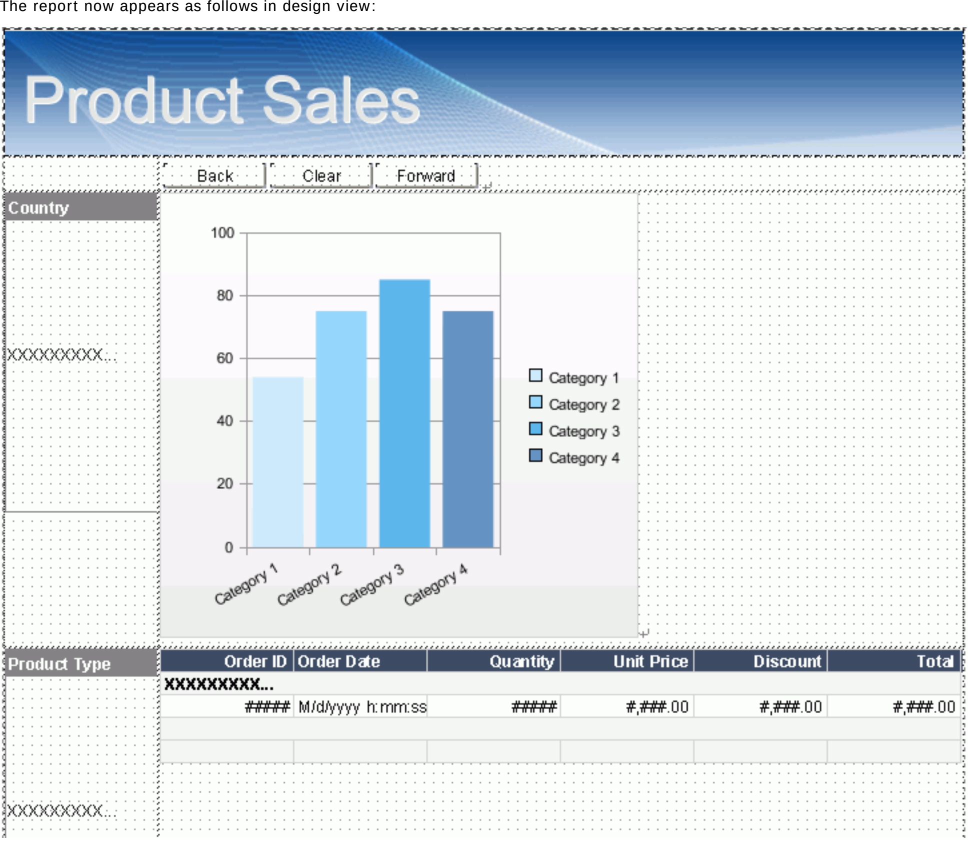
We will add a navigation control which helps end users deal with the filter applying status in all the filter controls in the report: go back to the previous filter applying status, go forward to the next, or clear all the filter applying histories. We will create the navigation control above the chart in the report.

19. Select the cell the chart is in, right-click and select **Insert Row Above** from the shortcut menu.
20. Select the right cell in the newly added tabular row, then in the Report Inspector, set its Height property to **0.28**.
21. Drag **Navigation Control** from the Toolbox panel to the right cell in the newly added tabular row, that is the cell above the chart.

Lastly, we will insert an image to show the title of the report in the report page header.

22. Select the tabular cell in the report page header, in the Report Inspector, set its Height property to **1**.
23. Drag **Image** from the Toolbox panel to the page header, in the Select an Image dialog, select **ProductSalesHeader.gif**, then click **Open**. Then select PageHeaderPanel in the Report Inspector to move the focus out of the image so as to cancel the scrollbars around the image.

   The report now appears as follows in design view:



Task 2: Format the web report

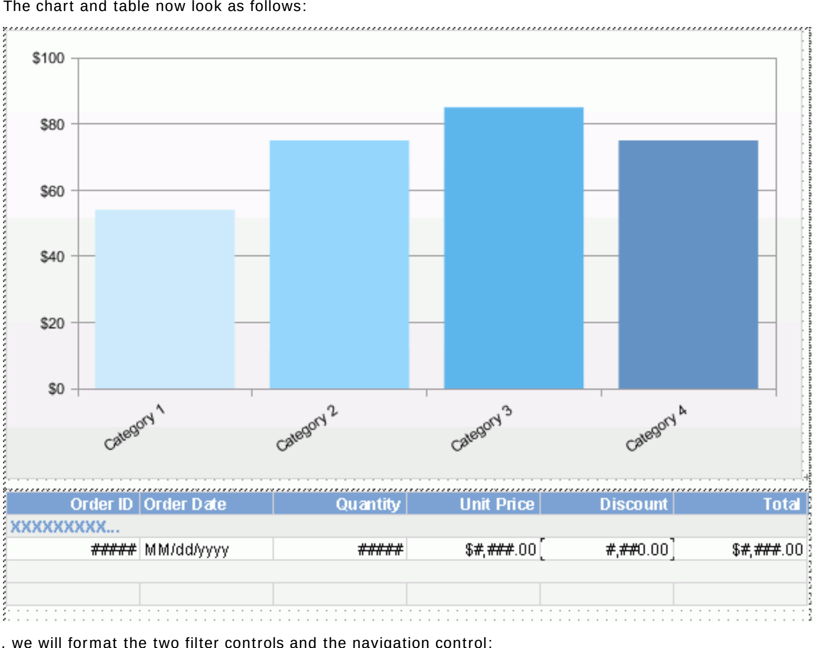
To improve the report appearance, we need to do some adjustments to the components in the report. First, we will format the chart.

1. Select the chart, or the **ChartObject** node in the Report Inspector, then resize it to make it fully occupy the cell that holds it.
2. Keep the chart selected, in the Report Inspector, set the Show Legend property in the Platform category to **false**.
3. Right-click on the chart, select **Format Axes** > **Format Category(X) Axis** from the shortcut menu.
4. In the Format Category(X) Axis dialog, switch to the **Orientation** tab, check the **Angle** radio button, set the value to **-35 deg**, then click **OK**. By doing this, we can make the tick mark labels on the category axis displayed completely.
5. Right-click on the chart again and click **Format Axes** > **Format Value(Y) Axis** on the shortcut menu.
6. In the Format Value(Y) Axis dialog, go to the **Format** tab, select **Number** from the Category box, select **$#,##0** from the Format box, then click **Add** to add the format in the Stack box. Click **OK** to exit the dialog.

Next, we will format the table:

7. Select the table header (TH) row, then in the Report Inspector, set its Background property to **0x7da1d1**.
8. Select the group header (GH) row and set its Background property to **0xeeeeee**.
9. Select the group by field in the group header, resize it horizontally to make sure the product names won't get truncated, then in the Report Inspector, set its Foreground property to **0x7da1d1**.
10. Select the **Order Date** DBField in the detail row, set its Format property in the Text Format category to **MM/dd/yyyy**.
11. Select the **Unit Price** and **Total** DBFields in the detail row and set their Format property to **$#,###.00**. Modify Discount's Format property to **#,##0.00**.

   The chart and table now look as follows:



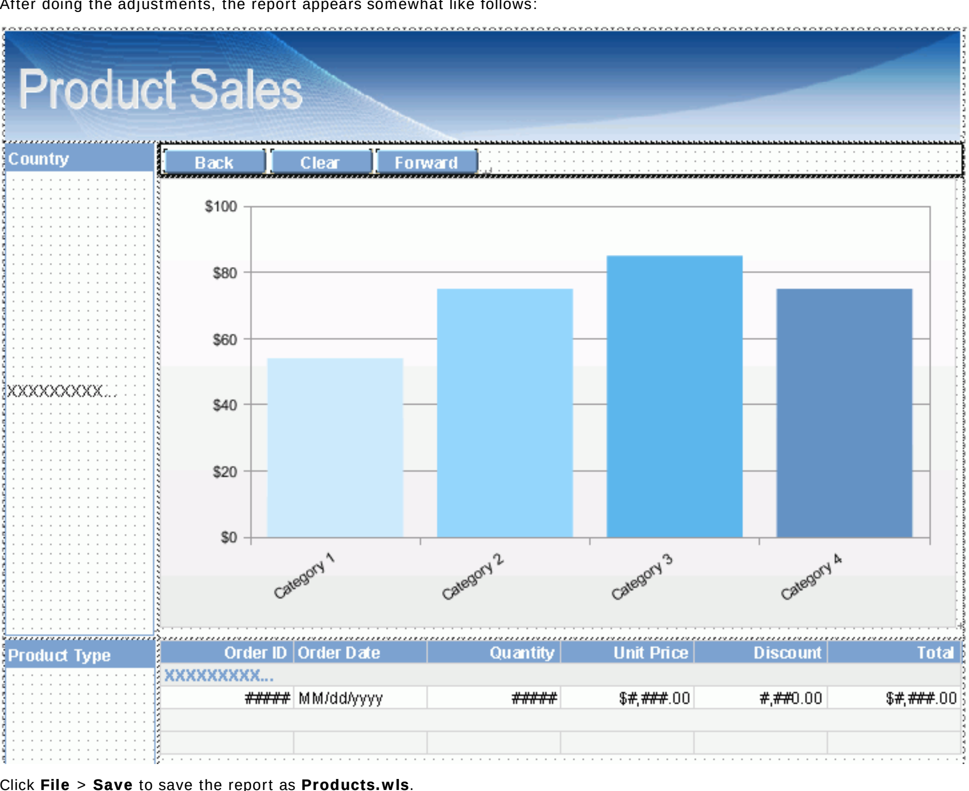
Next, we will format the two filter controls and the navigation control:

12. Select the **Country** filter control, resize it to make it the same size as the tabular cell that holds it. Resize the other filter control too.
13. Select the nodes that represent the two filter controls in the Report Inspector, then set their Border Color property in the Border category to **0x7ca2cf**, and Background in the Title category to **0x7ca2cf**.
14. Select the three buttons in the navigation control, then in the Report Inspector, set their Bold property to **true**, Background to **0x7ca2cf**, and Foreground to **White**.

The cell above the Country filter control is empty, to improve the report layout, we will make the cell merged with the cell that holds the Country filter control.

15. Select the empty cell and the cell the Country filter control is in by pressing the Ctrl button, right-click and select **Merge** from the shortcut menu. If you find it difficult to select the cell that holds the Country filter control, you can first select the filter control, then go to the Report Inspector, select the parent node of the filter control which represents the tabular cell.
16. Resize the Country filter control once again to make it fully occupies the merged cell.

   After doing the adjustments, the report appears somewhat like follows:



17. Click **File** > **Save** to save the report as **Products.wls**.

Task 3: Preview the web report

In this task we will preview the web report and perform some operations on it in JReport Designer. End users can also do the same thing on the web report after it is published to JReport Server.

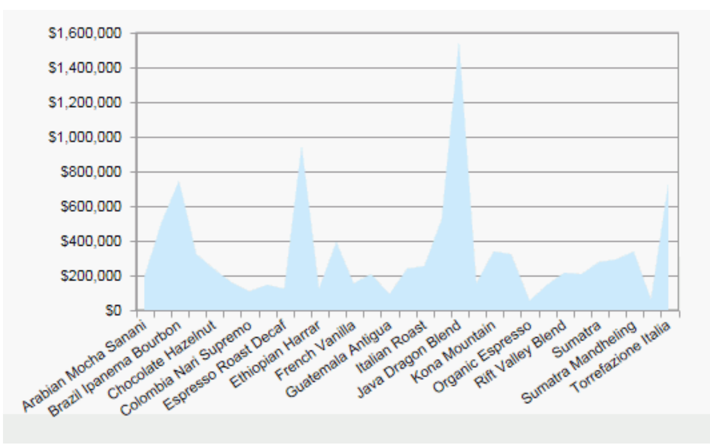
1. Click **View** > **Preview As** > **Web Report Result** on the menu bar. The report is opened in Web Report Studio.

Order ID	Order Date	Quantity	Unit Price	Discount	Total
Arabian Mocha Sanani					
3076	04/09/2014	9	$16.45	10.00	$133.24
3111	06/09/2014	62	$16.45	10.00	$917.91
3146	08/09/2014	62	$16.45	10.00	$917.91
3181	10/09/2014	62	$16.45	10.00	$917.91
3216	12/09/2014	62	$16.45	10.00	$917.91
3251	02/09/2015	62	$16.45	10.00	$917.91
3356	08/09/2015	841	$16.45	10.00	$12,451.00
3286	04/09/2015	841	$16.45	10.00	$12,451.00
3321	06/09/2015	841	$16.45	10.00	$12,451.00
3391	10/09/2015	62	$16.45	10.00	$917.91
3426	12/09/2015	62	$16.45	10.00	$917.91
3069	04/05/2014	7	$16.45	10.00	$103.63
3104	06/05/2014	233	$16.45	10.00	$3,449.56
3139	08/05/2014	233	$16.45	10.00	$3,449.56
3174	10/05/2014	233	$16.45	10.00	$3,449.56
3209	12/05/2014	46	$16.45	10.00	$681.03

First, we will change the chart type to area.

2. Click **Edit Mode** on the toolbar to switch Web Report Studio to the edit mode.
3. Select the chart platform, you will find the button available on the toolbar. Click the button, go to select **Area** > **Area 2-D** from the drop-down menu.

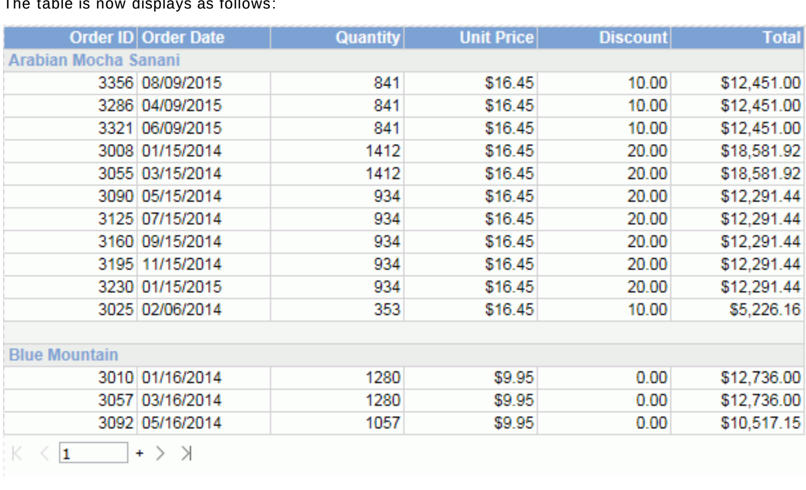
   The chart now looks as follows:



The table contains a lot of data. However, we would like to view data about total orders larger than $5000. So next, we will apply a filter to the table to narrow down data displayed on it.

4. Click the **Filter** button on the toolbar.
5. In the Filter dialog, select **TableComp** from the Apply to drop-down list, compose the condition as **Total >= 5000**, then click **OK**.

   The table is now displays as follows:

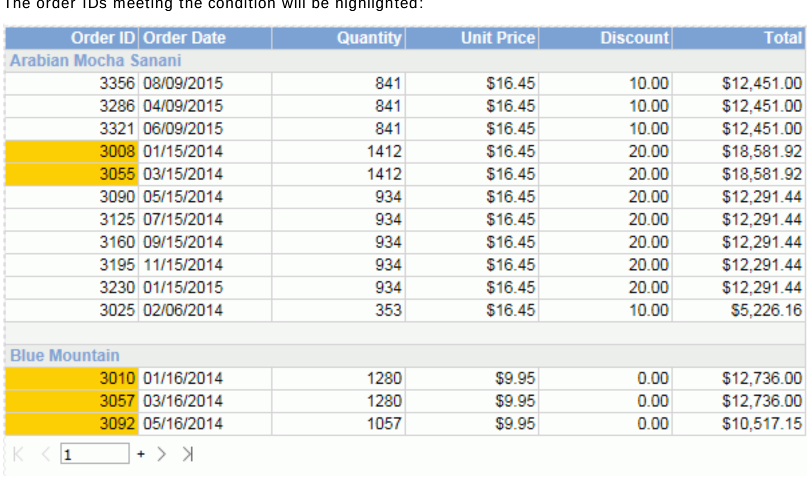


Order ID	Order Date	Quantity	Unit Price	Discount	Total
Arabian Mocha Sanani					
3356	08/09/2015	841	$16.45	10.00	$12,451.00
3286	04/09/2015	841	$16.45	10.00	$12,451.00
3321	06/09/2015	841	$16.45	10.00	$12,451.00
3008	01/15/2014	1412	$16.45	20.00	$18,581.92
3055	03/15/2014	1412	$16.45	20.00	$18,581.92
3090	05/15/2014	934	$16.45	20.00	$12,291.44
3125	07/15/2014	934	$16.45	20.00	$12,291.44
3160	09/15/2014	934	$16.45	20.00	$12,291.44
3195	11/15/2014	934	$16.45	20.00	$12,291.44
3230	01/15/2015	934	$16.45	20.00	$12,291.44
3025	02/06/2014	353	$16.45	10.00	$5,226.16
Blue Mountain					
3010	01/16/2014	1280	$9.95	0.00	$12,736.00
3057	03/16/2014	1280	$9.95	0.00	$12,736.00
3092	05/16/2014	1057	$9.95	0.00	$10,517.15

Next, we will apply a conditional formatting to the table to highlight the order IDs with the order quantity more than 1000.

6. Right-click in the **Order ID** column, and select **Conditional Formatting** from the shortcut menu.
7. In the Conditional Formatting dialog, click + to open the Edit Conditions dialog, set the filter condition to **Quantity >= 1000**, then click **OK** to return to the Conditional Formatting dialog. In the Format box, set Background Color to **#ffcc00**, then click **OK**.

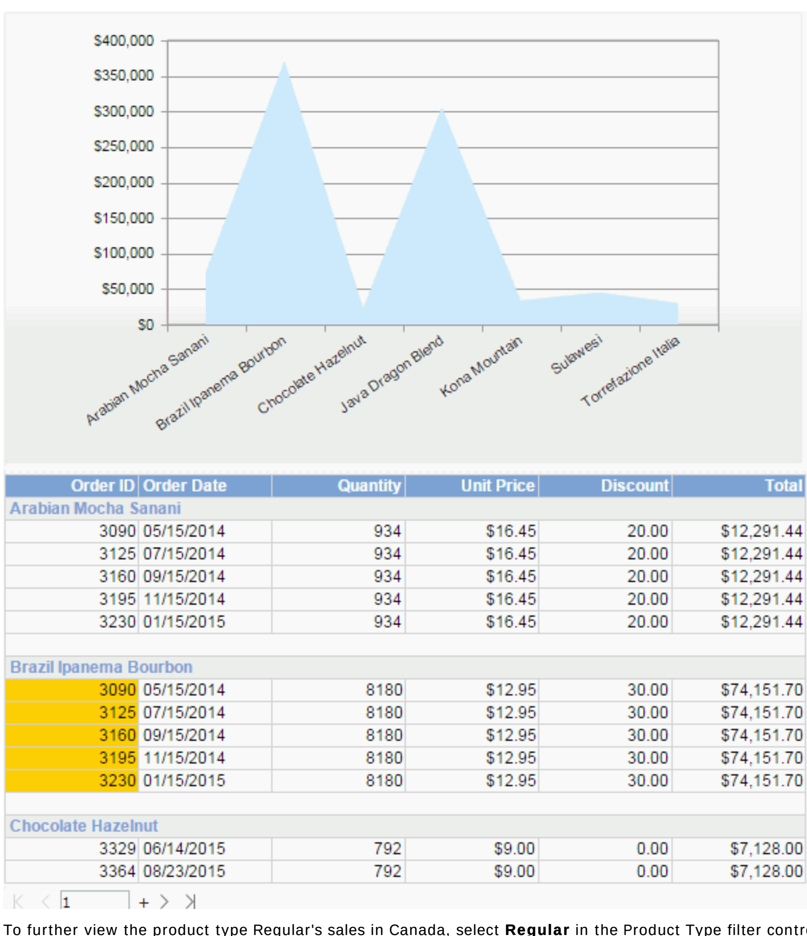
   The order IDs meeting the condition will be highlighted:



Order ID	Order Date	Quantity	Unit Price	Discount	Total
Arabian Mocha Sanani					
3356	08/09/2015	841	$16.45	10.00	$12,451.00
3286	04/09/2015	841	$16.45	10.00	$12,451.00
3321	06/09/2015	841	$16.45	10.00	$12,451.00
3008	01/15/2014	1412	$16.45	20.00	$18,581.92
3055	03/15/2014	1412	$16.45	20.00	$18,581.92
3090	05/15/2014	934	$16.45	20.00	$12,291.44
3125	07/15/2014	934	$16.45	20.00	$12,291.44
3160	09/15/2014	934	$16.45	20.00	$12,291.44
3195	11/15/2014	934	$16.45	20.00	$12,291.44
3230	01/15/2015	934	$16.45	20.00	$12,291.44
3025	02/06/2014	353	$16.45	10.00	$5,226.16
Blue Mountain					
3010	01/16/2014	1280	$9.95	0.00	$12,736.00
3057	03/16/2014	1280	$9.95	0.00	$12,736.00
3092	05/16/2014	1057	$9.95	0.00	$10,517.15

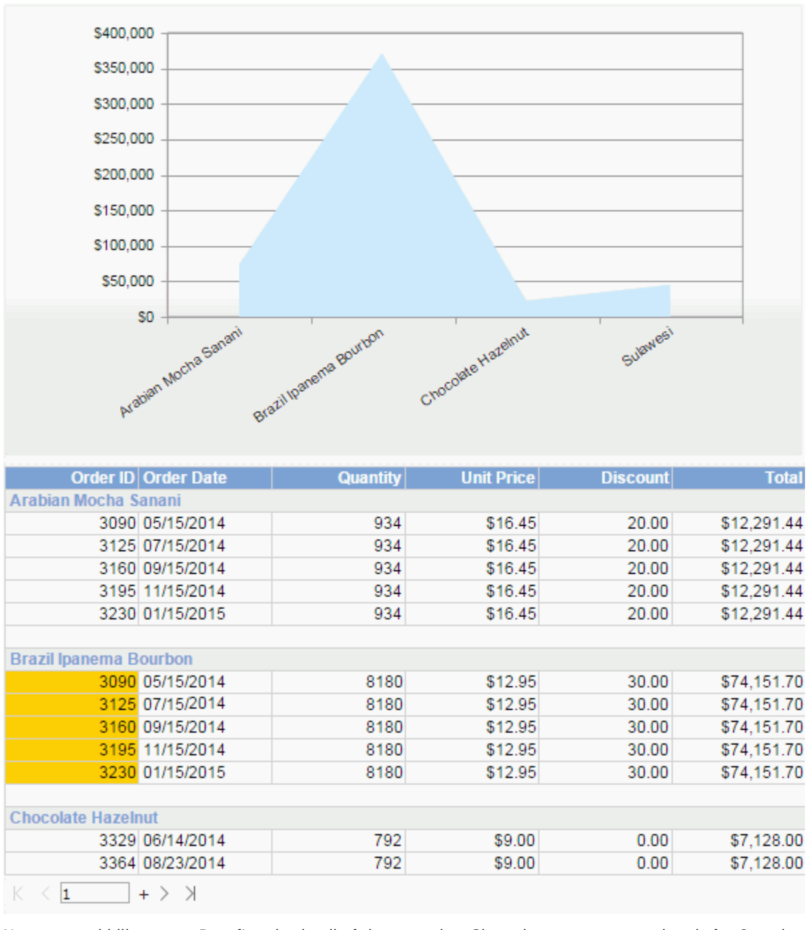
Now, we want to use the filter controls to dynamically change data displayed in the report. Since the chart and table are based on the same business view, the filters created via the filter controls will be applied to them both at the same time. First, we want to view the sales in Canada.

8. Click **Canada** in the Country filter control, then the chart and table is refreshed to display the data for Canada only.



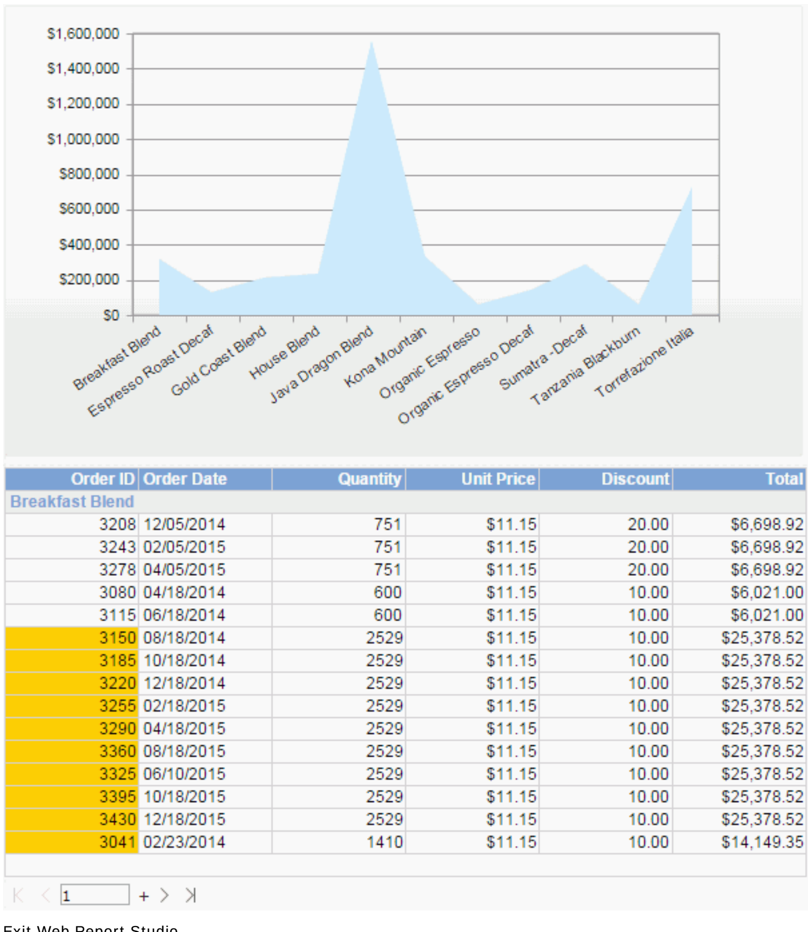
Order ID	Order Date	Quantity	Unit Price	Discount	Total
Arabian Mocha Sanani					
3090	05/15/2014	934	$16.45	20.00	$12,291.44
3125	07/15/2014	934	$16.45	20.00	$12,291.44
3160	09/15/2014	934	$16.45	20.00	$12,291.44
3195	11/15/2014	934	$16.45	20.00	$12,291.44
3230	01/15/2015	934	$16.45	20.00	$12,291.44
Brazil Ipanema Bourbon					
3090	05/15/2014	8180	$12.95	30.00	$74,151.70
3125	07/15/2014	8180	$12.95	30.00	$74,151.70
3160	09/15/2014	8180	$12.95	30.00	$74,151.70
3195	11/15/2014	8180	$12.95	30.00	$74,151.70
3230	01/15/2015	8180	$12.95	30.00	$74,151.70
Chocolate Hazelnut					
3329	06/14/2015	792	$9.00	0.00	$7,128.00
3364	08/23/2015	792	$9.00	0.00	$7,128.00

9. To further view the product type Regular's sales in Canada, select **Regular** in the Product Type filter control. The report will be refreshed to show the corresponding data:



Order ID	Order Date	Quantity	Unit Price	Discount	Total
Arabian Mocha Sanani					
3090	05/15/2014	934	$16.45	20.00	$12,291.44
3125	07/15/2014	934	$16.45	20.00	$12,291.44
3160	09/15/2014	934	$16.45	20.00	$12,291.44
3195	11/15/2014	934	$16.45	20.00	$12,291.44
3230	01/15/2015	934	$16.45	20.00	$12,291.44
Brazil Ipanema Bourbon					
3090	05/15/2014	8180	$12.95	30.00	$74,151.70
3125	07/15/2014	8180	$12.95	30.00	$74,151.70
3160	09/15/2014	8180	$12.95	30.00	$74,151.70
3195	11/15/2014	8180	$12.95	30.00	$74,151.70
3230	01/15/2015	8180	$12.95	30.00	$74,151.70
Chocolate Hazelnut					
3329	06/14/2014	792	$9.00	0.00	$7,128.00
3364	08/23/2014	792	$9.00	0.00	$7,128.00

10. Now we would like to see Decaf's sales in all of the countries. Since the current report data is for Canada, we need to remove the Canada filter. There are two methods to do this: in the navigation control above the chart, click **Back** twice to undo what we have just done in steps 8 and 9, or click **Clear** to simply remove the filter conditions added via the filter controls in steps 8 and 9. After back to the full data, click **Decaf** in the Product Type filter control, and you will see the desired data.



Order ID	Order Date	Quantity	Unit Price	Discount	Total
Breakfast Blend					
3208	12/05/2014	751	$11.15	20.00	$6,698.92
3243	02/05/2015	751	$11.15	20.00	$6,698.92
3278	04/05/2015	751	$11.15	20.00	$6,698.92
3080	04/18/2014	600	$11.15	10.00	$6,021.00
3115	06/18/2014	600	$11.15	10.00	$6,021.00
3150	08/18/2014	2529	$11.15	10.00	$25,378.52
3185	10/18/2014	2529	$11.15	10.00	$25,378.52
3220	12/18/2014	2529	$11.15	10.00	$25,378.52
3255	02/18/2015	2529	$11.15	10.00	$25,378.52
3290	04/18/2015	2529	$11.15	10.00	$25,378.52
3360	08/18/2015	2529	$11.15	10.00	$25,378.52
3325	06/10/2015	2529	$11.15	10.00	$25,378.52
3395	10/18/2015	2529	$11.15	10.00	$25,378.52
3430	12/18/2015	2529	$11.15	10.00	$25,378.52
3041	02/23/2014	1410	$11.15	10.00	$14,149.35

11. Exit Web Report Studio.

Lesson 2 summary

In this lesson, we created a web report from the view of a report developer in JReport Designer and previewed it. Filter controls and navigation control are applied so that end users can view data within various specific ranges.

⇦⇨

Track 3: Creating 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 published along with their catalog to the component library in JReport Server for use in dashboards.

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. Business views are created and managed in JReport Designer. For how to create a business view, see Creating a business view.

In this track, we will create two library components in JReport Designer as the role of a report developer, based on the business view WorldWideSalesBV created in the previous track. One library component contains a table that shows each product's order details, including order ID, quantity, unit price, and the total sales for each order. Another library component contains a chart that demonstrates the total sales of each product. In order for end users to see sales information by order ID in the table, we will define to deliver a built-in message 0002-Sort between the two library components. We will also add a text field in the configuration panel of the table library component and use it to change the background color of the Quantity field in the table. In addition, we will add a slider to dynamically filter the data.

Note: A JDashboard license for JReport Designer is required in order to perform this track. If you do not have the license, please contact your Jinfonet Software account manager to obtain it first.

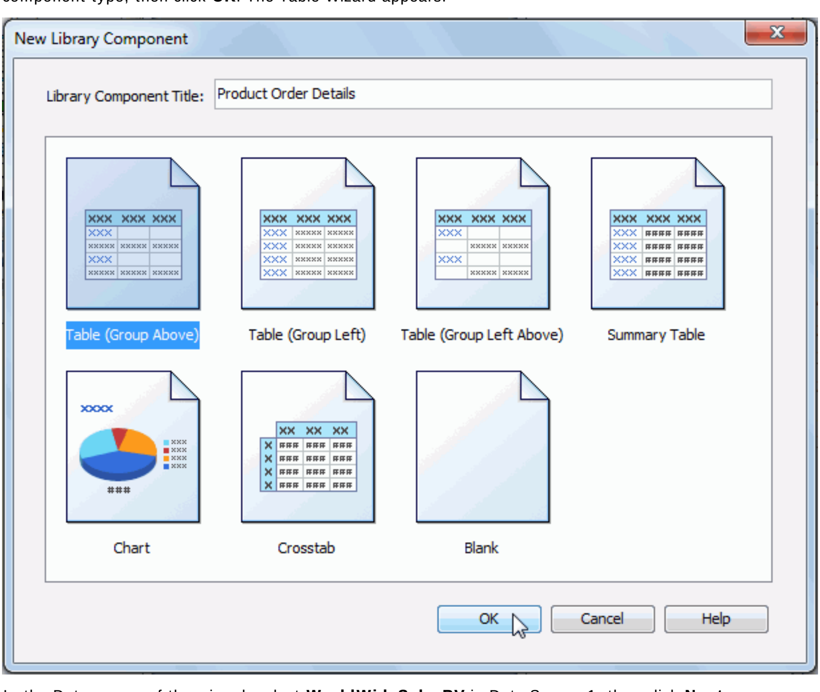
This track contains the following tasks:

- Task 1: Create two library components
- Task 2: Deliver a message between the library components
- Task 3: Insert a slider to filter data
- Task 4: Use text field to change property value

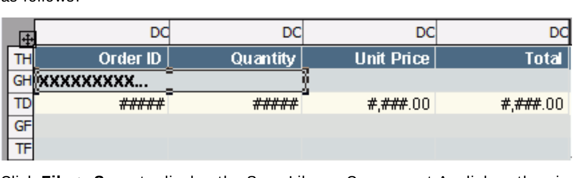
Task 1: Create two library components

We will create two library components, including a table and a chart, then we can filter, sort, or change properties of them.

1. Click **Start** > **All Programs** > **JReport 13.5 Update 2** > **Report Designer** to start JReport Designer. The JReport Designer window and the Welcome to JReport Designer dialog appear.
2. In the Welcome to JReport Designer dialog, click the **Catalog** link in the Open category.
3. Browse to select the **JinfonetGourmetJava.cat** catalog file in `<install_root>\Demo\Reports\JinfonetGourmetJava`, then click the **Open** button.
4. On the menu bar, click **File** > **New** > **Library Component** to display the New Library Component dialog.
5. Input **Product Order Details** in the Library Component Title text field, select the **Table (Group Above)** component type, then click **OK**. The Table Wizard appears.



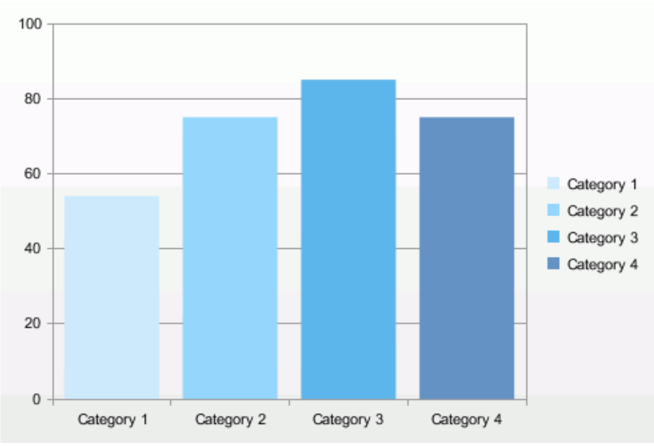
6. In the Data screen of the wizard, select **WorldWideSalesBV** in Data Source 1, then click **Next**.
7. In the Display Screen, add the following fields to be displayed in the table: **Order ID**, **Quantity**, **Unit Price**, and **Total**. Click **Next**.
8. In the Group screen, add **Product Name** as the group by field.
9. Switch to the **Style** screen and select **Commercial** from the style list, then click **Finish** to create the table.
10. Resize the group by field to make sure all the product names can be shown completely. The table displays as follows:



11. Click **File** > **Save** to display the Save Library Component As dialog, then input **ProductOrderDetails.lc** into the Report Name text field and click **Save**.

Next, we will create the second library component that contains the chart.

12. On the menu bar, click **File** > **New** > **Library Component** to display the New Library Component dialog.
13. Input **Product Sales** in the Library Component Title text field, select **Chart** from the component type box and click **OK**. The Chart Wizard is displayed.
14. In the Data screen of the wizard, select **WorldWideSalesBV** in Data Source 1, then click **Next**.
15. Keep the default selections in the Type screen, then click **Next**.
16. In the Display Screen, add **Product Name** to the Category box and **Total Sales** to the Show Values box.
17. Switch to the **Style** Screen, select **ClassicBlue** as the chart style.
18. Click **Finish** to create the chart. The chart displays as follows.

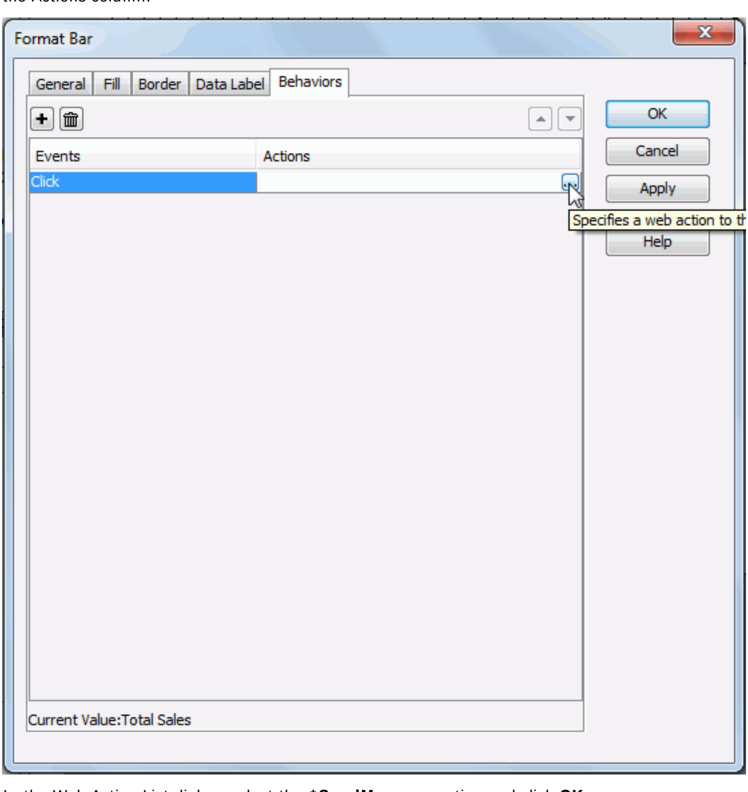


19. Click **File** > **Save**. In the Save Library Component As dialog, input **ProductSales.lc** into the Report Name text field and click **Save**.

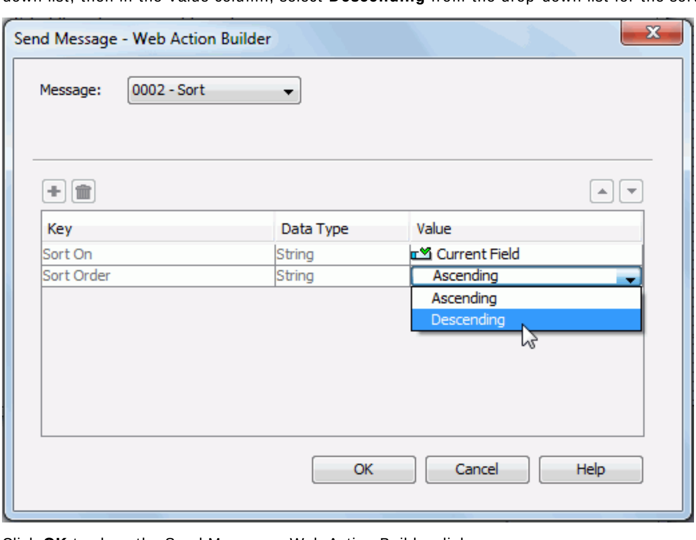
Task 2: Deliver a message between the library components

In this task, we will define a sort message to be delivered between the two library components. We will use the chart to send the message, and the table to receive the message.

1. Right-click a bar in the chart and select **Format 2D Bar** from the shortcut menu to display the Format Bar dialog.
2. Switch to the **Behaviors** tab, select **Click** from the drop-down list in the Events column, then click [...] in the Actions column.



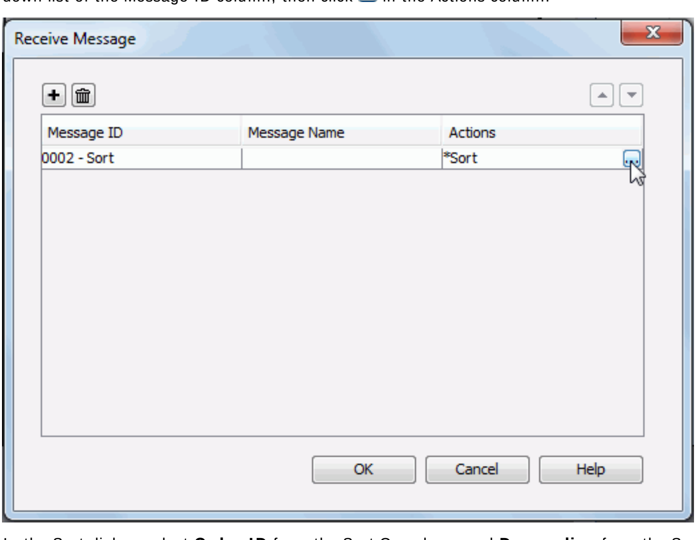
3. In the Web Action List dialog, select the ***SendMessage** action and click **OK**.
4. In the Send Message - Web Action Builder dialog, select the built in message **0002 - Sort** from the drop-down list, then in the Value column, select **Descending** from the drop-down list for the sort order.



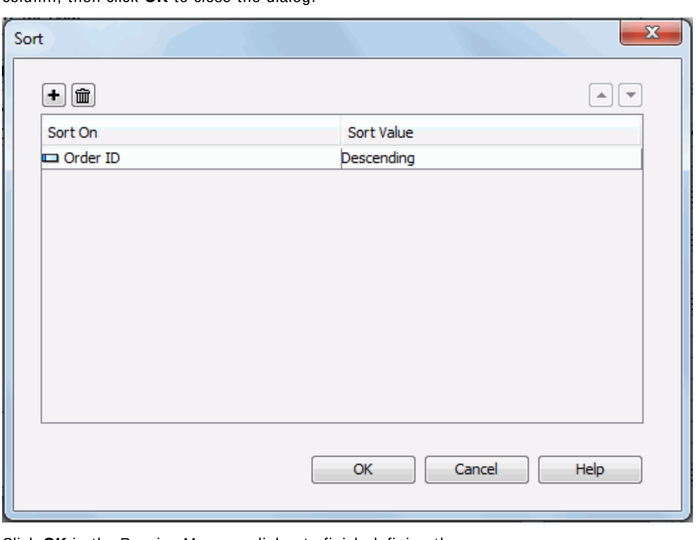
5. Click **OK** to close the Send Message - Web Action Builder dialog.
6. Click **OK** in the Format Bar dialog to finish defining the message.
7. Click **File** > **Save** to save the library component.

Next, we will define to make the table receive the sort message that will be sent out from the chart once the click action is triggered on the chart bars.

8. Click **Window** > **ProductOrderDetails.lc** on the menu bar to switch to the table library component.
9. Right-click the table and select **Receive Message** from the shortcut menu to display the Receive Message dialog.
10. In the dialog, click [+] to add a message line, select the built-in message **0002 - Sort** from the drop-down list of the Message ID column, then click [...] in the Actions column.



11. In the Sort dialog, select **Order ID** from the Sort On column and **Descending** from the Sort Value column, then click **OK** to close the dialog.



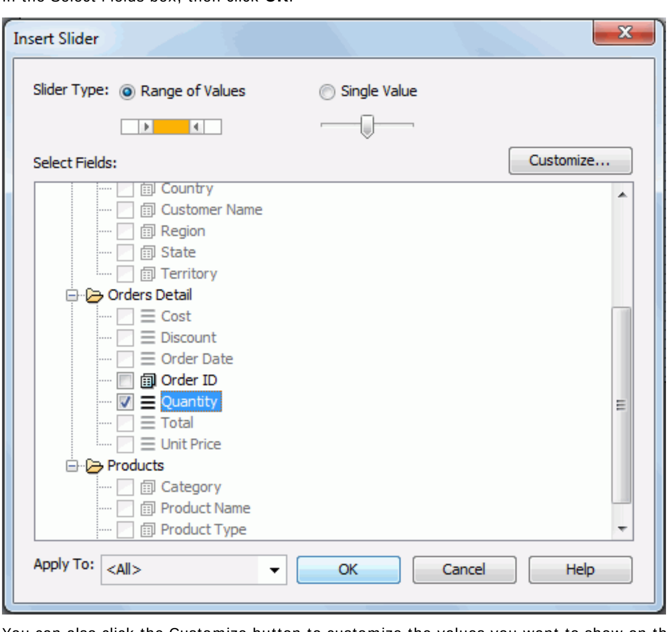
12. Click **OK** in the Receive Message dialog to finish defining the message.
13. Click **File** > **Save** to save the library component.

By now, the message between the chart and table is created and it is defined as follows: when any bar on the chart is clicked at runtime, the built-in message 0002 - Sort will be sent out from the chart to the table, and when the table receives the message, it will make the order IDs sorted descendingly as a response to the message.

Task 3: Insert a slider to filter data

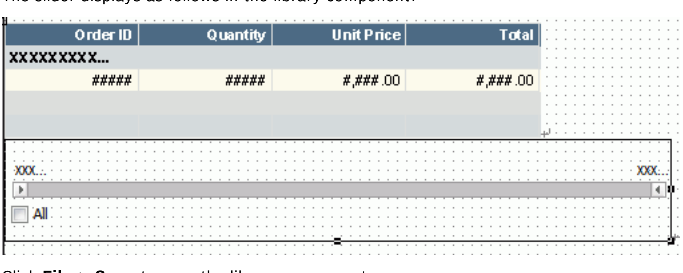
To see which products sell well, we can insert a slider to filter on the Quantity field for the table.

1. Drag **Slider** from the Toolbox panel to the blank area in the table library component.
2. In the Insert Slider dialog, select **Range of Values** as the slider type and **Quantity** from the resource list in the Select Fields box, then click **OK**.



You can also click the Customize button to customize the values you want to show on the slider. Here, we will use all the values in the Quantity field on the slider.

The slider displays as follows in the library component:

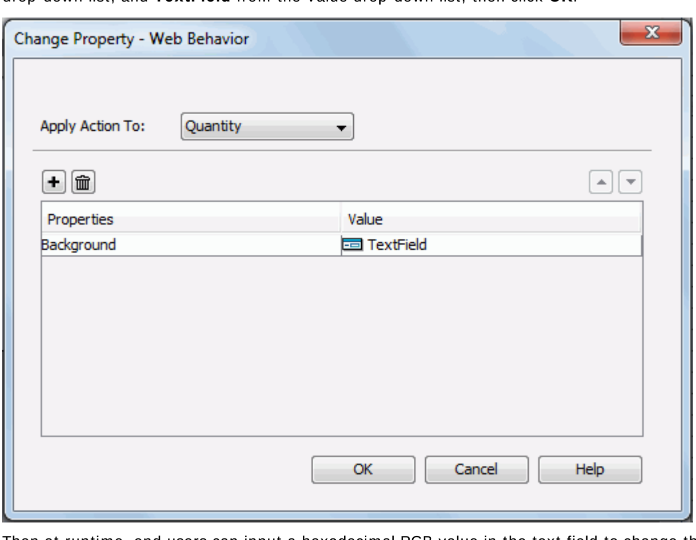


3. Click **File** > **Save** to save the library component.

Task 4: Use text field to change property value

In this task, we will insert a text field into the configuration panel of the table library component and use it to change the background color of the Quantity field in the table.

1. Check the **Display Configuration Panel** checkbox on the top-right corner to display the panel.
2. Drag **Text Field** from the Toolbox panel and drop it in the configuration panel.
3. Right-click the text field and select **Web Behaviors** > **Change Property** from the shortcut menu to display the Change Property - Web Behavior dialog.
4. In the dialog, select **Quantity** from the Apply Action To drop-down list, **Background** from the Properties drop-down list, and **TextField** from the Value drop-down list, then click **OK**.

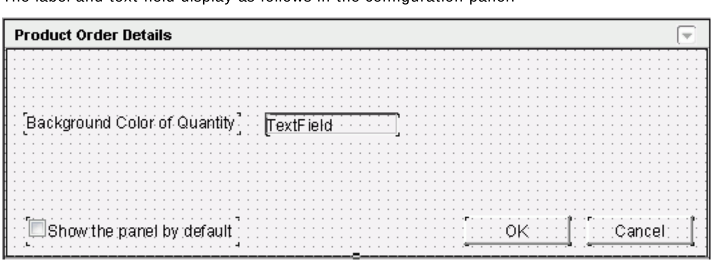


Then at runtime, end users can input a hexadecimal RGB value in the text field to change the background color of the Quantity field in JDashboard.

Next we will add a label ahead of the text field to help end users identify what it is used for.

5. Drag **Label** from the Toolbox panel and place it ahead of the text field.
6. Double-click the label and edit its text to **Background Color of Quantity**. Resize the label to make the whole text in it shown.

The label and text field display as follows in the configuration panel:



7. Click **File** > **Save** to save the library component.

Track 3 summary

In this track, we created two library components, defined a sort message to be delivered between the library components, added a slider to help end users filter data dynamically and used a web control to change object properties at runtime.

Track 4: Publishing, running and administering resources

In this track it is assumed that you are a report administrator. You need to receive the reports and library components which are designed in JReport Designer from the report developers and publish them to JReport Server, so the end users can access them and view them.

In this track you perform the following lessons:

- Lesson 1: Starting JReport Server
- Lesson 2: Publishing resources
- Lesson 3: Running reports
- Lesson 4: Scheduling reports
- Lesson 5: Administration of resources
- Lesson 6: Security

Click the right arrow (⇨) below to begin reading the track.

Lesson 1: Starting JReport Server

In this lesson, you will learn how to start JReport Server, open the JReport Console, run a public report, and learn the other report control commands available from the JReport Console.

- Task 1: Start JReport Server
- Task 2: Access the JReport Console page

Task 1: Start JReport Server

The lesson assumes that you have installed JReport Server with its default configuration. If you changed the default locations or port numbers using Custom Installation, then you will need to adjust the instructions in this lesson accordingly.

Start JReport Server in one of the following ways:

- Choose **Start** > **All Programs** > **JReport 13.5 Update 2** > **Start Server**.
- Double-click the **JReport Server 13.5 Update 2** shortcut on the Windows desktop.
- Run the JRServer.bat shell script in directory `<install_root>\bin`.

These methods run the JReport Server in its default non-logging mode. The JReport Server can also be run in logging mode. In this mode, JReport Server writes key actions to a log file. This is useful when you encounter problems and need to see the actual SQL calls issued to the DBMS, performance issues when reports are taking too long to run or failures when a report does not run at all. To run with full logging, start JReport Server using DJRServer.bat in `<install_root>\bin`.

After you issue the command to start JReport Server, a command window appears with various startup messages. When you see the message "JReport Server is ready for service.", you are ready to proceed to the next task. Do not close the command window.

Task 2: Access the JReport Console page

JReport Console page is an easy-to-use web-based interface of JReport Server for selecting, viewing and scheduling reports. To access the JReport Console page, follow these steps:

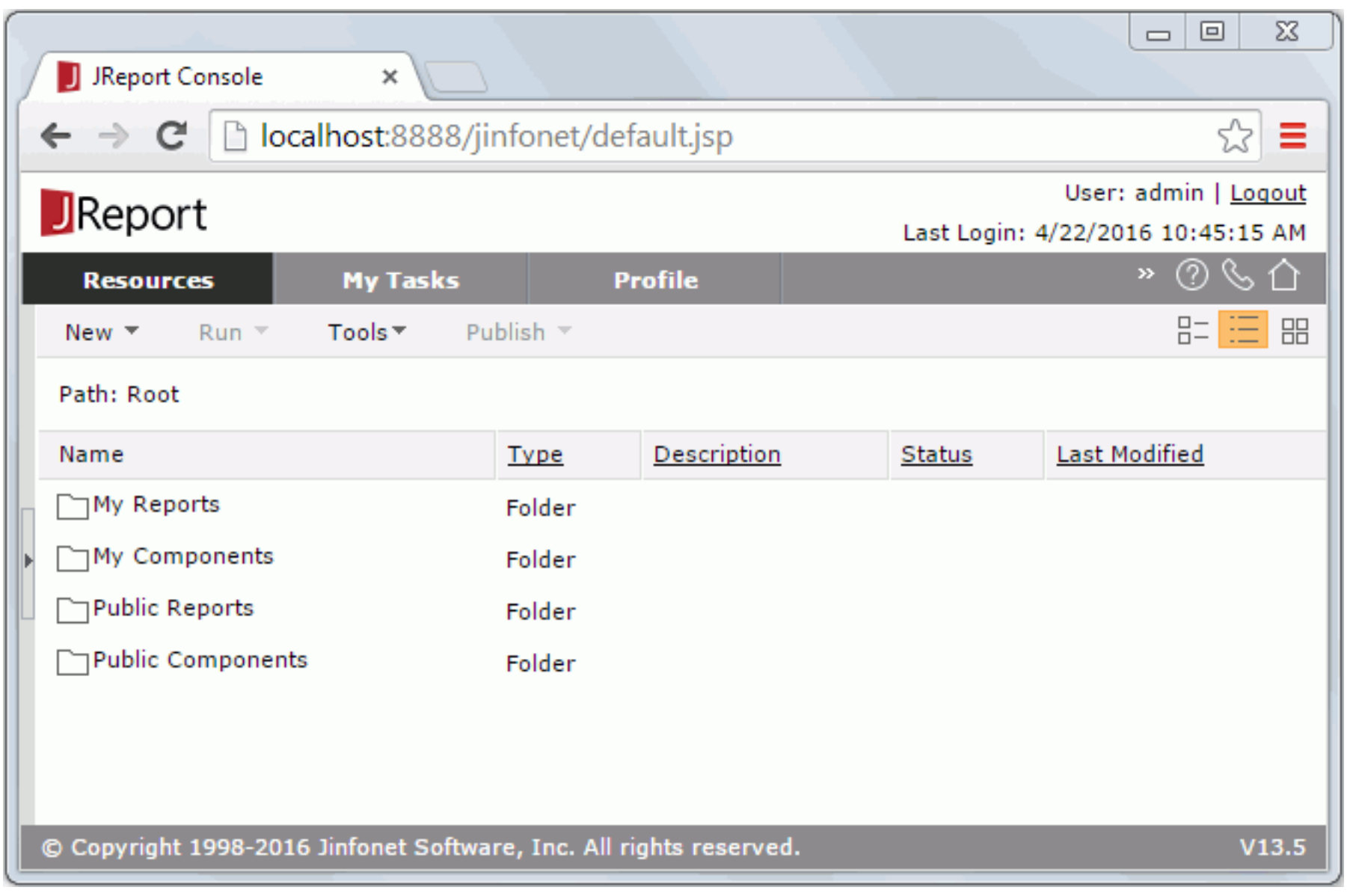
1. Open a web browser and set the URL to http://localhost:8888.

   Internet Explorer, Firefox, and Google Chrome are browsers that have been certified to work with JReport products. However other browsers should also work fine.

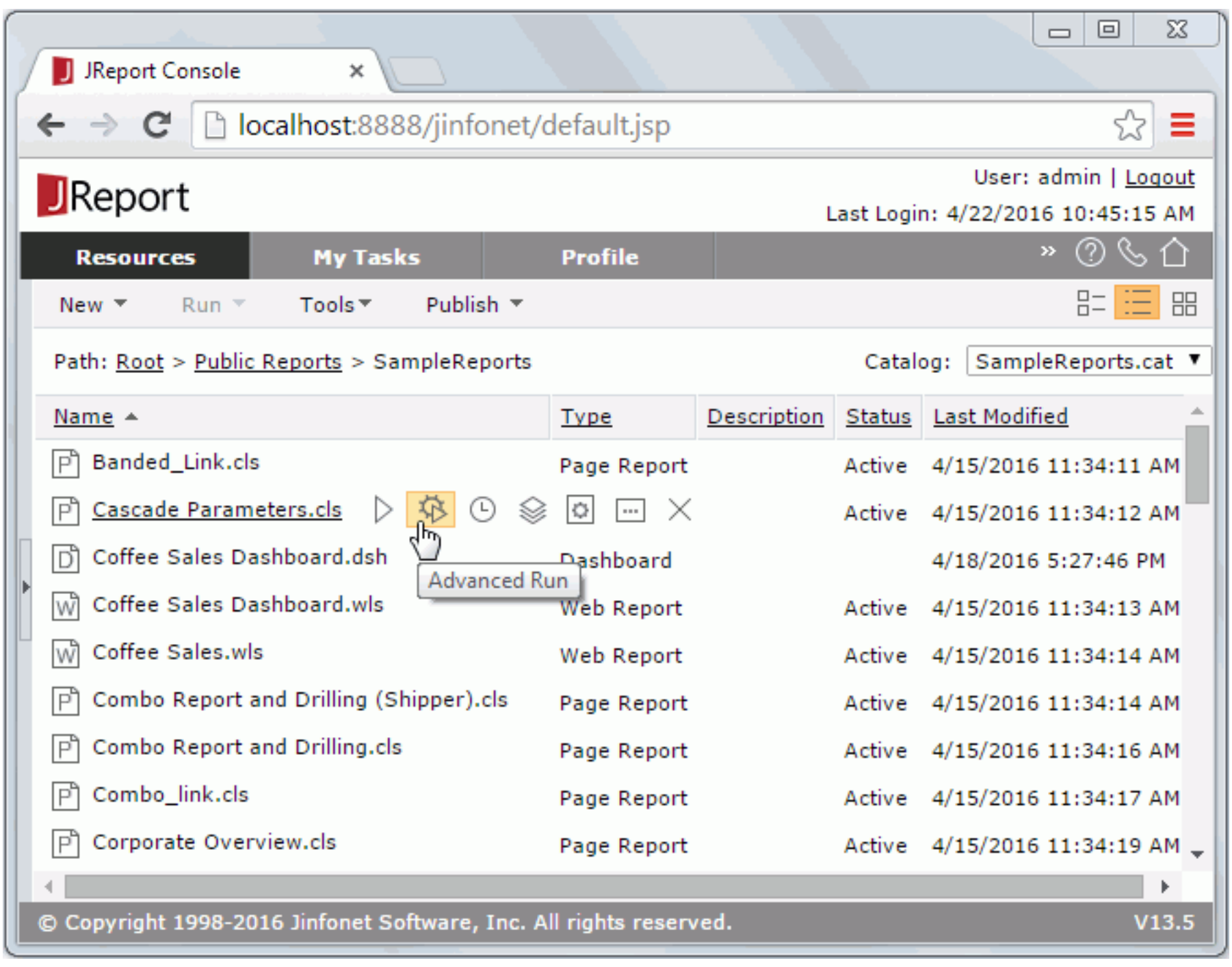
2. On the welcome page, enter **admin** for the user name and **admin** for the password, then click **Login**.

   These are the initial user credentials built in to JReport Server. One of the first things an administrator should do is to set up the correct user credentials so that appropriate reports can be accessed by each user.

   The JReport Console page appears.



3. By default, the root folder is open when you log onto the JReport Console page. The User Directory panel is hidden on the left showing the folders of published reports that are available to this user (admin in our case).
4. Click **Public Reports** and then **SampleReports**. The report list, along with pertinent information, is displayed.
5. Focus on any of the report row, a floating toolbar is then displayed, which contains the following commands in the order from left to right:



Run
Runs the report immediately using the default format.

Advanced Run
Runs the report immediately after collecting additional specifications from you, including which report tab to run for a page report, what output format to use, which style group to apply, and any encoding options.

Schedule
Runs the report based on a schedule. When you use Schedule to run a report, the request is placed into a queue and processed in order submitted. The size of the queue and priority of the queue can be configured to make the most efficient use of resources. Lesson 4 describes scheduled reports.

Version
Views report version and scheduled/advanced run result version information.

Properties
Does not run the report, but allows you to define the archive policy, security, and a description associated with the report. Lesson 6 describes security.

Parameter Settings
Customizes the default parameter values for the report.

Delete
Deletes the report.

Lesson 2: Publishing resources

JReport Server is only able to run reports and use library components that have been published to it. As you saw in the previous lesson, some reports are automatically published to JReport Server, in the Public Reports folder. In this lesson, we publish the reports and library components in the JinfonetGourmetJava.cat catalog.

A report or library component must initially be published along with its catalog. Updates to the resource can be published independently, as long as the catalog associated with the resource remains published. This allows you to quickly and easily install a change to a resource in the runtime environment.

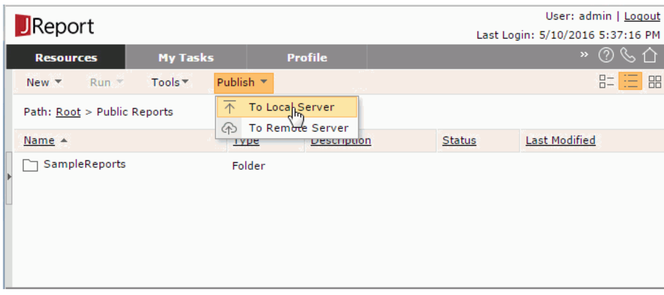
In this lesson, you can publish the catalog, reports and library components that you developed in previous tracks. If you haven't complete those tracks or do not want to use your own, you can use the version of the JinfonetGourmetJava catalog offered by JReport Server, which is located at `<install_root>\help\samples\JinfonetGourmetJava`.

This lesson contains the following tasks:

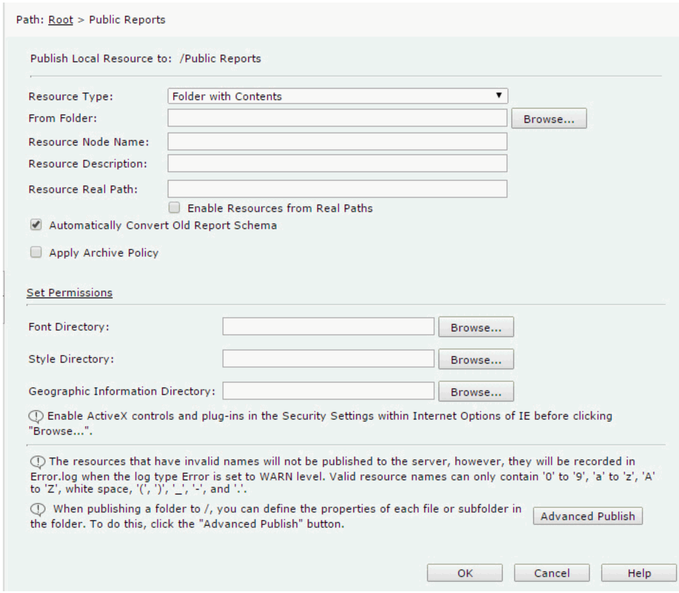
- Task 1: Publish reports
- Task 2: Publish library components

Task 1: Publish reports

1. On the JReport Console > Resources page, open the **Public Reports** folder.
2. Click **Publish** > **To Local Server** on the task bar of the Resources page.



The Publish to Local Server page is displayed:



3. From the Resource Type drop-down list, select **Folder with Contents** if it is not already selected.

   The Resource Type list allows you to specify the resource, the catalog, report, or folder with these objects, to be published to the My Reports or Public Reports folder of JReport Server.
4. Click the **Browse** button next to the From Folder field, and then browse to select the **`<install_root>\help\samples\JinfonetGourmetJava`** directory.
5. In the Resource Node Name field enter **JinfonetGourmetJava**.
6. In the Resource Description field enter **Jinfonet Gourmet Java catalog and reports**.
7. Click **OK** to publish the resources.

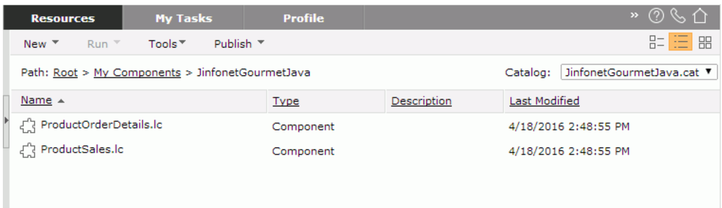
   After the resources have been successfully published, the JinfonetGourmetJava folder is displayed in the resource tree.
8. Click the folder name **JinfonetGourmetJava** and the reports in the folder are listed:

Task 2: Publish library components

1. On the JReport Console > Resources page, open the **My Components** folder.
2. Click **Publish** > **To Local Server** on the task bar of the Resources page. The Publish to Local Server dialog will be displayed.

   From the Resource Type drop-down list, select **Folder with Contents** if it is not already selected. This type allows for a folder with library components, a catalog, and sub folders containing these objects to be published.
3. Click the **Browse** button next to the From Folder field, and then browse to select the **`<install_root>\help\samples\JinfonetGourmetJava`** directory.
4. In the Resource Node Name field enter **JinfonetGourmetJava**.
5. In the Resource Description field enter **Jinfonet Gourmet Java catalog and library components**.
6. Click **OK** to publish the resources.

   After the resources have been successfully published, the JinfonetGourmetJava folder is displayed in the resource tree.
7. Click the folder name **JinfonetGourmetJava** and the library components in the folder will be listed.



Lesson 3: Running reports

This lesson describes how to run reports from JReport Console. This lesson represents one way that end users could access and run published reports. Reports that are embedded in a Java application would not be accessed from JReport Console, but instead from the Java application. The interaction with the report itself however would be the same as described here.

The reports you view in this lesson are in the JinfonetGourmetJava folder, which is the folder you published them to in Lesson 2 > Task 1 of this track.

- Task 1: Directly run a report
- Task 2: Run a parameter-based report
- Task 3: Run a report in Advanced mode

Task 1: Directly run a report

Direct running reports is to use the default format setting to view the report result. For page reports (for example, .cls) the default format is Page Report and they will run in Page Report Studio. For web reports (.wls) they will be opened in Web Report Studio by default.

If the reports are created on JReport Server, the default format is controlled by the Default Format for Viewing Report option in the Profile > Customize Server Preferences > General tab. When this option is set to Page Report or Applet, web reports will run in Web Report Studio by default.

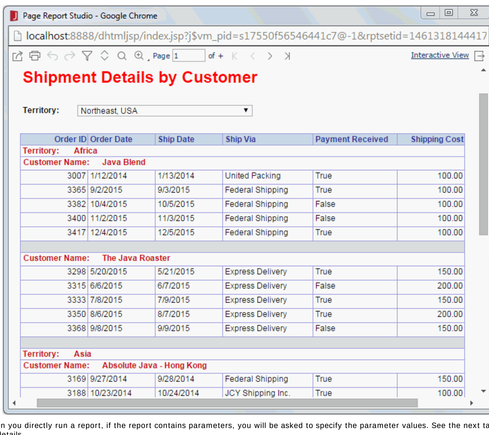
If the reports are created using JReport Designer, the default format is controlled by the Default Format for Viewing Report property in the Report Inspector. If this property is set to <Server Setting>, the viewing format will be determined by the above setting on server.

Since a page report can contain multiple report tabs and these report tabs may have different default format settings, for example, report tab 1 is Page Report, tab 2 HTML and tab 3 Excel. In this case, there will be different direct running results according to which report tab is the last-time focused tab in the page report when the page report is saved at report design time. If it is report tab 1, all the tabs in the page report will be opened in Page Report Studio. If it is report tab 2 or 3, only tab 2 or 3 will be displayed in the specified format, and this applies to all the other formats except Page Report.

In this task, we will run a page report in the JinfonetGourmetJava folder as an example:

1. On the JReport Console page, go to the **Public Reports** > **JinfonetGourmetJava** folder.
2. Click the **ShipmentDetailsbyCustomer.cls** link to run the report.

   A separate web browser window opens, with the report result for the ShipmentDetails report tab in the page report:



When you directly run a report, if the report contains parameters, you will be asked to specify the parameter values. See the next task for details.

Task 2: Run a parameter-based report

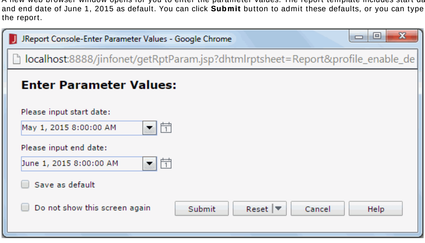
As you saw in the Creating a parameter-based report lesson, parameter-driven reports are another way to have a dynamic report. They allow the end users to specify the data that they want to see before the query is issued to the database. By only selecting the data that the report needs, the query is more efficient.

When a parameter-based report is run, JReport Server automatically generates an HTML page to prompt the user for the expected values. Default values, if included in the report template definition, are also displayed.

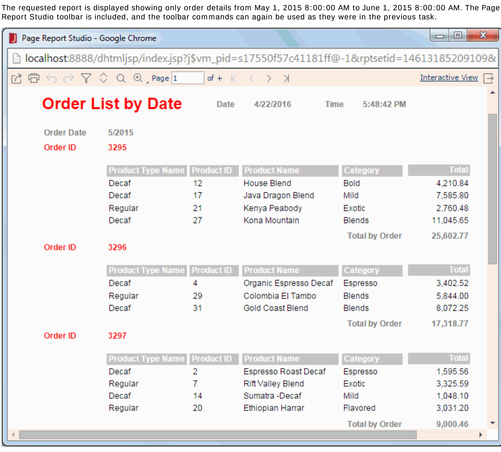
To run the parameter-based report:

1. In the JinfonetGourmetJava folder, click the **OrderListbyDate_Parameter.cls** link.

   A new web browser window opens for you to enter the parameter values. The report template includes start date of May 1, 2015 and end date of June 1, 2015 as default. You can click **Submit** button to admit these defaults, or you can type new values to view the report.



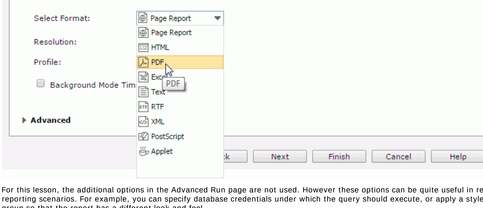
2. The requested report is displayed showing only order details from May 1, 2015 8:00:00 AM to June 1, 2015 8:00:00 AM. The Page Report Studio toolbar is included, and the toolbar commands can again be used as they were in the previous task.



Task 3: Run a report in Advanced mode

The Advanced Run command allows you to specify additional properties when running a report, for example you can specify a different report format instead of the default one. Options available include Applet, HTML, PDF, Text, Excel, Postscript, Rich Text Format (RTF), XML and Page Report. In this lesson we will run the report as PDF format.

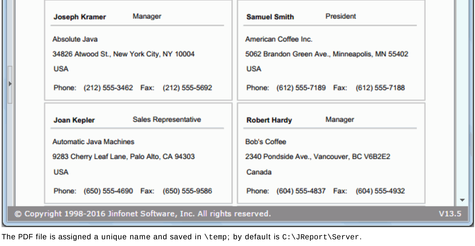
1. In the JinfonetGourmetJava folder, click in the **CustomerContactCard.cls** row, right-click in the row and then select **Advanced Run** from the shortcut menu.
2. Click the **Format** tab and select **PDF** from the Select Format drop-down list, as shown below:



   For this lesson, the additional options in the Advanced Run page are not used. However these options can be quite useful in real reporting scenarios. For example, you can specify database credentials under which the query should execute, or apply a style group so that the report has a different look and feel.

3. Click the **Finish** button.

   The report is displayed as a PDF file in your web browser. You can now browse the report, print it, e-mail it to a co-worker, or save it to your local disk using the Adobe Reader commands.



The PDF file is assigned a unique name and saved in `\temp`; by default is `C:\JReport\Server`.

Lesson 4: Scheduling reports

In Lesson 3 of this track we ran published reports by using the Run and Advanced Run commands in the JReport Console page. Reports run in this way are on-demand reports. By contrast, JReport Server allows reports to be scheduled, that is, automatically run at one or more designated dates and times.

Scheduled reports can greatly improve the performance of the reporting service in an organization. For example, reports can be run during low network traffic times. Scheduled reports can capture a time period exactly, for example the last day of the quarter.

A scheduled report is not delivered to an end user in the same way an on-demand report is. There are six destinations for scheduled reports:

- Publish to Versioning System
- Publish to File System
- Publish to E-mail
- Publish to Printer
- Publish to Fax
- Publish to FTP

In this lesson, we explore three of these :

- Example 1: Publish a report to the versioning system
- Example 2: Publish a report to the file system
- Example 3: Publish a report to the printer

Example 1: Publish a report to the versioning system

The built-in JReport Server Versioning System is a virtual file system maintained by JReport Server.

In this example, we will set up a schedule to run a report immediately and specify that the generated result be kept for 30 days. Even though we run the task immediately, the result is scheduled to run through the JReport Server scheduling system.

Another advantage of scheduling a report is that it can be saved in the versioning system while normally on-demand reports will not be saved unless explicitly saved by the user.

1. On the JReport Console page, go to the **Public Reports** > **JinfonetGourmetJava** folder.
2. Select the row with the OrderListbyDate_Parameter.cls report, right-click in the row and then select **Schedule** from the shortcut menu.
   The General tab of the Schedule page appears. The report tab OrderDetails is selected by default to be scheduled.

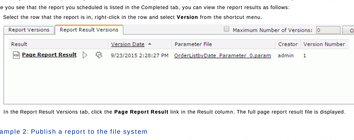
3. Enter **ToVersion** in the Schedule Name text field.
4. Expand the **Report Information** section. From both the Report Version and Catalog Version drop-down lists, choose the most recent report and catalog versions.
5. Expand the **Advanced** section, and select **Use the DB user and password defined in catalog** option if it is not selected.
6. Click **Next** to go to the Parameter tab. If desired, you can enter new parameter values or keep the default values.
7. Click **Next** to go to the Publish tab.
8. In the Publish tab, click **To Version** and then check **Publish to Versioning System**.
   The publishing options appear.

9. Select the **Page Report Result** format, check the **Result Auto-delete** checkbox and set **Result Expires in 30 Days**.
10. Switch to the **Conditions** tab to continue.
11. In the Time sub tab, choose **Run this task immediately** from the Time Type drop-down list. The Time Zone field should represent your current time zone.
12. Click **Finish** to submit the task.

You can monitor the status of the report by viewing the My Tasks page (to access this page, click **My Tasks** on the system toolbar). Reports that are running are listed in the Running tab; when complete they will be moved to the Completed tab. Reports which are scheduled but have not yet run are listed in the Scheduled tab.

Once you see that the report you scheduled is listed in the Completed tab, you can view the report results as follows:

1. Select the row that the report is in, right-click in the row and select **Version** from the shortcut menu.

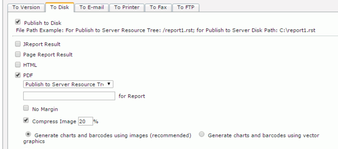


2. In the Report Result Versions tab, click the **Page Report Result** link in the Result column. The full page report result file is displayed.

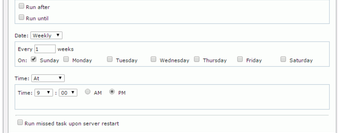
Example 2: Publish a report to the file system

In this example, we set up a schedule to output a weekly sales report to the file system. The report is run every Sunday night so it will be available first thing on Monday morning.

1. On the JReport Console > Resources page, go to the **Public Reports** > **JinfonetGourmetJava** folder.
2. Select the row with the CustomerContactCard.cls report, right-click in the row and select **Schedule** from the shortcut menu.
3. In the General tab, enter **ToPDF** as the schedule name and select the report tab **CustomerInformation** in the report.
4. Click the **Publish** > **To Disk** tab, and then check **Publish to Disk**.
5. Check **PDF**. The PDF options appear.



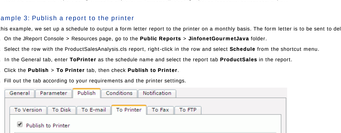
6. From the drop-down list, choose **Publish to Server Resource Tree** and directly below the list enter **/CustomerContactCard.pdf**.
   This publishes the report to the server resource tree under the name /CustomerContactCard.pdf.
   The Server Disk Path is set by default to <install_root>/reports, thus if we want to place the output file in this folder the path must start with a "/". Alternatively we can enter a full disk path if we want the file to be created outside of the server resource tree, for example, C:\temp\CustomerContactCard.pdf.
7. Select the **Conditions** tab to specify the time when the task is to be performed.
8. In the Time sub-tab, choose **Run this task periodically** from the Time Type drop-down list. Select **Weekly** from the Date drop-down list and define to repeat every one week on **Sunday**. Select **At** from the Time drop-down list and specify the time as **9:00 PM**.



   Although we won't use it in this lesson, the Notification tab allows you to specify e-mail notifications to be sent automatically when the report is finished or if it was unable to complete successfully.
9. Click **Finish** to submit the scheduled report request.

Since it isn't practical to wait until Sunday night to see the report in our lesson, we're going to cheat a bit and request an early version of the report.

1. Click **My Tasks** on the system toolbar. The report we requested above should appear as a scheduled task.

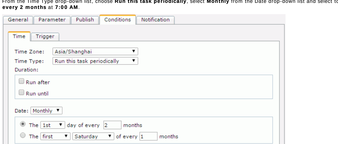


2. Select the task and click **Run** on the task bar of the My Tasks page to immediately schedule the report.
   Now we can find the report using Windows Explorer in <install_root>\jreports\CustomerContactCard.pdf.

Example 3: Publish a report to the printer

In this example, we set up a schedule to output a form letter report to the printer on a monthly basis. The form letter is to be sent to delinquent customers.

1. On the JReport Console > Resources page, go to the **Public Reports** > **JinfonetGourmetJava** folder.
2. Select the row with the ProductSalesAnalysis.cls report, right-click in the row and select **Schedule** from the shortcut menu.
3. In the General tab, enter **ToPrinter** as the schedule name and select the report tab **ProductSales** in the report.
4. Click the **Publish** > **To Printer** tab, then check **Publish to Printer**.
5. Fill out the tab according to your requirements and the printer settings.



6. Click the **Conditions** tab, and then click the **Time** sub tab to specify the time when the task is to be performed.
7. From the Time Type drop-down list, choose **Run this task periodically**, select **Monthly** from the Date drop-down list and select to run the task on **The 1st day of every 2 months** at **7:00 AM**.

8. Click **Finish** to enter the scheduled report request.
9. Click **My Tasks** on the system toolbar. When the task is running, you can see the record of this task in the Running table and it will be put into the Completed table when the task is finished by the server. You probably want to go to the Scheduled tab and delete this entry so it does not print to your printer every month!
10. Exit the JReport Console page by clicking the **Logout** link.

Lesson 5: Administration of resources

JReport Server provides Administration pages that allow the system analyst, report administrator, or application server administrator to manage the resources and operations of JReport Server. This lesson describes the administrative tasks that you can perform through the Administrative pages.

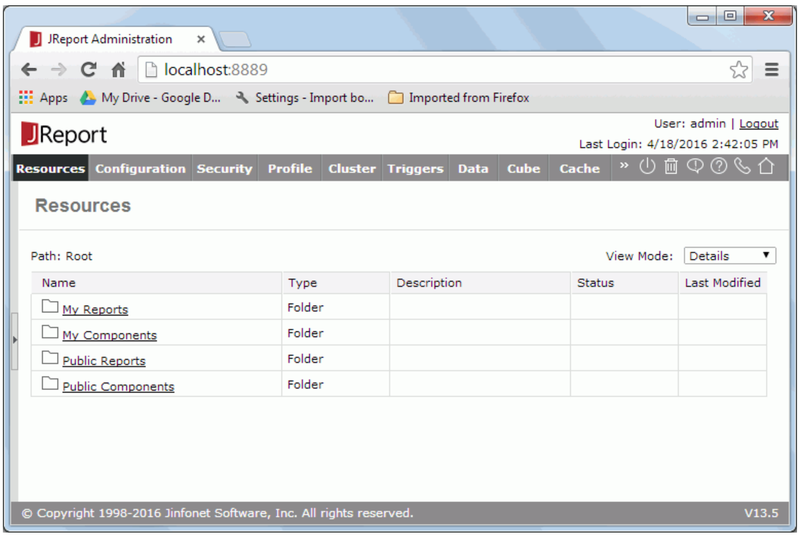
- Task 1: Access the JReport Administration page
- Task 2: Review the Resources page
- Task 3: Review the Configuration page
- Task 4: Understand the Monitor page
- Task 5: Review the remaining pages

Task 1: Access the JReport Administration page

1. Click **Start** > **All Programs** > **JReport 13.5 Update 2** > **System Admin**, or open a web browser and set the URL to http://localhost:8889.
2. In the Sign in dialog, enter **admin** for the user name and **admin** for the password, then click **Login**.

   These are the initial administrative user credentials built in to JReport Server. One of the first things an administrator should do is to set up the correct user credentials so that appropriate access to the reports is granted.

   The JReport Administration page is displayed:



Task 2: Review the Resources page

The Resources panel of JReport Administration lists the resources that have been published to JReport Server.

The User Directory is hidden on the left showing the folders of published resources. By default, the root folder is open when you first log onto JReport Administration page.

The resource list shown in the Resources panel is similar to that in the JReport Console; however, the report name is not a link and there isn't an interface to run a report.

The Versions button allows you to set the maximum number of versions of report results to keep in the JReport Version library. When reports are run, either in Advanced mode or scheduled, the report results are stored in the Version library. The report administrator can control how many versions to maintain for each report.

The NLS Editor button allows you to translate a report into a different language from the original one.

Another two buttons, Properties and Delete , are the same as described in Lesson 1 of this track.

Task 3: Review the Configuration page

The Configuration page allows you to control the operation of JReport Server as follows:

- The Service panel is used for setting attributes of the HTTP connection.
- The Log panel gives you fine grained control of what information is logged.
- The Performance panel allows you to specify preloading particular catalogs and reports in cache so they will be faster to access the first time they are used.
- The Advanced panel is a miscellaneous set of options and settings.
- The LDAP panel allows you to configure LDAP as a security provider for JReport Server. A large number of different LDAP products are supported. In order to use them, a detailed knowledge of how to talk to your LDAP product is required.
- The Export panel allows you to configure the default settings for exporting report results to e-mail, to configure your FAX hardware and software settings, and to define the precision level with which the report result will be exported. You cannot distribute reports via e-mail or fax until these settings are configured.
- The Connection panel allows you to reload the connection settings after datasource.xml file is modified when JReport Server is running.
- The Connection Pool panel is used to keep cached connection for reuse.
- The Upload panel allows you to specify the allowed image size and types for the images which are uploaded from the local file system when end users insert images into reports in Page Report Studio.
- The Custom Field panel allows you to define and manage user defined fields which can be used as resource properties in JReport Server.
- The Dashboard Listener panel is used for managing implementations of the Dashboard Listener API.
- The Dynamic Connections panel is used for managing dynamic connections.
- The Dynamic Display Names panel is used for managing dynamic display names for business view elements.
- The Organizations panel is used for organizing users into different groups.
- The E-mail Server panel allows you to set up the linkage to your SMTP server to enable sending report results and notifications via e-mails.

Task 4: Understand the Monitor page

JReport Server Monitor allows system administrators to log, monitor and report on real-time system status and the performance of the JReport Server. It can be accessed via any web browser, enabling administrators to remotely monitor and manage JReport Servers.

JReport Server Monitor allows you to track the status of JReport Server, see what reports are running or have run as well as which users are accessing the system. JReport Server Monitor is downloaded separately from JReport Server, and must be installed before this page can be used.

Task 5: Review the remaining pages

- **Security**
  Allows you to create and manage realms, users, groups, roles, privileges, and alias. This is required if you are going to use My Reports so that users will only see their own reports in the My Reports folder. For more details about security, see Lesson 6 in this track.
- **Profile**
  Allows you to customize general settings of JReport Server, Page Report Studio, Web Report Studio and JDashboard. The Customize Profile panel is particularly useful to control the options that are displayed when users view reports in Page Report Studio, or dashboards in JDashboard.
- **Cluster**
  Allows you to set up groups of servers in a clustered environment.
- **Triggers**
  Allows you to create and manage triggers.
- **Data**
  JReport Server needs to read and write to a DBMS system to store information about the server and the available reports and report version. By default the DBMS is the 100% Java Apache Derby DBMS embedded in JReport. This can easily be changed to use your preferred DBMS such as Oracle or MySQL.
- **Cube**
  Creates and manages in-memory cubes for business views.
- **Cache**
  Allows you to cache data, reports, security objects, and images. These settings affect how quickly resources can be accessed; however, the more resources that are cached, the more memory is used and thus would harm performance rather than increase performance.

Exit JReport Administration pages by clicking the Logout link in the upper right corner of the page.

Lesson 6: Security

JReport Server provides a security system that protects the resources on the server from inappropriate access by users. In addition to controlling resource-level access, the security system can control access to certain fields or certain information based on the user's role or group level information.

As an alternative to the built-in security system, an LDAP Server can be used for JReport Security. JReport provides interfaces to synchronize with a variety of LDAP servers as well as directly authenticate to an LDAP directory.

Conversely, a Security token can be passed to JReport for authentication. JReport provides an interface for authentication.

- Task 1: Create a new user
- Task 2: Enable auditing for the user
- Task 3: Assign permissions to the user

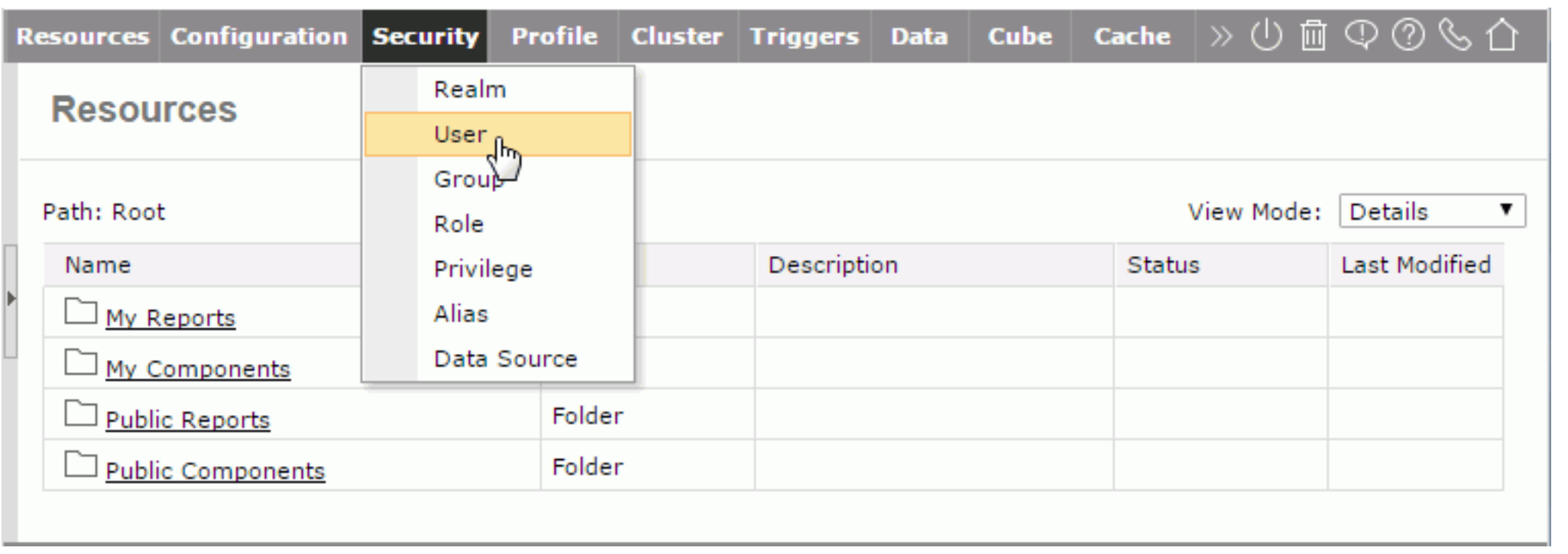
Task 1: Create a new user

To use JReport Server, you must have a user account, which consists of a unique user name and a password. JReport Server verifies your identity when you type your user name and your password and then log on. If your user account has been disabled or deleted, JReport Server prevents you from accessing the web services that JReport Server provides, in order to ensure that only valid users can access.

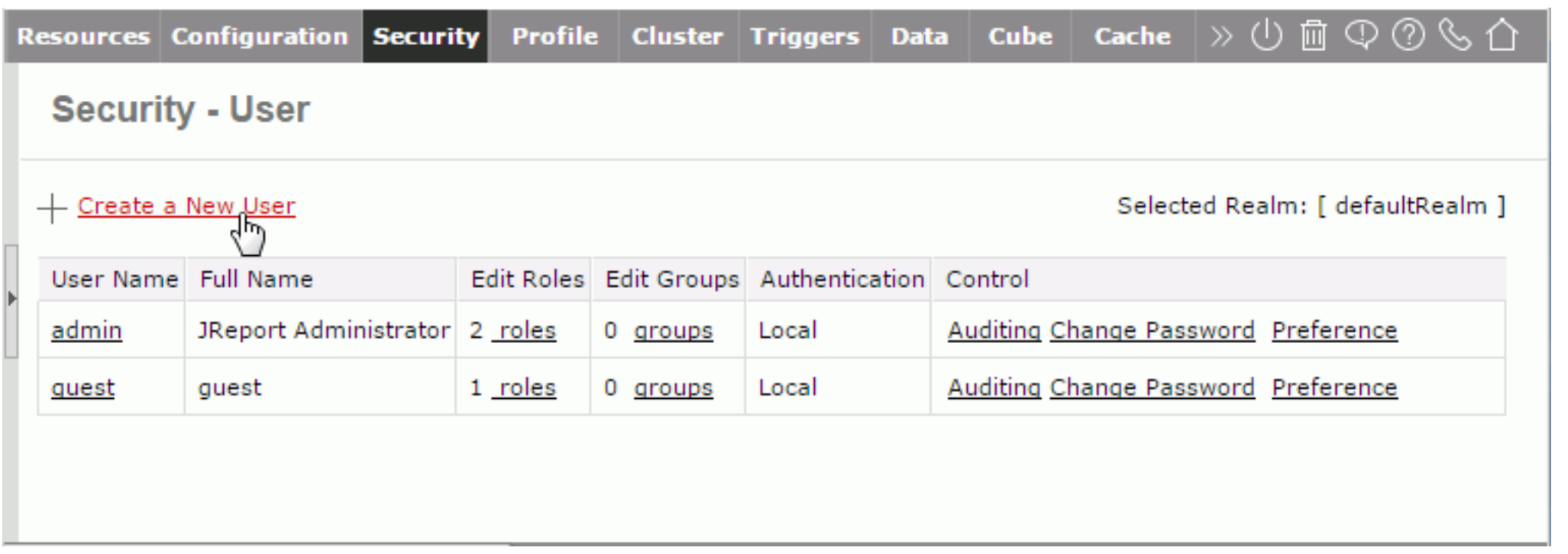
JReport Server comes with two built-in user accounts, which are admin and guest. The built-in user accounts cannot be deleted. The admin user account can neither be deleted nor disabled.

To create a user account:

1. On the JReport Administration page, click **Security** on the system toolbar, and then select **User** from the drop-down menu.



2. Click the **Create a New User** link.



3. Complete the New User fields with any user information you would like to use and click **OK** to create the user.

   Although it doesn't matter for this lesson, the Publish privilege is an important aspect of user definition. Users can either be granted or denied the ability to publish resources to the JReport Server based on the checkbox.

   Make a note of the user credentials you specified in this step so that you can access them in the remainder of this lesson.

4. The newly-created user is listed in the User panel.

   Users can be grouped. Often a set of users require the same security privileges. That is, everyone in the Sales organization can view the Sales reports. By creating a named group, you can efficiently manage the security of a set of users.

Task 2: Enable auditing for the user

JReport Server can record user access and management information in the log files by auditing the user.

To audit the user:

1. Locate the user you created in Task 1, and click the corresponding **Auditing** link in the Control column of the user table.
2. In the Auditing dialog, click the **OK** button to accept the supplied defaults.

   Events are recorded in the log file.

Task 3: Assign permissions to the user

Permissions, associated with resources and folders which locate in the Public Reports or Public Components folder, are the rules that are granted to users to control their access to resources and folders. The permissions include: Visible, Read, Write, Execute, Schedule, Delete, and Grant.

After you set permissions on a parent folder, new resources and sub folders created in the folder inherit these permissions. If you do not want them to inherit permissions, enable their user permissions and set their permissions separately. The resources and folders will inherit permission from their parent folder if their user permissions are not enabled.

To set, view, change, or remove resource permissions:

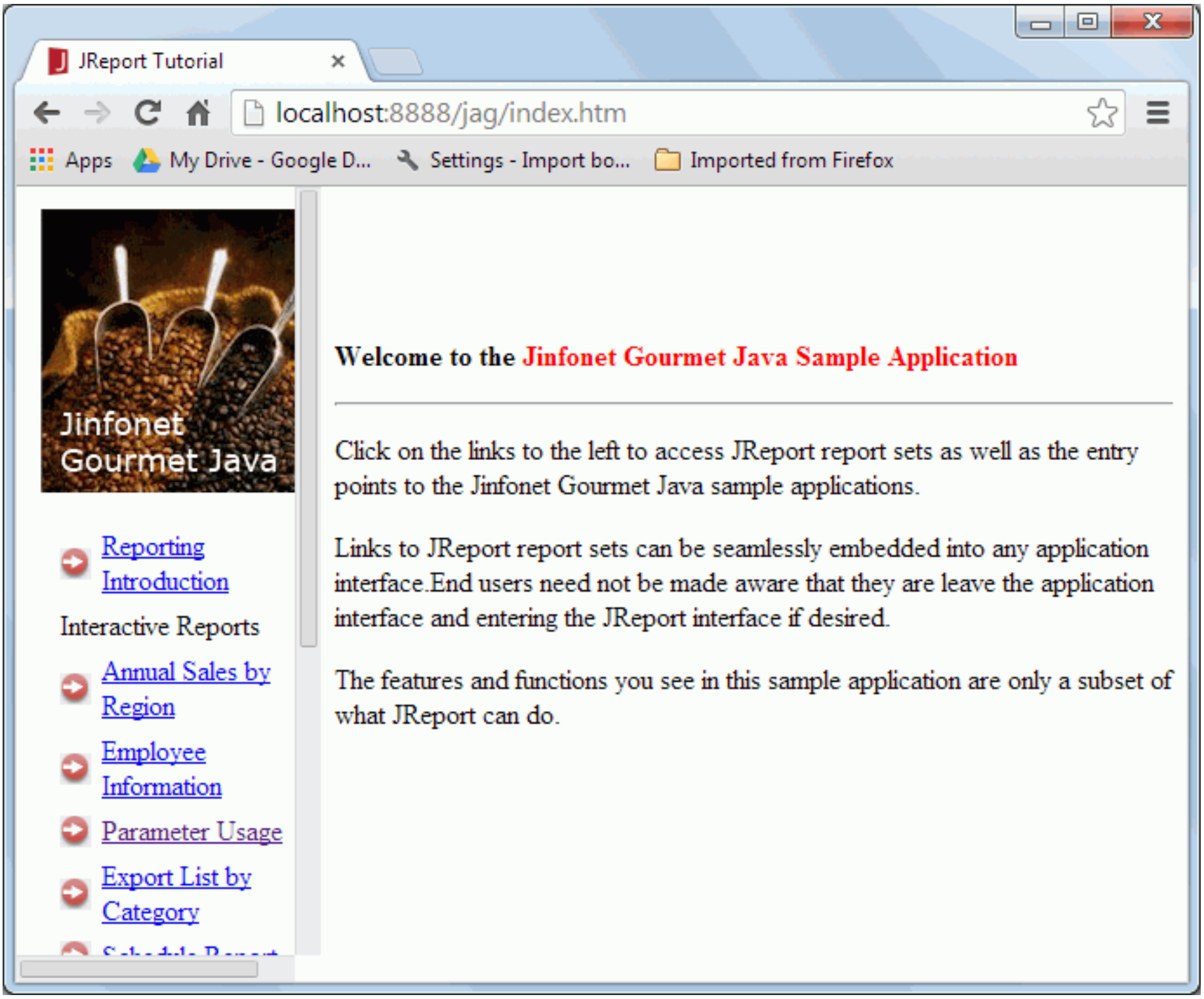
1. On the JReport Administration page click **Resources** > **Resources** on the system toolbar.
2. In the Resources panel, browser to the resource on which you want to set permissions.
3. Click the **Properties** button in the Control column of the resource.
4. In the Properties dialog, click the **Set Permissions** link.
5. In the Set Permissions dialog, set the permissions according to your own requirements, then click **OK** to confirm.

Users must have certain user rights and permissions to perform tasks on certain resources. Roles help you efficiently assign those user rights and permissions to users. Assigning one or more roles to users gives the users all of the user rights and permissions the roles have to perform their jobs with. A role can also be assigned to other groups or roles, and hence the groups or roles will inherit the resource and folder permissions that the roles have.

JReport Server comes with two built-in roles, which are administrators and everyone. The built-in roles cannot be deleted.

Track 5: Integrating reports with JSP

For this track, assume that you are a report developer working once again for the Jinfonet Gourmet Java company. You have been asked to integrate various JReport-developed reports into JSP pages that are part of a sales order application. The links to generate the reports should appear on the same JSP page that is the entry point to the application:



Accessing reports from a JSP page is just one way that a report can be embedded. A report can also be accessed directly from a Java application, servlet or EJB by using the JReport Server API or the JReport RMI API. Reports can also be called by simple URL hyperlinks from HTML pages, as described in the next lesson.

Note: The lessons in this track assume that the Jinfonet Gourmet Java catalog has been published as described in Task 1 of the Publishing resources lesson.

In this track, you perform the following lessons:

- Lesson 1: Calling a page report from HTML
- Lesson 2: Calling a page report from JSP
- Lesson 3: Running a report with parameters
- Lesson 4: Exporting a report from JSP
- Lesson 5: Scheduling a report from JSP
- Lesson 6: Application integration

Click the right arrow (⇨) below to begin reading the track.

Lesson 1: Calling a page report from HTML

You have been asked to embed AnnualSalesbyRegion.cls with the Jinfonet Gourmet Java Sales Orders application. This report shows annual sales information. The report will be called from a JSP page in the application.

To access this report without being embedded, the user has to leave the application interface, open a new web browser window, point it to the JReport Console page, locate the report, and then run it. Embedding the report in the application is a more convenient way to present the report. By simply putting a link or a button in the application interface, users can run the report on demand. The report will be displayed in a new web browser window in Page Report Studio and not affect the state of the application.

To access JReport functionality from a JSP page, you use the JReport JSP Tag Library. The JReport JSP Tag Library provides a collection of reusable embedded and component tags that can be incorporated into web applications. Using these tags, developers can easily embed JReport's entire Page Report Studio interface or components of it directly into an application interface in minimal time without resource intensive efforts. Component tags can be used for:

- Specifying report outputs.
- Adding toolbars and menu bars to the report, providing easy access to commonly-used functions and commands.
- Adding an interactive table of contents to the report, providing dynamic navigation to any component or section.

This lesson is a simplified version of integration. Extend what this lesson shows to see the full power of JReport Server running in either of its two modes: as an integrated part of your application or as a standalone report server independent of the application.

- Task 1: Access the JSP file that runs the report
- Task 2: Run the report using an HTML file that prepares the Tag lib parameters

Task 1: Access the JSP file that runs the report

Any JSP page can call the report. For this lesson, a simple JSP page is used. To view the .jsp file, follow these steps:

1. Create a new directory named jag in `<install_root>\public_html`.
2. Browse to and copy runReport.jsp in `<install_root>\help\samples\JSPSamples\JinfonetGourmetJavaDemo` to `<install_root>\public_html\jag`.
3. In your favorite HTML editor, open runReport.jsp and review the JSP code. It contains:

```
<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN">
<%@ page import="jet.web.dhtml.WebActLogic"%>
<html>
<head>
<title>Demo</title>
</head>
<BODY>
<iframe style="width:100%; height:100%;" align="middle" scrolling="yes" src="../
dhtmljsp/dhtml.jsp?jrs.cmd=jrs.try_vw&jrs.report=/JinfonetGourmetJava/
AnnualSalesbyRegion.cls&jrs.catalog=/JinfonetGourmetJava/JinfonetGourmetJava.cat&"
/>
</BODY>
</HTML>
```

   Note: The sc string value is broken for display purposes only in this document. In the actual .jsp file the value is continuous.

4. Start JReport Server by clicking **Start** > **All Programs** > **JReport 13.5 Update 2** > **Start Server**.
5. Open a web browser window and access http://localhost:8888/jag/runReport.jsp. First log in and then the report is displayed.

Task 2: Run the report using an HTML file that prepares the Tag lib parameters

This task provides an example of how the parameters to the main page tag interact with each other to control the report output. In this example, you enter values for each parameter into an HTML page before running the report. A more likely scenario for a real application is to have the to application code specify the parameters rather than prompt the user for them.

1. Copy setmain.html in `<install_root>\help\samples\JSPSamples\JinfonetGourmetJavaDemo` to `<install_root>\public_html\jag`.

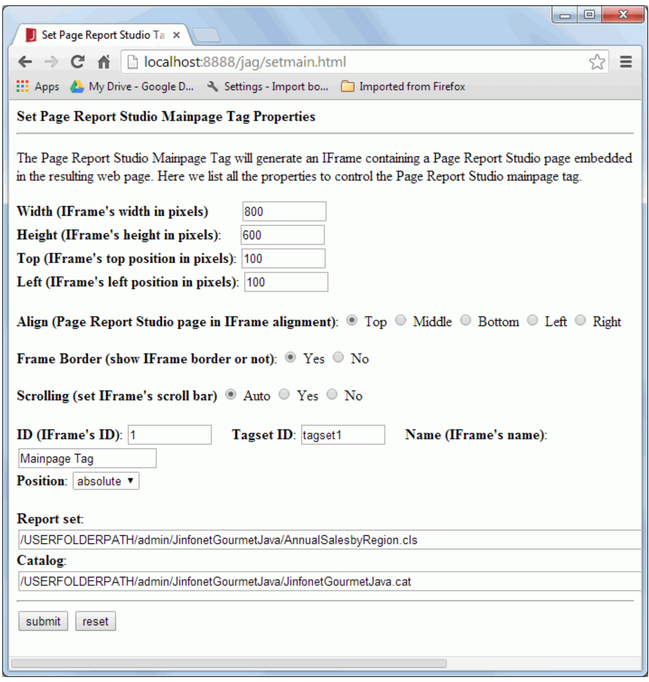
   Review the code if desired. This code creates a frame and prompts for the parameter values that the main page tag requires. When you click the submit button the parameters are passed to the JSP page and the report is run.

2. Copy demomain.jsp in `<install_root>\help\samples\JSPSamples\JinfonetGourmetJavaDemo` to `<install_root>\public_html\jag`.

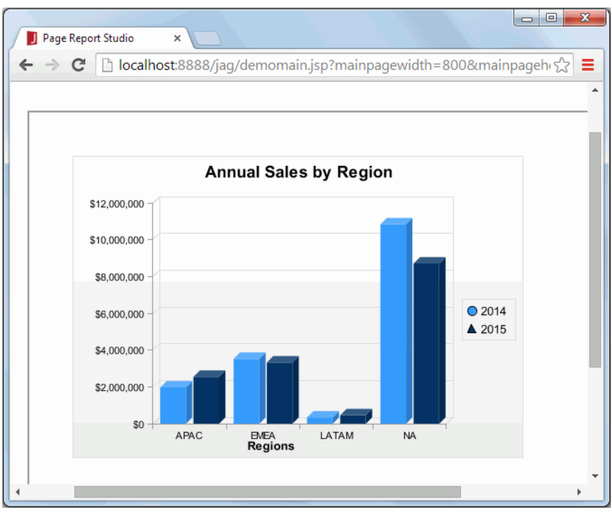
   Review the code if desired. This is very similar to the code from Task 1 except that the values are passed in from the HTML page and the report we run is the AnnualSalesbyRegion.cls.

3. Open a web browser and access http://localhost:8888/jag/setmain.html.

   The Set Page Report Studio Taglib Properties page appears:



4. Click **submit** to run the report with the defaults.



If desired, repeat the steps in this task and experiment with different values on the Set Page Report Studio Taglib Properties page to learn how they affect the way the report is displayed in the Frame (the Tag set ID must be a new one for each running).

Lesson 1 summary

In this lesson you have learned how to use HTML and JSP to run a report using the JReport JSP Tag Library.

Lesson 2: Calling a page report from JSP

This lesson assumes that you have been asked to integrate ProductSalesAnalysis.cls with a JSP page in the Sales Order application. However, you need more flexibility than the JReport mainpage tag provides. This lesson will display ProductSalesAnalysis.cls in Page Report Studio in a new web browser window.

In this lesson you will use Page Report Studio API to run the report instead of using the tag library. The JSP page that runs the report is provided as part of the lesson.

1. Browse to and copy the runProductSalesAnalysis.jsp file in `<install_root>\help\samples\JSPSamples\JinfonetGourmetJavaDemo` to `<install_root>\public_html\jag`.

   Create the jag directory if you did not already do so in Lesson 1.

2. Open runProductSalesAnalysis.jsp in your favorite editor or an editor that recognizes JSP such as Eclipse or Dreamweaver.

```
<%@ page import="jet.web.dhtml.*" %>
<%@ page import="java.util.*" %>
<!-- include the jsp for check user login -->
<%@ include file="../dhtmljsp/AuthCheck.jsp" %>
<%
        Hashtable htparams = new Hashtable();
        //set repor info
        htparams.put("jrs.cmd", "jrs.try_vw");
        htparams.put("jrs.result_type", "8");
        htparams.put("jrs.report", "/JinfonetGourmetJava/ProductSalesAnalysis.cls");
        htparams.put("jrs.catalog", "/JinfonetGourmetJava/JinfonetGourmetJava.
cat");

        Map tmpmap = DHTMLUtil.runReport(request, htparams);
        String targetURL = null;
        String errorMsg = null;

        if(tmpmap != null){
                errorMsg = (String)tmpmap.get(DHTMLConstant.ERROR_MSG);
                targetURL = (String)tmpmap.get(DHTMLConstant.TARGET_URI);
                if(errorMsg != null && !errorMsg.trim().equals("")){
                        PrintWriter jwout = response.getWriter();
                        jwout.print(errorMsg);
                        return;
                }else{
                        //forward to refresh jsp
                        if(targetURL != null && !targetURL.trim().equals("")) {
                                response.sendRedirect(targetURL);
                        }
                }
        }
%>
```

3. Open your browser to http://localhost:8888/jag/runProductSalesAnalysis.jsp.

   The report is displayed according to the specifications provided.

Lesson 2 summary

In this lesson you have learned how to use JSP and Page Report Studio API to run a report.

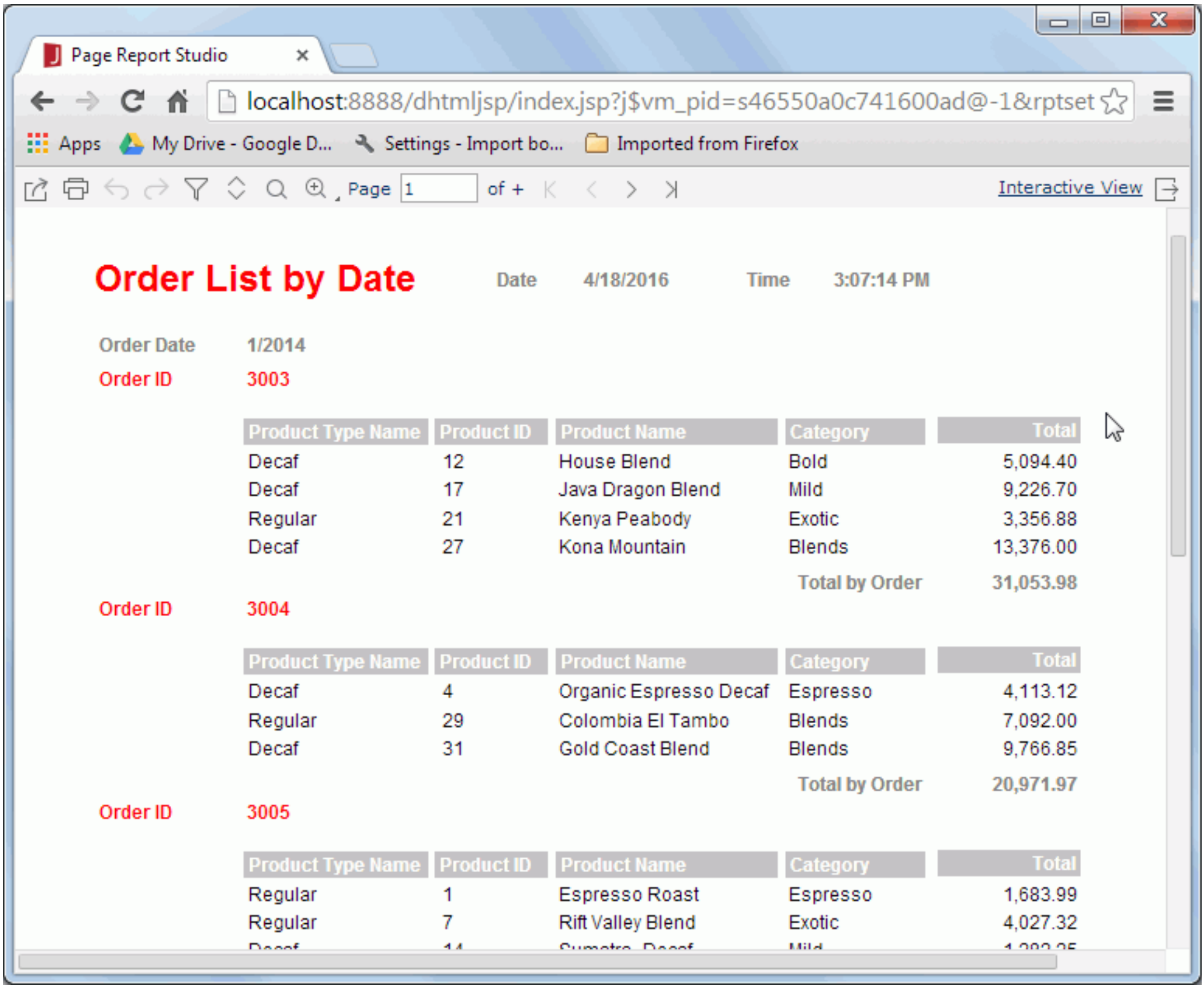
Lesson 3: Running a report with parameters

You have been asked to run OrderListbyDate_Parameter.cls from your JSP application and this report requires parameters to run. You don't want JReport Server to display the default JSP page to enter the parameter value so you need to write a JSP page to request the parameter values, and then run the report with the specified values.

In this lesson you will also need to use Page Report Studio API to run the report without using the tag library.

This lesson comes with an HTML file that requests the parameters and a JSP file that runs and displays OrderListbyDate_Parameter.cls in Page Report Studio in a new web browser window.

1. Copy the runWithParameter.html file in `<install_root>\help\samples\JSPSamples\JinfonetGourmetJavaDemo` to `<install_root>\public_html\jag`.
2. Open runWithParameter.html in your favorite editor and review the code if desired.
3. Copy runWithParameter.jsp in `<install_root>\help\samples\JSPSamples\JinfonetGourmetJavaDemo` to `<install_root>\public_html\jag`.
4. Open this file in an editor and review the code if desired.
5. Browse http://localhost:8888/jag/runWithParameter.html.
6. Select the start date and end date and click **Run report**. You may see the result generated with the specified parameter values, for example:



Lesson 3 summary

In this lesson you have learned how to use JSP and Page Report Studio API to run a report with parameter collected through the user interface. Often this information is known by your application so there is no need to prompt the user for the information, it can be provided directly from the application calling the report.

Lesson 4: Exporting a report from JSP

You have been asked to run EmployeeInformation.cls from the Sales Order application and allow the user to choose the format of the report. The allowable formats are: PDF, Excel, RTF (Microsoft Word), HTML, Text, PostScript, and XML.

In this lesson you again use Page Report Studio API to run the report; however, we will use the One Step Export feature to export the page report in the specified format without displaying it in a separate web browser window.

- Task 1: Review the JSP file to run the report
- Task 2: Export the report result

Task 1: Review the JSP file to run the report

The JSP page that runs the report is provided as part of the lesson. Follow these steps to review the JSP code:

1. Copy exportReport.html in `<install_root>\help\samples\JSPSamples\JinfonetGourmetJavaDemo` to `<install_root>\public_html\jag`.

2. Open exportReport.html in your favorite editor.

```
<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.0 Transitional//EN">
<html>
<head>
<title>Export Report to user selected format</title>
<script id="j$vm" type="text/javascript" src="../webos/jsvm/lib/
jreportapi.js"></script>
<script src="../webos/jsvm/lib/jquery-1.5.2.min.js"></script>
<script language="JavaScript">
(function(){
var thi$ = self;
var server = {
url: "../jinfonet/tryView.jsp",
user: "admin",
pass: "admin",
jrd_prefer:{
// For page report
pagestudio:{
feature_UserInfoBar:false,
feature_ToolBar: false,
feature_Toolbox: false,
feature_DSOTree: false,
feature_TOCTree: false,
feature_PopupMenu: false,
feature_ADHOC: false
},

// For web report
wrptstudio:{
viewMode:{
hasToolbar: false,
hasSideArea: false
}
}
},
jrd_studio_mode: "view",
"jrs.param_page": true
},

prptRes = {name:"/JinfonetGourmetJava/EmployeeInformation.cls"},
catRes = {name:"/JinfonetGourmetJava/JinfonetGourmetJava.cat"},

params = {
};

var Factory = com.jinfonet.api.AppFactory;
var entryId="report1";

thi$.openPageReport = function(){
var app = Factory.runReport(
server, prptRes, catRes, params, entryId);

app.attachEvent("jreportstatus", 4, this, function(e){
var msg = e.getData();

});
};


thi$.guiExport = function(){
var app = Factory.getApp(entryId), rptset;
if(!app) return;

rptset = app.getReportSet();
rptset.exportTo();
};

})();

</script>
</head>
<body >
<input type="button" value="Open Report" onClick="openPageReport
();">
<input type="button" value="Export Report" onClick="guiExport
();">
<div id="report1" style="position:relative;height:640px;"></div>
</body>
</html>
```

Note: The result types are: 0- HTML, 2 - PDF, 3 - Postscript, 4 - RTF, 5 - TEXT, 6 - Excel, 7 - XML.

Task 2: Export the report result

1. Browse http://localhost:8888/jag/exportReport.html.

2. Choose the desired export format and whether you want to download the result or view it now.

3. Click **Export** to generate the exported report.

   The report displays according to the format provided. For downloaded result, use the program on your system to view the results, such as Acrobat Reader or Microsoft Word.

Lesson 4 summary

In this lesson you have learned how to use JSP and Page Report Studio API to run a report and export the current report tab in this report to any format.

Lesson 5: Scheduling a report from JSP

You have been asked to run EmployeeInformation.cls from the JSP application and allow the user to schedule the report from the JSP application. We will use an HTML page to ask the user for the scheduling information and then run the report simply by calling submitScheduledTask().

This lesson needs an HTML and a JSP file:

1. Copy scheduleRequest.html in `<install_root>\help\samples\JSPSamples\JinfonetGourmetJavaDemo` to `<install_root>\public_html\jag`.

2. Open scheduleRequest.html in your favorite editor.

```
<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.0 Transitional//EN">
<html>
<head>
<title>Schedule a report set to run once at a future time.</title>
</head>
<body>
<p>Schedule running EmployeeInformation.cls in PDF format and publish the pdf file to the servers file
system:
<form method=get action="scheduleReport.jsp">
        Specify the pdf file path and name :
        <input type="text" name="filepath" value="C:\temp\" size="30">
        <input type="text" name="filename" value="EmployeeInformation.pdf" size ="20"><br>
Run this task on: <input type="text" name="month" value="12" size="5"> (Month)
                  <input type="text" name="day" value="31" size="5"> (Day),
                                  <input type="text" name="year" value="2014" size="7"> (Year), at
                                  <input type="text" name="hour" value="17" size="5"> (Hour)
                                  <input type="text" name="min" value="0" size="5"> (Minute) <br><br><br>
<input type="submit" value="Schedule this task">
</form>
</body>
</html>
```

3. Copy scheduleReport.jsp in `<install_root>\help\samples\JSPSamples\JinfonetGourmetJavaDemo` to `<install_root>\public_html\jag`.

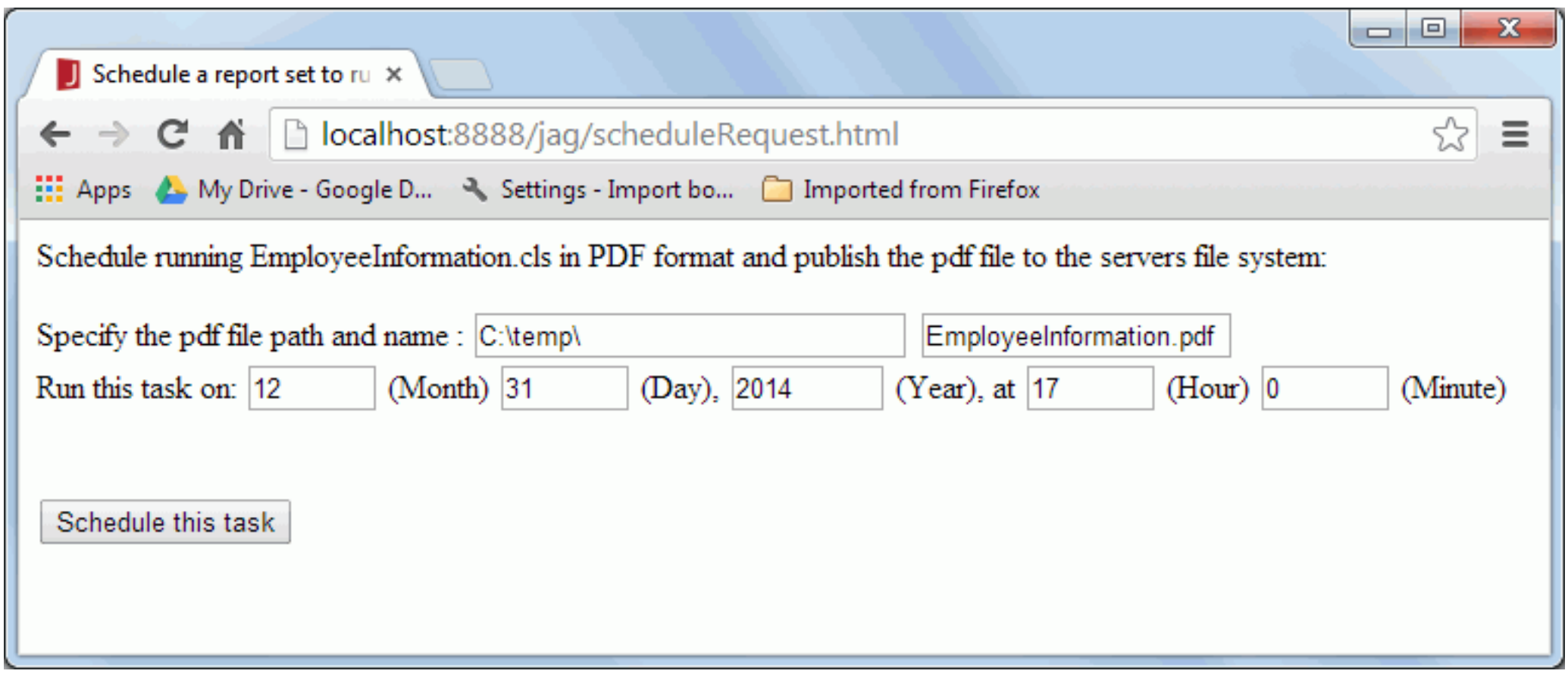
4. Open scheduleReport.jsp in your favorite editor to view it.

```
<%@ page import="jet.web.dhtml.*"%>
<!-- specify the error page for this is jsp, a user can customize it and it's optional. -->
<%@ page errorPage="../dhtmljsp/errorpage.jsp" %>
<!-- include the jsp for check user login -->
<%@ include file="../dhtmljsp/AuthCheck.jsp" %>
<html>
<head>
<title> Schedule according to date and time provided by scheduleRequest.html </title>
</head>
<body>
<%
        //get RptServer
        HttpRptServer server =  jet.server.api.http.HttpUtil.getHttpRptServer(request);
        Properties props = new Properties();
        props.put("jrs.cmd", "jrs.submit_schedule");
        props.put("jrs.task_class", "jet.server.schedule.jrtasks.PublishToDiskTask");
        props.put("jrs.to_disk", "true");
        props.put("jrs.to_disk_pdf_path_type", "1");
        props.put("jrs.result_type", "2");
        props.put("jrs.to_pdf", "true"); // run in PDF format
        props.put("jrs.launch_type", "1");
        props.put("jrs.catalog", "/JinfonetGourmetJava/JinfonetGourmetJava.cat");
        props.put("jrs.report", "/JinfonetGourmetJava/EmployeeInformation.cls");
        props.put("jrs.auth_uid", "admin"); // set log user ID to avoid login dialog popup
        props.put("jrs.auth_pwd", "admin"); // set log user password to avoid login dialog popup
        props.put("jrs.pdf_dir", request.getParameter("filepath"));
        props.put("jrs.pdf", request.getParameter("filename"));
        props.put("jrs.exe_year", request.getParameter("year"));
        props.put("jrs.exe_month", request.getParameter("month"));
        props.put("jrs.exe_day", request.getParameter("day"));
        props.put("jrs.exe_hour", request.getParameter("hour"));
        props.put("jrs.exe_min", request.getParameter("min"));

        //get user information
        HttpUserSessionManager umg = server.getHttpUserSessionManager();
        UserSession us = umg.getUserSession(request);
        String userId = us.getUserID();

        //Submit a scheduled task and return the task ID.
        try{
                String taskID = server.submitScheduledTask(userId, props);
                out.print("The task has been scheduled. The task ID is:" + taskID);
        }catch(Exception e){
                out.print(e.getMessage());
        }
%>
</body>
</html>
```

5. Browse http://localhost:8888/jag/scheduleRequest.html, and the following page appears:



6. Specify the current date, but choose a time about 10 minutes from the current time. You can change the default file name and scheduled location if desired.

   Note: Only whole numbers can be specified for each field.

7. After the specified time, view the report result in its target location.

   Scheduling reports in this way has two advantages:

   - The data captured by the report can correlate to a specific window, that is, a report of monthly activity can be generated at the exact minute the month is over.
   - JReport Server can operate at off hours, thereby allowing more throughput during peak time.

Lesson 5 summary

In this lesson you have learned how to use JSP and JReport Server API to run a report according to a schedule provided by an application.

Lesson 6: Application integration

You have been asked to incorporate the reports from the previous lessons of this track into an application portal. The application portal allows the staff concerned to easily request the reports they need as well as access the applications they use on a daily basis. For this lesson, the application portal was built using frames; however, in most corporate settings there would be an application portal which would be used to launch the application and common reports. The techniques and code required to call the reports are identical whether it is a portal or a simple menu so the sample code can be utilized in either situation.

- Task 1: View the source for the application menu
- Task 2: View the application menu

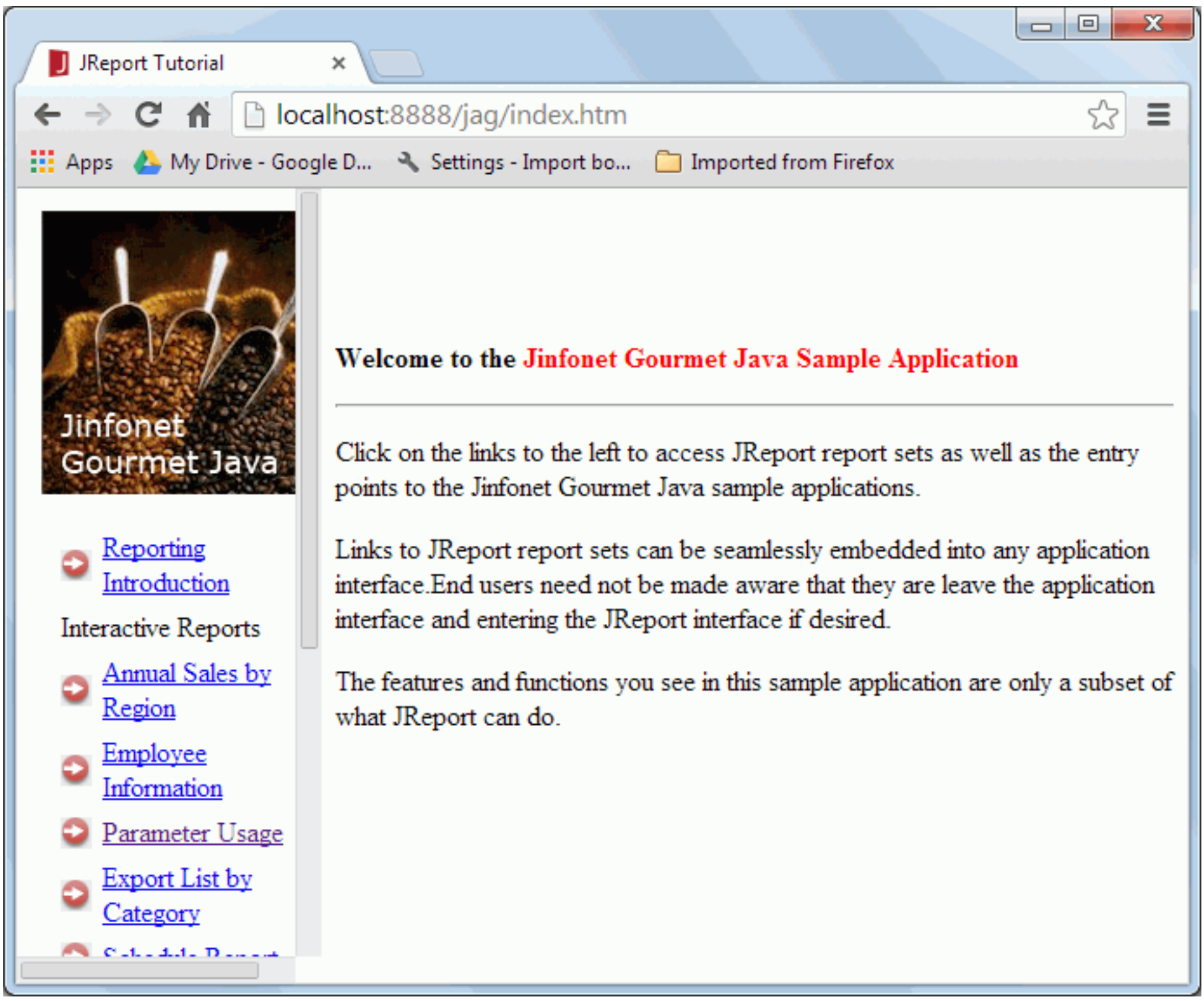
Task 1: View the source for the application menu

This lesson includes HTML which creates a menu on the left frame and displays the requested HTML and reports in the right side frame.

1. Copy the contents of the directory `<install_root>\help\samples\JSPSamples\JinfonetGourmetJavaDemo` to `<install_root>\public_html\jag`.
2. Open index.htm in the jag directory in your favorite editor. This HTML file simply creates two frames, namely leftFrame which displays left.htm and rightFrameSet which by default has application/introduction.htm displayed.
3. Open left.htm in your editor. You will see it is a table with the menu items we want to present to the user.

Task 2: View the application menu

1. Open a web browser to http://localhost:8888/jag/index.htm. The page is something like this:



2. Click the links in the left frame to view the introduction or reports.

Lesson 6 summary

In this lesson you have learned how to use HTML to integrate the reports into a menu system. This strategy provides a user-friendly way to access reports from an existing and familiar application interface.