TIBCO JASPERREPORTS® SERVER USER GUIDE

RELEASE 6.3

http://www.jaspersoft.com

This is version 0616-JSP63-28 of the *JasperReports Server User Guide*.

TABLE OF CONTENTS

CHAPTER 1 INTRODUCTION TO JASPERREPORTS SERVER

TIBCO JasperReports® Server builds on TIBCO JasperReports® Library as a comprehensive family of Business Intelligence (BI) products, providing robust static and interactive reporting, report server, and data analysis capabilities. These capabilities are available as either stand-alone products, or as part of an integrated end-to-end BI suite utilizing common metadata and provide shared services, such as security, a repository, and scheduling. The server exposes comprehensive public interfaces enabling seamless integration with other applications and the capability to easily add custom functionality.

This section describes functionality that can be restricted by the software license for JasperReports Server. If you don't see some of the options described in this section, your license may prohibit you from using them. To find out what you're licensed to use, or to upgrade your license, contact Jaspersoft.

The heart of the TIBCO Jaspersoft® BI Suite is the server, which provides the ability to:

- Easily create new reports based on views designed in an intuitive, web-based, drag and drop Ad Hoc Editor.
- Efficiently and securely manage many reports.
- Interact with reports, including sorting, changing formatting, entering parameters, and drilling on data.
- Schedule reports for distribution through email and storage in the repository.
- Arrange reports and web content to create appealing, data-rich Jaspersoft Dashboards that quickly convey business trends.

For users interested in multi-dimensional modeling, we offer Jaspersoft® OLAP, which runs as part of the server.

While the Ad Hoc Editor lets users create simple reports, more complex reports can be created outside of the server. You can either use Jaspersoft® Studio or manually write JRXML code to create a report that can be run in the server. We recommend that you use Jaspersoft Studio unless you have a thorough understanding of the JasperReports file structure. See **"Adding Reports Directly to the Repository" on page 163** and the *Jaspersoft Studio User Guide* for more information.

You can use the following sources of information to learn about JasperReports Server:

- Our core documentation describes how to install, administer, and use JasperReports Server and Jaspersoft Studio. Core documentation is available as PDFs in the doc subdirectory of your JasperReports Server installation. You can also access PDF and HTML versions of these guides online from the Documentation section of the Jaspersoft Community website.
- Our Ultimate Guides document advanced features and configuration. They also include best practice recommendations and numerous examples. You can access PDF and HTML versions of these guides online from the Documentation section of the Jaspersoft Community website.

- Our Online Learning Portal lets you learn at your own pace, and covers topics for developers, system administrators, business users, and data integration users. The Portal is available online from Professional Services section of our website.
- Our free samples, which are installed withJasperReports Library, Jaspersoft Studio, and JasperReports Server, are documented online. See **"Adding Reports Directly to the Repository" on page 163** and *Jaspersoft Studio User Guide* for more information.

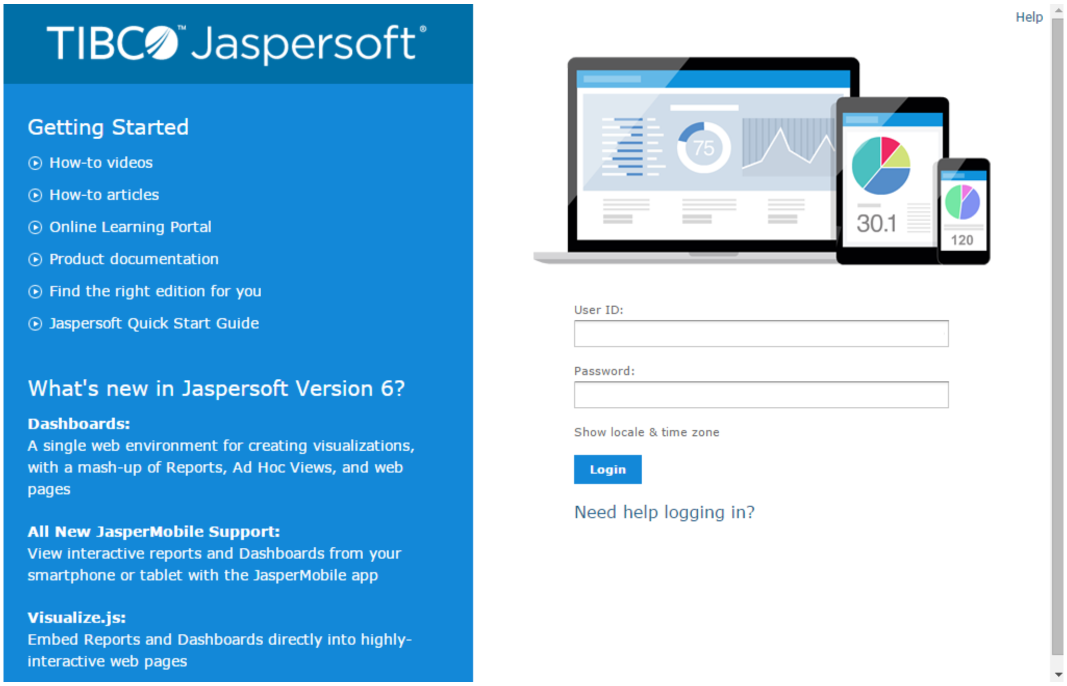
JasperReports Server is a component of both a community project and commercial offerings. Each integrates the standard features such as security, scheduling, a web services interface, and much more for running and sharing reports. Commercial editions provide additional features, including Ad Hoc charts, flash charts, dashboards, Domains, auditing, and a multi-organization architecture for hosting large BI deployments.

This chapter contains the following sections:

- **Logging In**
- **The Getting Started Page**
- **The Library Page**
- **Browsing the Repository**
- **Searching the Repository**
- **Using Repository Resources**
- **Sorting the Repository List**

1.1 Logging In

Launch JasperReports Server by entering `http://<hostname>:8080/jasperserver-pro` in a web browser, where `<hostname>` is the name of the computer that hosts JasperReports Server. The Login page appears.



What's new in Jaspersoft Version 6?

Dashboards:
A single web environment for creating visualizations, with a mash-up of Reports, Ad Hoc Views, and web pages

All New JasperMobile Support:
View interactive reports and Dashboards from your smartphone or tablet with the JasperMobile app

Visualize.js:
Embed Reports and Dashboards directly into highly-interactive web pages

Figure 1-1 Jaspersoft Login Page

To log in to the server, JavaScript and cookies must be enabled in your browser.

Before logging in, review the information on the login page. There are links to the online help and additional resources.

To log in to the server:

1. Enter your user ID and password.

If you installed an evaluation server with the sample data, you can log in with the sample user IDs and passwords. For more information, click **Need help logging in?**

If the Organization field appears in the Login panel, enter the ID or alias of your organization. If you don't know it, contact your administrator. For more information, see **"Logging into a Server with Multiple Organizations" on page 11**.

The default administrator login credentials are superuser/superuser and jasperadmin/jasperadmin.

For security reasons, administrators should always change the default passwords immediately after installing JasperReports Server, as described in the *JasperReports Server Administrator Guide*.

2. If you want to use a different locale and time zone than the server uses, click **Show locale & time zone**.

   The Locale and Time Zone fields appear in the Login panel. Select your locale and time zone from the drop-down menus.

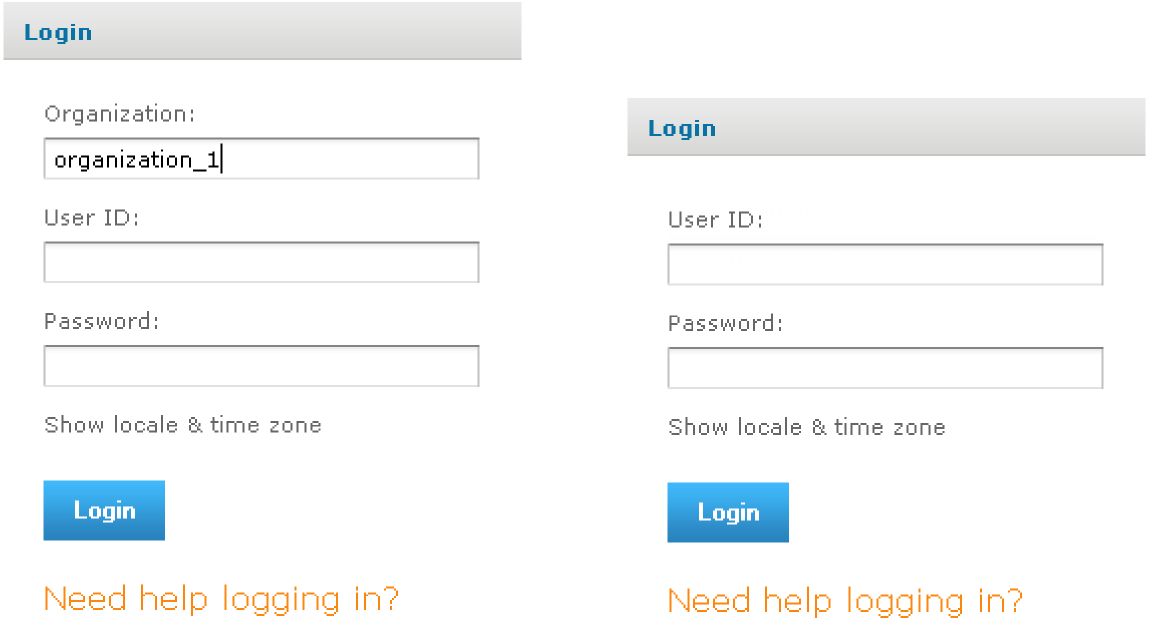
3. Click **Login**.

   If you entered a valid user ID and password, the server displays the Getting Started page, as shown in **Figure 1-3**.

1.1.1 Logging into a Server with Multiple Organizations

If the administrator has configured your server to use the multi-tenancy feature, it supports multiple organizations. Each organization has its own private area for storing files and resources. The default Login dialog for a multi-tenant server has an additional field: **Organization**. The left side of **Figure 1-2** shows this field. Enter the ID or alias of your organization. For example, enter the ID of the default organization: `organization_1`.

You don't have to enter the organization ID each time you log in. The first time you log in, include the organization ID in your login URL, as shown on the right side of **Figure 1-2**. Bookmark the URL and use it for subsequent log ins. The **Organization** field does not appear in the dialog when you specify it in the URL.



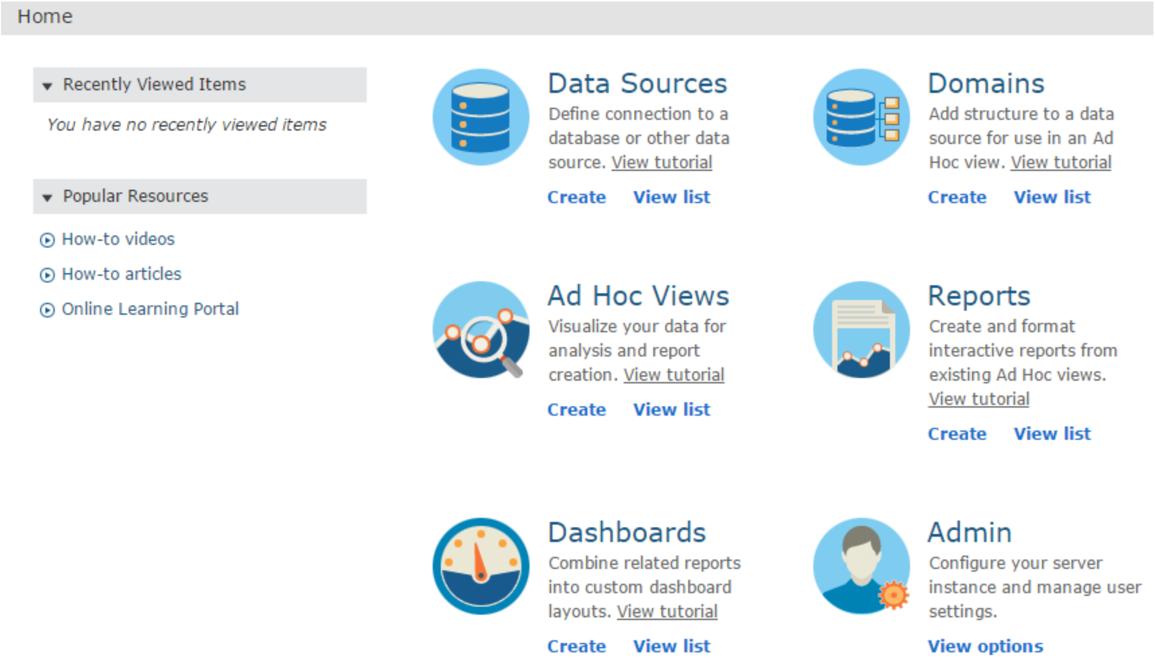
Organization: organization_1

Figure 1-2 Login Methods for Multiple Organizations

The superuser account does not specify an organization because it is the system-wide administrator. If the **Organization** field appears in the Login dialog when you log in as superuser, leave it blank. If you try to log in as superuser with an orgID in the URL, the server returns an error.

1.2 The Getting Started Page

From the Getting Started page, you can quickly access the most frequently used features of the server.



Home

▾ Recently Viewed Items

You have no recently viewed items

▾ Popular Resources

- How-to videos
- How-to articles
- Online Learning Portal

Data Sources
Define connection to a database or other data source. View tutorial
Create View list

Domains
Add structure to a data source for use in an Ad Hoc view. View tutorial
Create View list

Ad Hoc Views
Visualize your data for analysis and report creation. View tutorial
Create View list

Reports
Create and format interactive reports from existing Ad Hoc views. View tutorial
Create View list

Dashboards
Combine related reports into custom dashboard layouts. View tutorial
Create View list

Admin
Configure your server instance and manage user settings.
View options

Figure 1-3 Getting Started Page

1.2.1 Core Workflows

The Getting Started page for standard users has multiple blocks that link to the core workflows of JasperReports Server, that may include some or all of the following options:

- **Data Sources** – Select or define a connection to a database or other data source.
- **Domains**– Add structure to your data source for use in an Ad Hoc view.
- **Ad Hoc Views** – Select or create the visualization for your data.
- **Reports** – Create an interactive report from an Ad Hoc view, or select an existing report.
- **Dashboards** – Combine related reports into one layout, or select from existing layouts.
- **Admin** – Configure your server and manage user settings. This block is visible only to users with administrator privileges.

Each workflow block on the Getting Started page may contain links to video tutorials, pages or wizards to create related elements, and filtered repository lists containing relevant items. Click these links, rather than the blocks themselves, to access these resources. Users with administrator access may have more of these options available to them.

1.2.2 The Getting Started Column

On the left side of the page, there are two lists to help you locate and access relevant information and assets.

- **Popular Resources** – Includes links to educational and support resources.
- **Recently Viewed Items** – Includes links to up to 10 recently viewed repository items, such as reports, Ad Hoc views, dashboards, and the like.

1.2.3 Menu Items

The menu items along the top of the Getting Started page are available from every page on JasperReports Server. 🏠 and the Library, View, Manage, and Create menus offer the options described in the table below.

Menu	Description
🏠	Returns to the Getting Started page.
Library	Displays a pared-down repository page that contains the Ad Hoc views, reports, and dashboards the currently logged-in user has rights to
View	• **Search Results** – Displays the repository of resources filtered by criteria selected in the Filters panel. • **Repository** – Displays the repository of files and folders containing resources, such as reports, report output, data sources, and images. • **Schedules** – Lists the reports that you have scheduled. • **Messages** – Lists system messages, such as an error in a scheduled report. • **UI Samples** – (administrators only) Presents galleries of UI components that you redesign using Themes.
Manage	• **Organizations** – Opens the Manage Organizations page. • **Users** – Opens the Manage Users page. • **Roles** – Opens the Manage Roles page. • **Server Settings** – Opens the Server Settings | Log Settings page.
Create	• **Ad Hoc View** – Launches the Ad Hoc Editor for designing views interactively. • **Report** – Launches the Create Report dialog for creating a report based on an Ad Hoc view and a report template. • **Dashboard** – Launches the Dashboard Designer for laying out multiple reports with input controls, labels, and images. • **Domain** – Launches the Domain Designer for setting up a Domain. • **Data Source** – Launches the New Data Source page for specifying the attributes of the new data source.

If you log in as an administrator, the Home page has additional options and menu items for managing users, roles, organizations, and settings, such as repository folder names. Administrator functions are documented in the *JasperReports Server Administrator Guide*. The links to the **Online Help**, **Log Out**, and a search field appear on all JasperReports Server pages. For more information about searching, see **"Filtering Search Results" on page 18**.

1.2.4 JasperReports Server Keyboard Shortcuts

JasperReports Server provides keyboard shortcuts to help you navigate its web UI without the use of a mouse or other pointing device. These shortcuts are available on the Login page, the Home page, the Library Page, the Repository page, the Search Results page, and the interactive report viewer, allowing you to log in, move between major navigational elements of these page, select and open reports, and navigate tabular reports.

Shortcuts include:

Key	Action
Left Arrow	Navigate left one column, cell, or item.
Right Arrow	Navigate right one column, cell, or item.
Up Arrow	Navigate up one row, cell, or item.
Down Arrow	Navigate down one row, cell, or item.
Enter	Select an item or navigate into an inner control.
Escape	Cancel an action or navigate out to an outer control.
Tab	Navigate to the next major region or form field. Focus moves from left to right.

Note that, on some pages, the Shift key can also be used in conjunction with the arrow keys or Tab:

- When used with the arrow keys, Shift multi-selects items, such as reports listed in the Library page.
- When used with Tab, Shift changes the direction that focus moves from left to right to right to left.

In addition, the web UI has improved compatibility with screen readers, which assist visually impaired users in using computers. The implementation follows the WAI-ARIA (Web Accessibility Initiative Accessible Rich Internet Applications Suite) technical specification, and has been certified for certain versions of JAWS (Job Access With Speech) with certain browsers:

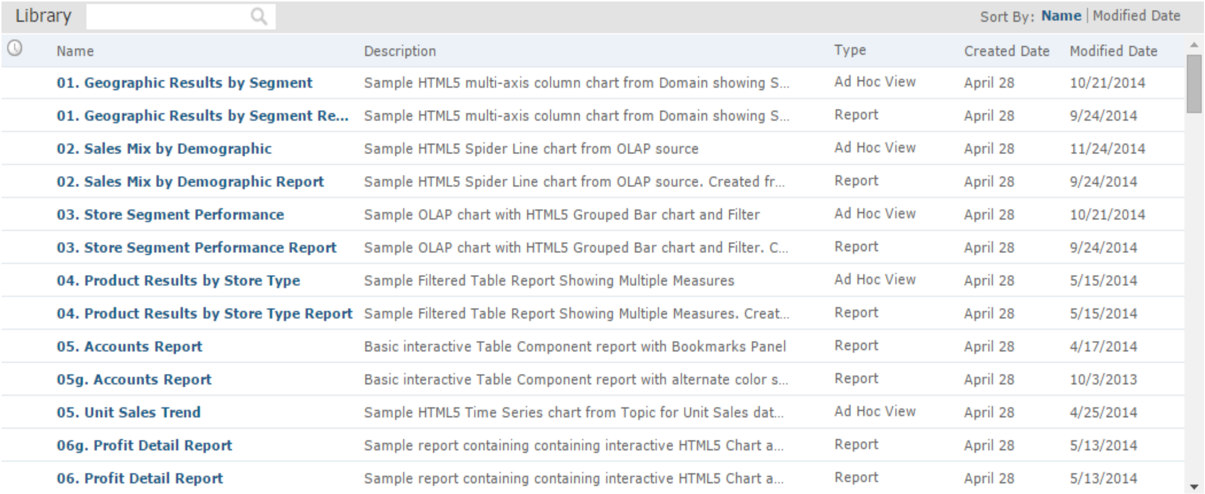
- Internet Explorer 8 with JAWS 14
- Internet Explorer 11 with JAWS 16

To further increase JasperReports Server's accessibility, we recommend that you enable the Easy Access theme, which increases color contrast and highlighting in the web UI. It can improve the user experience of those with visual impairment. For more information on themes, see the *JasperReports Server Administrator Guide*.

1.3 The Library Page

The Library page offers a more focused view of the repository objects. It contains only the Ad Hoc views, reports, and Dashboards that the currently logged-in user has rights to view and work with.

Click **Library** to view your Library list.



Name	Description	Type	Created Date	Modified Date
01. Geographic Results by Segment	Sample HTML5 multi-axis column chart from Domain showing S...	Ad Hoc View	April 28	10/21/2014
01. Geographic Results by Segment Re...	Sample HTML5 multi-axis column chart from Domain showing S...	Report	April 28	9/24/2014
02. Sales Mix by Demographic	Sample HTML5 Spider Line chart from OLAP source	Ad Hoc View	April 28	11/24/2014
02. Sales Mix by Demographic Report	Sample HTML5 Spider Line chart from OLAP source. Created fr...	Report	April 28	9/24/2014
03. Store Segment Performance	Sample OLAP chart with HTML5 Grouped Bar chart and Filter	Ad Hoc View	April 28	10/21/2014
03. Store Segment Performance Report	Sample OLAP chart with HTML5 Grouped Bar chart and Filter. C...	Report	April 28	9/24/2014
04. Product Results by Store Type	Sample Filtered Table Report Showing Multiple Measures	Ad Hoc View	April 28	5/15/2014
04. Product Results by Store Type Report	Sample Filtered Table Report Showing Multiple Measures. Creat...	Report	April 28	5/15/2014
05. Accounts Report	Basic interactive Table Component report with Bookmarks Panel	Report	April 28	4/17/2014
05g. Accounts Report	Basic interactive Table Component report with alternate color s...	Report	April 28	10/3/2013
05. Unit Sales Trend	Sample HTML5 Time Series chart from Topic for Unit Sales dat...	Ad Hoc View	April 28	4/25/2014
06g. Profit Detail Report	Sample report containing containing interactive HTML5 Chart a...	Report	April 28	5/13/2014
06. Profit Detail Report	Sample report containing containing interactive HTML5 Chart a...	Report	April 28	5/13/2014

Figure 1-4 Library Panel

From the Library page, you can:

- Run and schedule reports
- Open Ad Hoc views and generate reports from them
- Run and edit dashboards
- Run OLAP views

All of these functions are available by right-clicking the item you want to work with and selecting an action from the context menu.

1.3.1 Created vs. Modified Dates

The Library table has two columns that refer to when the repository items were created and last modified.

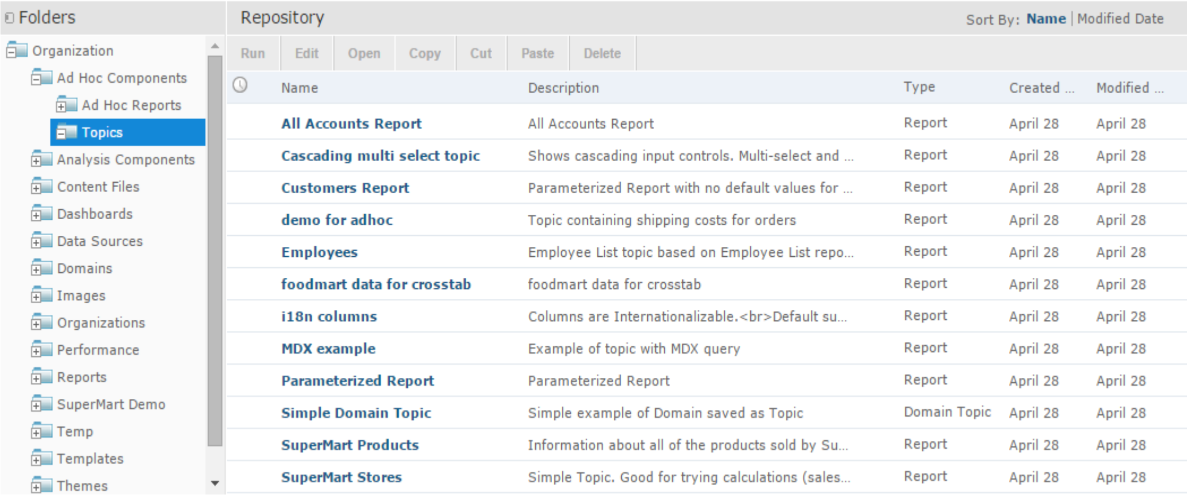
Generally, the created date will be earlier than the modified date. In some situations, however, the created date may be after the modified date. This can happen for one of two reasons:

- When an existing report (A) is modified, then subsequently copied into a new report (B). In the Library list, report B's created date is the day it was created, but its modified date reflects the last time report A was changed.
- An existing report is exported from one system and imported into another. In the Library list, the reports created date is the date it was imported into the new system, and the modified date is the date it was last modified in the original system.

1.4 Browsing the Repository

The repository is the server's internal storage for reports, analysis views, and related files. The repository is organized as a structure of folders containing resources, much like a file system. However, unlike a file system, the repository is stored as a private database that only JasperReports Server can access directly.

To browse the repository, select **View > Repository**. From the repository page, you access the reports, themes, and other files stored on the server. You can browse the repository contents that you have permission to view by expanding icons in **Folders**. Click a folder name to view its contents. In **Figure 1-5**, you'll see the Repository page.



Name	Description	Type	Created ...	Modified ...
All Accounts Report	All Accounts Report	Report	April 28	April 28
Cascading multi select topic	Shows cascading input controls. Multi-select and ...	Report	April 28	April 28
Customers Report	Parameterized Report with no default values for ...	Report	April 28	April 28
demo for adhoc	Topic containing shipping costs for orders	Report	April 28	April 28
Employees	Employee List topic based on Employee List repo...	Report	April 28	April 28
foodmart data for crosstab	foodmart data for crosstab	Report	April 28	April 28
i18n columns	Columns are Internationalizable. Default su...	Report	April 28	April 28
MDX example	Example of topic with MDX query	Report	April 28	April 28
Parameterized Report	Parameterized Report	Report	April 28	April 28
Simple Domain Topic	Simple example of Domain saved as Topic	Domain Topic	April 28	April 28
SuperMart Products	Information about all of the products sold by Su...	Report	April 28	April 28
SuperMart Stores	Simple Topic. Good for trying calculations (sales...	Report	April 28	April 28

Figure 1-5 Repository Folders Panel

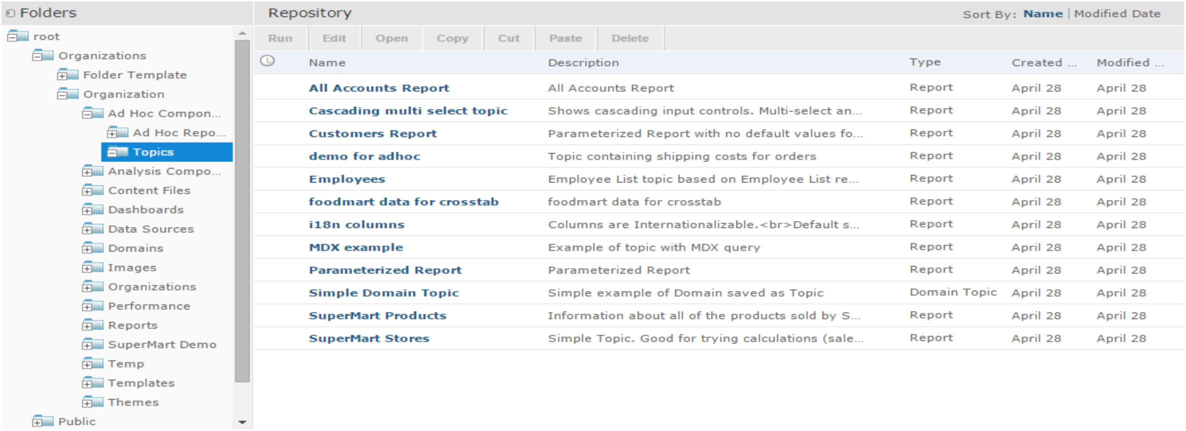
1.5 Searching the Repository

You can search the entire repository, subject to permissions, or narrow the search using filters. **Filters** restrict a search by name, who changed the resource, type of resource, date of the resource, and schedule.

1.5.1 Searching the Entire Repository

To search the repository, select **View > Search Results**. The search results page appears. Instead of only viewing resources by folder, use intuitive search criteria, such as who modified the resource and when, to find pinpoint resources.

On the search results page, use either the **Filters** panel or **Search** field to find resources. The search results page displays results of searches and filters.



Name	Description	Type	Created ...	Modified ...
All Accounts Report	All Accounts Report	Report	April 28	April 28
Cascading multi select topic	Shows cascading input controls. Multi-select an...	Report	April 28	April 28
Customers Report	Parameterized Report with no default values fo...	Report	April 28	April 28
demo for adhoc	Topic containing shipping costs for orders	Report	April 28	April 28
Employees	Employee List topic based on Employee List re...	Report	April 28	April 28
foodmart data for crosstab	foodmart data for crosstab	Report	April 28	April 28
i18n columns	Columns are Internationalizable. Default s...	Report	April 28	April 28
MDX example	Example of topic with MDX query	Report	April 28	April 28
Parameterized Report	Parameterized Report	Report	April 28	April 28
Simple Domain Topic	Simple example of Domain saved as Topic	Domain Topic	April 28	April 28
SuperMart Products	Information about all of the products sold by S...	Report	April 28	April 28
SuperMart Stores	Simple Topic. Good for trying calculations (sale...	Report	April 28	April 28

Figure 1-6 Search Results Page

To search all resources in the repository:

1. Select one of these filters: **All available**, **Modified by me**, or **Viewed by me.**
2. Click the ✖ icon in the search field to clear the search term if there is one.
3. Select **All types**, as shown in Figure 1-6.
4. Click 🔍.

   The search results appear listing files that your user account has permission to view. Click a resource in the list to view it or right-click a resource to see what functions are available from the context menu.

The server remembers your settings on the Search Results page, so the most commonly needed resources remain visible when you return to the page.

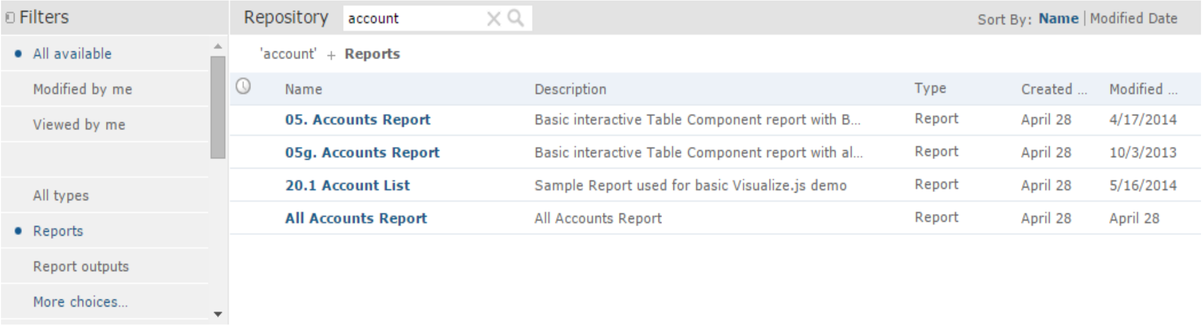
1.5.2 Filtering Search Results

If you enter a search term and click 🔍 at the top of any server page, the server doesn't use filters. The search uses these default settings:

- Include subfolders
- Start at the top-most folder visible to the user
- Search for reports, report outputs, OLAP views, or other resources
- Sort alphabetically by name

If you click **View > Search Results** and click 🔍 on the search results page, the server uses the filters you set in the **Filters** panel.

In Figure 1-7, you can see the results of a search for the term "account" using the filters All available and All types.



Name	Description	Type	Created ...	Modified ...
05. Accounts Report	Basic interactive Table Component report with B...	Report	April 28	4/17/2014
05g. Accounts Report	Basic interactive Table Component report with al...	Report	April 28	10/3/2013
20.1 Account List	Sample Report used for basic Visualize.js demo	Report	April 28	5/16/2014
All Accounts Report	All Accounts Report	Report	April 28	April 28

Figure 1-7 Search Field and Search Results

The search term you enter in the search field isn't cleared automatically. To clear the search term, click the ✖ icon in the search field.

You refine a search using filters. For example, filters can help you find your most recently viewed reports. You can set each filter independently. You can set the following types of filters:

- User
- Resource
- Access time
- Scheduled report

The user filter has the following settings:

Filter Setting	Description
All Available (default)	All resources.
Modified by me	Selects only resources that were last modified by the user who's logged in.
Viewed by me	Selects only resources that were run and viewed by the user who's logged in. This filter not only applies to visualization types, but also to resources that are included in reports such as images.

The resource type filter has the following settings:

Filter Setting	Description
All types (default)	All resources.
Reports	Displays only reports, both JRXML reports and Ad Hoc reports.
Report outputs	Displays only the output from reports that were scheduled or run in the background. Report output can be any of the supported export types, such as HTML and PDF.
Dashboards	Displays only dashboards.

Filter Setting	Description
OLAP views	Displays only analysis views (if you implement Jaspersoft OLAP).
Domains	Displays only Domains.
Data sources	Displays only data sources.

The access time filter has the following settings. All times are relative to the user's effective time zone:

Filter Setting	Description
Any time (default)	All resources.
Today	Resources viewed or modified since the previous midnight.
Yesterday	Resources viewed or modified during the previous day ending at midnight.
Past week	Resources viewed or modified during the past 7 days, including today.
Past month	Resources viewed or modified during the past 30 days, including today.

The scheduled report filter has the following settings:

Filter Setting	Description
Any schedule (default)	All resources.
Scheduled	Only reports that have scheduled jobs.
Scheduled by me	Only reports that have jobs scheduled by the currently logged in user.
Not scheduled	Only reports that don't have scheduled jobs and all other resource types.

Remember these do's and don'ts when searching for resources:

- Do use word fragments.
- Do search for the display name or part of the display name of a resource.
- Do search for words or fragments in the description of a resource.
- Do use multiple words.
- Don't search for folder names.
- Don't enter quotes around terms or symbols between terms.
- Don't worry about using upper- or lower-case letters in search terms.

1.6 Using Repository Resources

After finding a resource in the repository, naturally you want to do something with it. Options are:

- Click the name of a report to run and view it.
- Right-click the name of a resource to access other operations on the context menu, for example **Edit** or **Open in Designer**. Items appear on the context menu according to your permissions.

- Click anywhere in the row except the resource name to select a resource. Ctrl-click anywhere in the rows to select multiple resources. Use the context menu or buttons above the results list: **Run**, **Edit**, **Open**, **Copy**, **Cut** (move), or **Delete**. If the button is unavailable, the resource doesn't support the operation or you don't have permission for the operation. For example, the Open button is available when you select a dashboard or an Ad Hoc report if you have permission to write to it.

  You might also need permission to access the folder or dependent file, such as an image, of a resource. For example, to schedule a report, you need to have read/write/delete permission on the folder where server saves the report output. For more information about permissions, see the JasperReports Server Administrator Guide.

These two icons may appear in the **Repository** panel:

- ▸ indicates that the report has saved options for its input controls. Click the ▸ icon to list the saved options. For more information, see **"Running a Report with Input Controls or Filters" on page 69**.
- 🕓 indicates that the report is scheduled to run or is running in the background. Click this icon to view the list of jobs scheduled for the report. For more information, see **"Scheduling Reports and Dashboards" on page 75**.

1.7 Moving Folders

If you have read permission on folders and resources, you can copy and cut them from one folder and paste them to another if you have write permission on the destination folder. The server pastes all contents of the folder into the new location.

You can drag-and-drop the objects instead of using the paste menu item. Move folders one at a time. You can move other resources in batches.

Relocated objects inherit permissions from the destination folder, losing the permissions in place before the move. To change permissions on an object, set the permissions explicitly.

To move folders and resources by cutting and pasting:

1. Log into the server as a user who has these permissions:
   - Read permission on the folder or resource to move
   - Write permission on the destination folder

   For example, log in as joeuser (use the password, joeuser).
2. Click **View > Repository**.
3. In the **Folders** panel, right-click **Reports** > **Samples** and select **Add Folder**.
4. In the **Add Folder** dialog, enter a name, such as Financial Reports, and click **Add**.

   The Financial Reports folder appears as a subfolder of Samples and inherits Joe User's default permissions (read-write-delete) on the parent folder.

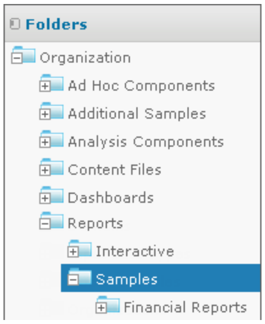


Figure 1-8 New Financial Reports Folder

5. The Financial Reports folder deserves a more prominent location. Move it up one level:
   a. In **Folders**, right-click Financial Reports, and select **Cut**.
   b. Right-click **Reports**, and select **Paste**.

      The Financial Reports folder now appears in **Reports** at the same level as **Samples**.

You can relocate a folder, subject to permissions, anywhere in the repository with one exception: The server doesn't support copying and pasting a folder to the same location. If the Paste command is disabled when you right-click a destination folder, you don't have write permission on the folder.

1.8 Sorting the Repository List

To change the order of the list of reports and other resources, use the **Sort By** controls:

- Click **Name** to sort alphabetically (A at the top). This is the default sort order.
- Click **Modified Date** to sort by the latest modified time and date (most recent at the top).

Chapter 2 Working with Dashboards

This section describes functionality that can be restricted by the software license for JasperReports Server. If you don’t see some of the options described in this section, your license may prohibit you from using them. To find out what you're licensed to use, or to upgrade your license, contact Jaspersoft.

A Jaspersoft dashboard displays several reports in a single, integrated view. A dashboard can include input controls for choosing the data displayed in one or more dashlets, and custom dashlets that point to URLs for other content. By combining different types of related content, you can create appealing, data-rich dashboards that quickly convey trends.

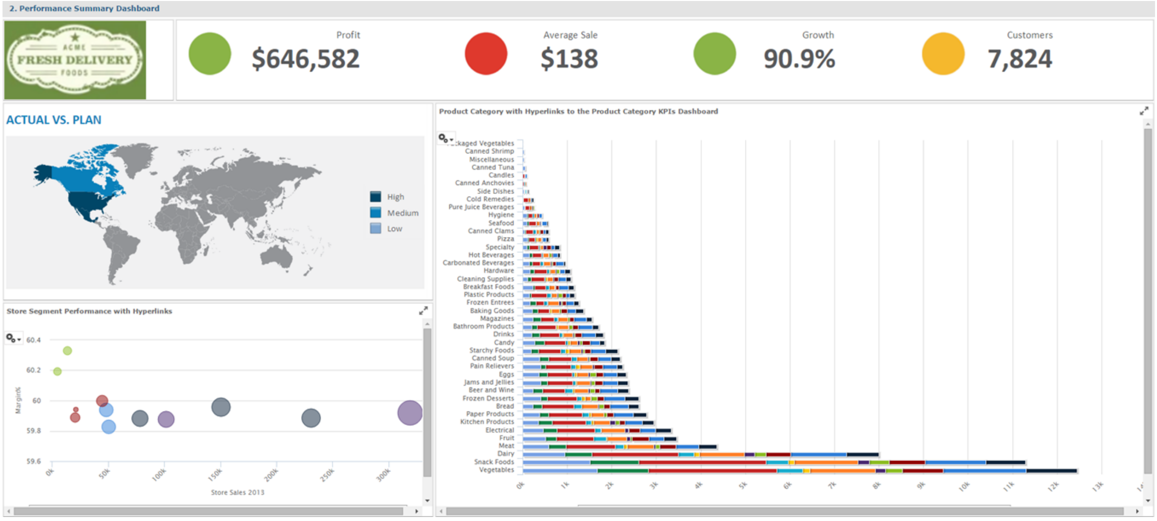


Figure 2-1 Sample dashboard

Dashboards created with JasperReports Server 6.0 and later use the Dashboard Designer, Jaspersoft's web UI for grouping all of these elements, along with creating new content from existing JasperReports Server data sources. The majority of the information in this chapter applies to dashboards created with JasperReports Server 6.0 and later.

If you're working with an earlier version of JasperReports Server, or dashboards created in an earlier version (regardless of your currently-installed version of JasperReports Server), you'll find information on working with dashboards at the end of this chapter, in section **2.8, “Viewing Legacy Jaspersoft Dashboards,” on page 50**

This chapter contains the following sections:

- **Overview of the Dashboard Designer**
- **Viewing a Dashboard**
- **Creating a Dashboard**
- **Editing a Dashboard**
- **Tips for Designing Dashboards**
- **Viewing Legacy Jaspersoft Dashboards**

2.1 Viewing a Dashboard

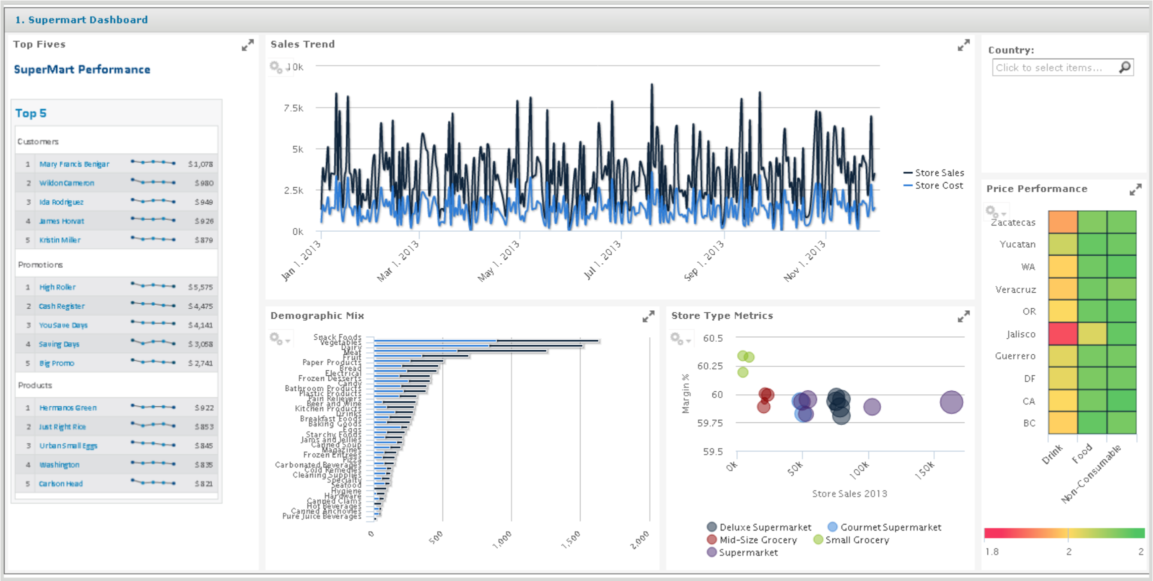
You can view a dashboard if you have the proper permissions. The default location for dashboards is the /Dashboards folder. The following procedure walks you through opening one of JasperReports Server's samples, the SuperMart dashboard.

To view the SuperMart dashboard:

1. Log in as user `demo`, using the password `demo`.

   Passwords are case-sensitive. You must use lowercase when you type `demo`.

   The SuperMart dashboard opens in the dashboard viewer. This dashboard includes five reports and an input control.



SuperMart Performance

Top 5

Customers		
1	Mary Francis Benigar	$1,078
2	Wildon Cameron	$980
3	Ida Rodriguez	$949
4	James Horvat	$926
5	Kristin Miller	$879
Promotions		
1	High Roller	$5,575
2	Cash Register	$4,475
3	You Save Days	$4,141
4	Saving Days	$3,058
5	Big Promo	$2,741
Products		
1	Hermanos Green	$922
2	Just Right Rice	$853
3	Urban Small Eggs	$845
4	Washington	$835
5	Carlson Head	$821

Figure 2-2 SuperMart Dashboard Example

2. In the **Country** input control dashlet, click the text entry box.
3. Select **USA**, then click **Close**to change the data displayed.

   The four reports with Country data update to display data for USA only. the Top 5 report, which does not contain Country data, does not change.

4. When done, click **View > Repository** to go to the repository page.

Keep these points in mind when viewing a dashboard that has input controls:

- An input control may appear as a text field, a drop-down, a check box, a multi-select list box, or a calendar icon.
- If one of the dashlets in a dashboard does not refer to an input control, that dashlet does not update when you change that input control's value. Only reports that use the input control reflect the change.
- If the ▼ button appears on the toolbar, then the dashboard was set up to display input controls as a pop-up window instead of a dashlet. Click the button to view the controls.

If a dashboard does not appear when you click on its name in the repository, it may already be open in another window or tab of your browser.

2.2 Overview of the Dashboard Designer

The Dashboard Designer is a web-based UI for embedding reports, Ad Hoc views, and other BI objects into a single, interactive space. You can compile dashboards that include pre-existing elements, such as reports and views, and create new charts, tables, and crosstabs from your data sources directly from the designer.

Each element on your dashboard is called a **Dashlet**. Dashlets have unique names and resource IDs, and editable properties that vary depending on the Dashlet type.

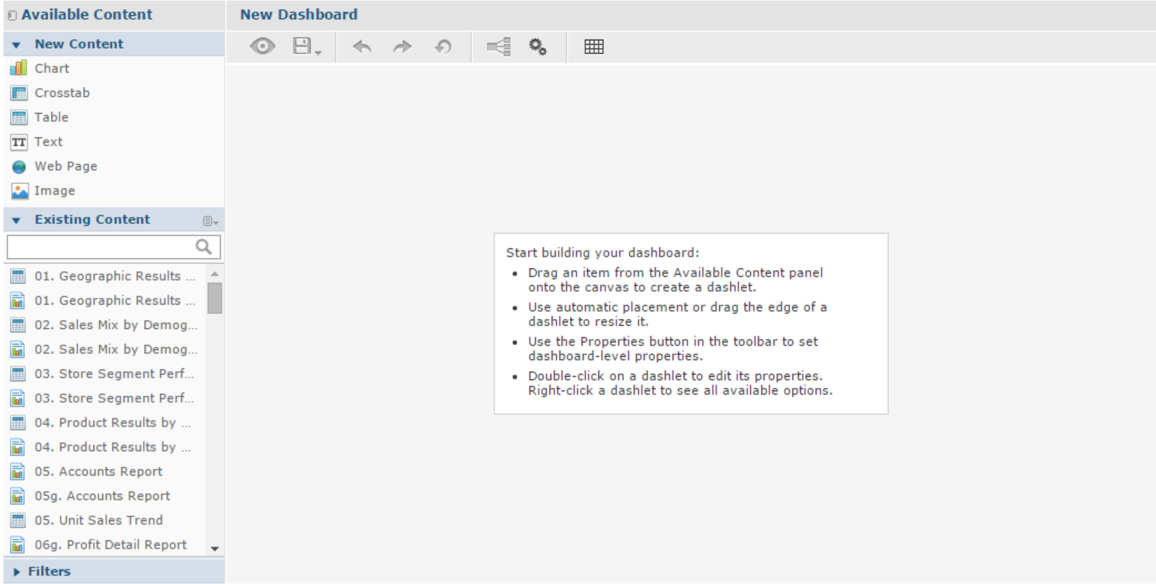
Your permissions to access the repository may limit the content you can add and the location where you can save the dashboard.

This section includes:

- **The Dashboard Designer Interface**
- **Dashlets and Dashboard Elements**
- **Previewing a Dashboard**
- **Dashboard Properties**
- **Dashlet Properties**
- **Parameter Mapping**

2.2.1 The Dashboard Designer Interface

The following figure shows the basic layout of the Dashboard Designer.


Available Content
New Content
Chart
Crosstab
Table
Text
Web Page
Image
Existing Content
01. Geographic Results ...
01. Geographic Results ...
02. Sales Mix by Demog...
02. Sales Mix by Demog...
03. Store Segment Perf...
03. Store Segment Perf...
04. Product Results by ...
04. Product Results by ...
05. Accounts Report
05g. Accounts Report
05. Unit Sales Trend
06g. Profit Detail Report
Filters
New Dashboard

Start building your dashboard:

- Drag an item from the Available Content panel onto the canvas to create a dashlet.
- Use automatic placement or drag the edge of a dashlet to resize it.
- Use the Properties button in the toolbar to set dashboard-level properties.
- Double-click on a dashlet to edit its properties. Right-click a dashlet to see all available options.

Figure 2-3 The Dashboard Designer UI

The Dashboard Designer UI includes the following panels:

- **Available Content**. From here, you can drag content onto the Dashboard Canvas. This panel includes the following sections:
  - **New Content**, which lists the content elements you can create for your dashboard.
  - **Existing Content**, which lists the Ad Hoc views and reports you can access from the Repository.
  - **Filters**, which lists all filters associated with any resource added to the dashboard.
- **Toolbar Buttons**. See table below for details.

Icon	Name	Description
👁	Preview	Click to display current dashboard as viewed by the end user.
💾	Save/Save As	Hover cursor over icon to open a menu of save options.
↶	Undo	Click to undo the most recent action.
↷	Redo	Click to redo the most recent undone action.
⟲	Undo All	Click to revert the dashboard to the most recently saved state.
⋲	Parameter Mapping	Click to open the Parameter Mapping dialog. See **Parameter Mapping** for more information.

Icon	Name	Description
⚙	Dashboard Properties	Click to open the Dashboard Properties box. See **Dashboard Properties** for more information.
▦	Show/Hide Grid	Click to display or hide a grid.
▼	Show/Hide Filter Dashlet Pop-up	Click to display or hide a filter pop-up window. This button only appears when you enable filter dashlet pop-ups in Dashboard Properties.

- **Dashboard Canvas**. This is where you create and edit your dashboard. It includes the following sections:
  - **Title Bar**, which displays the name of the dashboard (in the figure above, the name is "New Dashboard").
  - **Main Creation Area**, where you build your dashboard. Drag elements from the Available Content panel here to get started.

2.2.2 Dashlets and Dashboard Elements

Each element added to your dashboard is called a Dashlet.

To add a Dashlet to your dashboard, simply select a content element and drag it onto the dashboard canvas.

Dashlets can include the following elements, which you can access from the **Available Content** panel:

- **New Content**:
  - **Chart**: Allows you to create a chart within the Dashboard Designer.
  - **Crosstab**: Allows you to create a crosstab within the Dashboard Designer.
  - **Table**: Allows you to create a table within the Dashboard Designer.
  - **Text**: A free-form text entry field. Use free text items to add titles and instructional text to the dashboard.
  - **Web Page**. Any URL-addressable web content. The dashboard can point to web content.
  - **Image**: An image from the repository or that is accessible by a web address URL. For example, you might include a dashlet that displays your corporate logo. The logo's image file can either be in your repository or on the server of your corporate website.
- **Existing Content**: Reports and Ad Hoc views accessible to you.
- **Filters**: If a dashlet you include on the dashboard is designed to use input controls or filters, you can add that capability to the dashboard. The server maps input controls to one or more dashlets.
- **Title Bar**: You can enable a title bar in the Dashlet Properties dialog box. The title bar includes the following elements:
  - The Dashlet name, as entered in the Dashlet properties.
  - Dashlet toolbar, which can contain the following:

Icon	Name	Description
⤢	Maximize	Click to open the dashlet as a larger view.

Icon	Name	Description
⟳	Refresh	Click to refresh the dashlet.
📤	Export	Click to export the dashlet and save the output to your computer. See **"Exporting Dashboards and Dashlets" on page 48** for more information.

For more advanced functionality, you can access dialog boxes that let you edit the overall appearance of your dashboard, modify the functionality of your dashlets, and create mappings between your dashboard input controls and your dashlets.

2.2.3 Previewing a Dashboard

You can preview your dashboard in display mode to see how it will appear when an end-user views it.

To preview a dashboard:

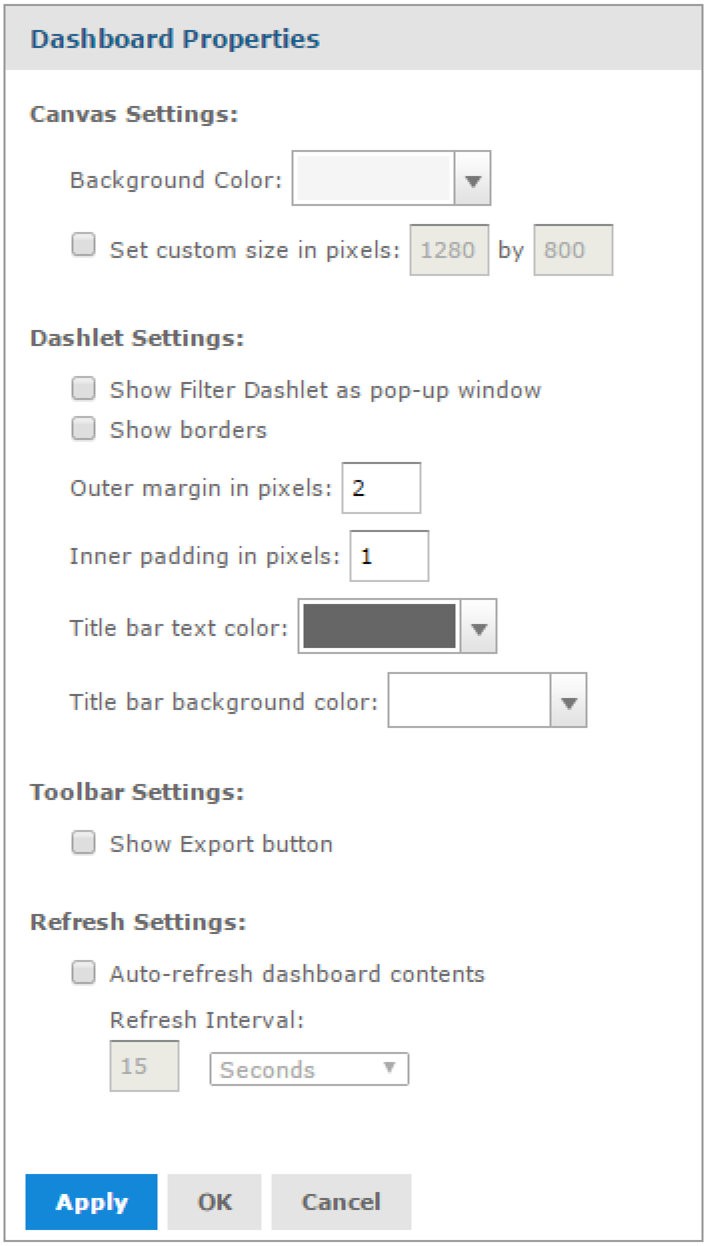
1. Click 👁. The dashboard opens in display mode.
2. To close the preview and return to the Dashboard Designer, click 👁.

2.2.4 Dashboard Properties

You can view and edit the basic appearance of all dashlets on your dashboard, and determine the refresh settings, through the Dashboard Properties.

To view and edit the Dashboard Properties:

1. On the dashboard toolbar, click ⚙ to open the Dashboard Properties window.



Dashboard Properties

Canvas Settings:

Background Color:

□ Set custom size in pixels: 1280 by 800

Dashlet Settings:

□ Show Filter Dashlet as pop-up window

□ Show borders

Outer margin in pixels: 2

Inner padding in pixels: 1

Title bar text color:

Title bar background color:

Toolbar Settings:

□ Show Export button

Refresh Settings:

□ Auto-refresh dashboard contents

Refresh Interval:

15 Seconds

Figure 2-4 Dashboard Properties

2. Edit the Canvas Settings, if needed:
   - **Background Color**: Select the color of the dashboard's background.
   - **Set custom size in pixels**: Select when you want the dashboard to be displayed at a specific width and height instead of dynamically resizing based on the browser window. Enter the desired width and height of the dashboard.
3. Edit the **Dashlet Settings**, if needed:
   - **Show filter dashlet as pop-up**: Select when you want the filter dashlet to appear as a pop-up window instead of a dashlet pinned on the dashboard.
   - **Show dashlet borders**: Select or deselect this to show or hide the thin lines around each dashlet.

- **Dashlet outer margin in pixels**: Enter the desired width, in pixels, of the margins between dashlets.
- **Dashlet inner padding in pixels**: Enter the desired width, in pixels, of the padding inside each dashlet.
- **Title bar text color**: Select the text color for the title bars of all of the dashboard's dashlets.
- **Title bar background color**: Select the background color for the title bars of all of the dashboard's dashlets.

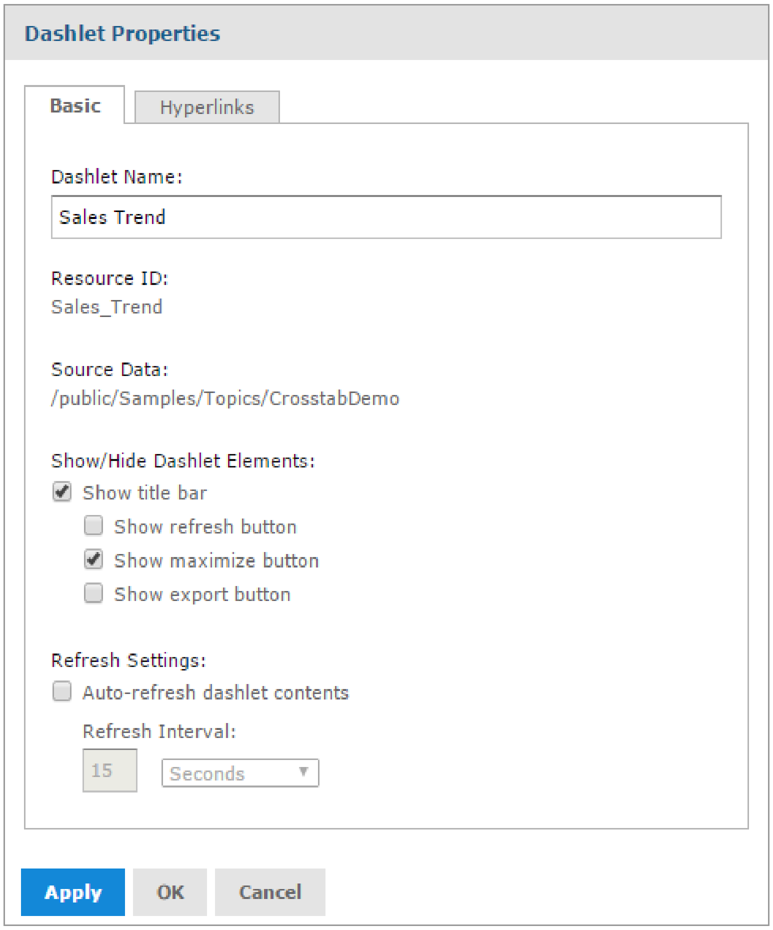
4. Edit the **Toolbar Settings**, if needed:
   - **Show Export Button**: Select or deselect to show or hide the Export button in the dashboard viewer.
5. Edit the **Refresh Settings**, if needed:
   - **Auto-refresh dashboard contents**: Select or deselect this to enable or disable automatic refresh for your content.
   - **Refresh Interval**: Enter the number of minutes or seconds between each content refresh, using the text entry and drop-down menu.

2.2.5 Dashlet Properties

You can view and edit basic information and appearance for each dashlet on your dashboard using the Dashlet properties. For some dashlets, you can also create parameters which you can then map to in Parameter Mapping. The available properties vary based on the type of dashlet you are working with.

To view and edit the Dashlet properties:

1. Right-click on the dashlet and select **Properties** to open the Dashlet Properties window. The **Hyperlinks** tab is available for Ad Hoc-based charts and charts created inside the dashboard, as well as text and image dashlets. A **Formatting** tab is available for text dashlets.



Dashlet Properties

Dashlet Name:
Sales Trend

Resource ID:
Sales_Trend

Source Data:
/public/Samples/Topics/CrosstabDemo

Show/Hide Dashlet Elements:
☑ Show title bar
□ Show refresh button
☑ Show maximize button
□ Show export button

Refresh Settings:
□ Auto-refresh dashlet contents
Refresh Interval:
15 Seconds

Figure 2-5 Dashlet Properties

2. Click **Apply** to view the changes, **OK** to accept the changes, and **Cancel** to discard the changes.

Properties for Dashlets containing charts, crosstabs, tables, reports, and Ad Hoc views include:

- **Dashlet Name**: Editable field for the displayed dashlet name.
- **Resource ID**: Non-editable ID taken from the original dashlet name.
- **Source Data**: Non-editable path of the source data.
- **Show/Hide Dashlet Elements**: Select or deselect to show or hide the title bar, which includes the dashlet name, refresh button, export button, and maximize button.
- **Scale to Fit**: Use the drop-down menu to determine how the element is scaled in the dashlet.
- **Refresh Settings**: Select or deselect to enable or disable auto-refresh, and use the text entry and drop-down menu to set the time between each content refresh.

Properties on the Hyperlink tab:

The **Hyperlinks** tab is available for Ad Hoc-based charts and charts created inside the dashboard, as well as text and image dashlets.

- **Enable hyperlinks**: Select or deselect to enable a hyperlink for the dashlet.
- **Action**: Select link behavior for this dashlet:

  - **Update dashboard**: Select this option to update dashlets linked to this one when a user clicks on a link in the dashlet.

For an example of linked dashlets, open the 2. Performance Summary Dashboard, and click on one of the bubbles in the Store Segment Performance with Hyperlinks chart.

  - **Open new page**: Select this to have the dashlet open a web page or report, dashboard, or an Ad Hoc view in the repository when a user clicks on the dashlet. Click **Browse** to select a resource from the repository.

    You can link directly to a web page using `http:` syntax:

    ```
    http://en.wikipedia.org/wiki/$P{Customer Country}
    ```

    A repository URL must begin with `repo:`, for example,

    ```
    repo:/public/Samples/Ad_Hoc_Views/05._Unit_Sales_Trend
    ```

    You can add a parameter to a hyperlink for a web page, dashboard, or report. Parameters are not available for Ad Hoc views. See **2.4, “Specifying Parameters in Dashlets,” on page 41** for more information about adding parameters to hyperlinks.

- **Available parameters**: If you want to add a parameter in the hyperlink, use a name in this list to have Parameter Mapping create the link to the parameter automatically.
- **Add New Parameter**: For text and image dashlets, you can define a parameter to include in the hyperlink. Click the **Add New Parameter** button and enter the parameter's name and value in the new table row. Click ✓ to save the new parameter.

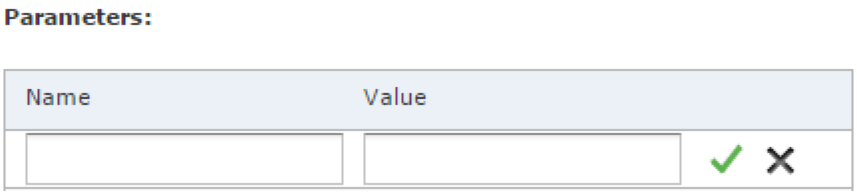


Figure 2-6 Adding a New Parameter to a Dashlet

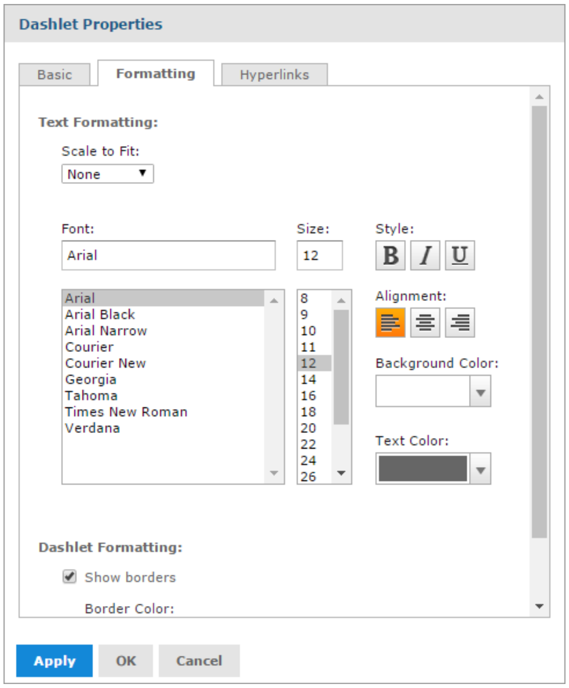
- **Map Parameters**: When you add a parameter, click this button to save and close the Dashlet Properties window and open Parameter Mapping. See **Parameter Mapping** for more information.

Properties for Dashlets containing text include:

- **Dashlet Name**: Editable field or the displayed dashlet name.
- **Resource ID**: Non-editable ID taken from the original dashlet name.
- **Text**: Editable field for the text displayed in the dashlet.

Properties on the Formatting tab for text dashlets include:

The **Formatting** tab is only available on text dashlets.



Dashlet Properties

Text Formatting:

Scale to Fit: None

Font: Arial

Size: 12

Dashlet Formatting:

☑ Show borders

Border Color:

Figure 2-7 Text Dashlet's Formatting Tab

- **Scale to Fit**: Use the drop-down menu to determine how the text is scaled in the dashlet. This overrides the specified font size.
- **Font**: Use the selection lists and buttons to set the font, font size, font style, alignment, and font color for the text displayed in the dashlet.
- **Show Borders**: Select or deselect to show or hide the dashlet's border. If you select to show the dashlet's border, you can choose the border's color.

Properties for dashlets containing a Web Page include:

- **Dashlet Name**: Editable field for the displayed dashlet name.
- **Resource ID**: Non-editable ID taken from the original dashlet name.
- **Web Page Address (URL)**: Editable field for the URL displayed in the dashlet.
- **Show/Hide Dashlet elements**: Select or deselect to show or hide the title bar, which includes the dashlet name, refresh button, and maximize button.
- **Show scroll bars**: Select or deselect to show or hide scroll bars.
- **Refresh Settings**: Select or deselect to enable or disable auto-refresh, and use the text entry and drop-down menu to set the time between each content refresh. This setting overrides refresh properties set at the dashboard level.

Properties for dashlets containing an image include:

- **Dashlet Name**: Editable field for the displayed dashlet name.
- **Resource ID**: Non-editable ID taken from the original dashlet name.
- **Web Address/Repository URI**: Editable field for the location of the image displayed in the dashlet. This can include parameters. Click **Browse** to select an image from the repository.

- **Show/Hide Dashlet elements**: Select or deselect to show or hide the title bar, which includes the dashlet name, refresh button, maximize button, and dashlet borders. If you select to show dashlet borders, you can choose the border color.
- **Scale to Fit**: Use the drop-down menu to determine how the image is scaled in the dashlet.

Properties for dashlets containing a filter include:

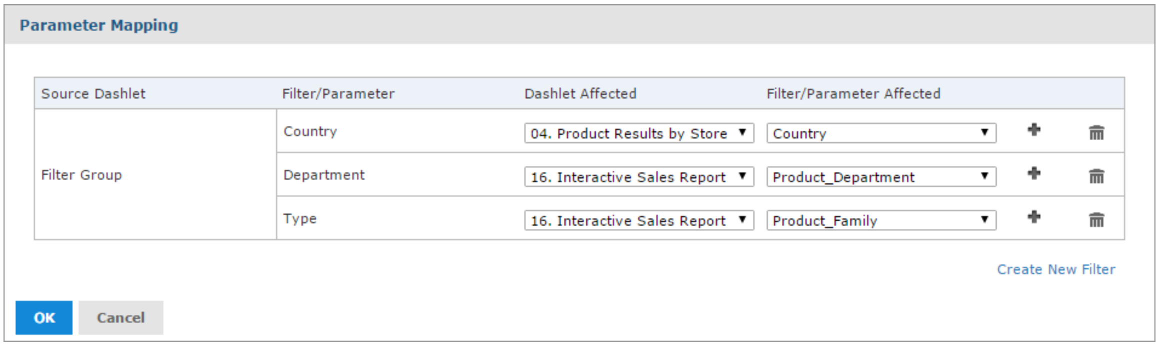
- **Dashlet Name**: Editable field for the displayed dashlet name.
- **Resource ID**: Non-editable ID taken from the original dashlet name.
- **Show/Hide Dashlet buttons**: Select or deselect to show or hide the Apply button or Reset button.
- **Position of dashlet buttons**: Use the drop-down menu to select bottom or right.

2.2.6 Parameter Mapping

Parameter Mapping helps you refine your filters by letting you specify which dashlets and which parameters are affected by a filter.

To open Parameter Mapping:

1. Open a dashboard with filters, such as the Sales Dashboard created in **Creating a Dashboard**, in the Dashboard Designer.
2. Click to open Parameter Mapping.



Parameter Mapping

Source Dashlet	Filter/Parameter	Dashlet Affected	Filter/Parameter Affected		
Filter Group	Country	04. Product Results by Store ▼	Country ▼	✚	🗑
	Department	16. Interactive Sales Report ▼	Product_Department ▼	✚	🗑
	Type	16. Interactive Sales Report ▼	Product_Family ▼	✚	🗑

Figure 2-8 Parameter Mapping for the Sales Dashboard

Parameter Mapping displays the filter-to-dashlet mapping, and includes the following columns and buttons:

- **Source Dashlet**, the name of the dashlet where the filter originates. Can also display Filter Group (multiple filters in a single dashlet) or Manually Created Filter (filter created using Parameter Mapping, as described below).
- **Filter/Parameter**, the name of the filter.
- **Dashlet Affected**, with a dropdown menu including all dashlets that can be affected by that filter.
- **Filter/Parameter Affected**, with a dropdown menu including all parameters associated with the selected dashlet in the Dashlet Affected column.
- **Add** button ✚, used to add additional dashlet/parameter combinations to a filter.
- **Delete** button 🗑, used to delete a dashlet/parameter combination.

From Parameter Mapping, you can add, delete, or edit an existing dashlet/parameter combination, and create a new filter to add to the dashboard.

To add a filter using Parameter Mapping:

1. Open a dashboard with filters, and open Parameter Mapping as described above.
2. In the filter row you want to add the new filter to, click ✚. A row containing new affected dashlet and filter/parameter dropdown menus appears.
3. Using these new line-items, select the dashlet and parameter combination you want to apply to the dashboard.
4. Click **OK** to apply and save or **Cancel** to discard your changes.

To delete a filter using Parameter Mapping:

1. Open Parameter Mapping.
2. In the filter row you want to delete, click 🗑. The filter row disappears from Parameter Mapping.

To create a new filter via Parameter Mapping:

1. Open Parameter Mapping.
2. Click **Create New Filter**. A new row is added to the manager.
3. In the **Filter** column, enter a name for the new filter. Click outside the text box to apply the name.
4. Click ✚ and select the dashlet and parameter combination you want to apply to the new filter.
5. Click **OK** to apply and save, or **Cancel** to discard the new filter.

To delete a newly-created filter:

1. In Parameter Mapping, click 🗑 in the filter row you want to delete. The Dashlet Affected and Filter/Parameter affected dropdown menus disappear.
2. Click 🗑 again in the row you want to delete. The Filter row disappears.
3. Click **OK** to apply and save, or **Cancel** to discard the new filter.

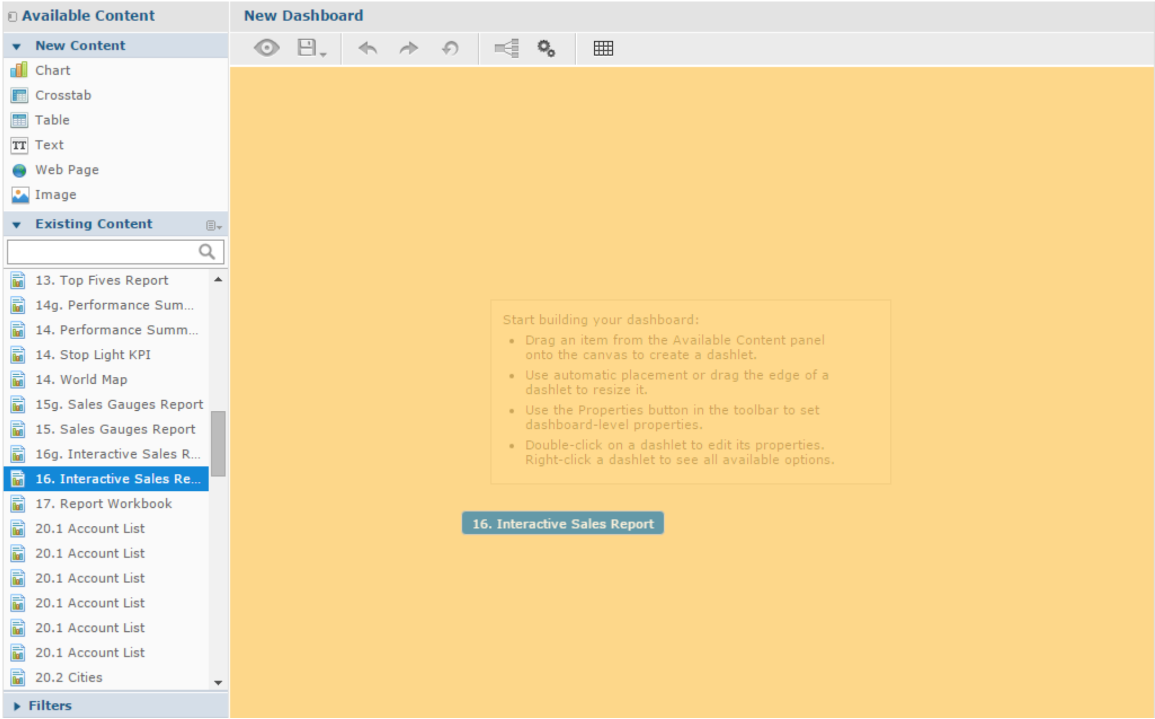
2.3 Creating a Dashboard

This section describes the creation of a simple dashboard.

To add a report or Ad Hoc view to a dashboard, you must have permission to view those elements.

To create a simple dashboard:

1. Click **Create > Dashboard**. The Dashboard Designer appears, displaying the list of available content and the canvas.
2. In the **Existing Content** section of the **Available Content** panel, find report 16. Interactive Sales Report.
3. Click and drag the report onto the Dashboard Canvas.



Start building your dashboard:

- Drag an item from the Available Content panel onto the canvas to create a dashlet.
- Use automatic placement or drag the edge of a dashlet to resize it.
- Use the Properties button in the toolbar to set dashboard-level properties.
- Double-click on a dashlet to edit its properties. Right-click a dashlet to see all available options.

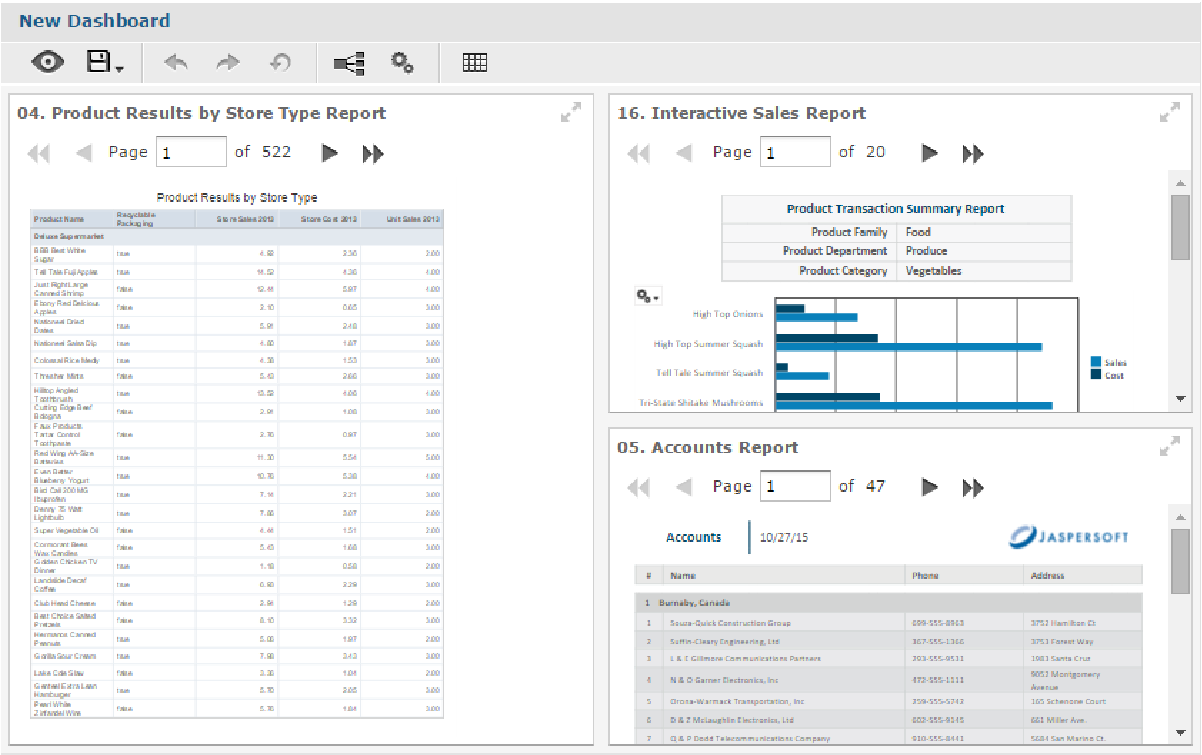
Figure 2-9 Dragging report onto Dashboard Canvas

4. In **Available Content**, find 04. Product Results by Store Type Report.
5. Click and drag the report onto the canvas, until the left half of the Interactive Sales Report turns orange.

   The Product Results by Store Type Report appears on the canvas, next to the Interactive Sales Report. Both report dashlets are sized to fit side-by-side on the canvas.
6. Right-click the Product Results by Store Type Report and click **Properties**.
7. Use the **Scale to Fit** dropdown to select **Dashlet**.
8. Click **OK**. The resized Product Results by Store Type Report now displays the entire crosstab in the dashlet.
9. Click and drag the report 05. Accounts Report onto the canvas, until the lower half of the Interactive Sales Report turns orange.

   The Accounts Report is placed under the Interactive Sales Report, and both dashlets are resized to fit on the right half of the canvas.
10. In the Dashboard Designer toolbar, click ⚙ to open the Dashboard Properties dialog.
11. Deselect **Show dashlet borders** and click **OK**.

    In **Figure 2-10** you can see how the dashboard canvas looks at this point.



New Dashboard

04. Product Results by Store Type Report

Page 1 of 522

Product Results by Store Type

Product Name	Recyclable Packaging	Store Sales 2013	Store Cost 2013	Unit Sales 2013
Deluxe Supermarket				
BBB Best White Sugar	true	4.92	2.36	2.00
Tell Tale Fuji Apples	true	14.52	4.36	4.00
Just Right Large Canned Shrimp	false	12.44	5.97	4.00
Ebony Red Delicious Apples	false	2.10	0.65	3.00
Nationeel Dried Dates	true	5.91	2.48	3.00
Nationeel Salsa Dip	true	4.80	1.87	3.00
Colossal Rice Medly	true	4.38	1.53	3.00
Thresher Mints	false	5.43	2.66	3.00
Hilltop Angled Toothbrush	true	10.52	4.06	4.00
Cutting Edge Beef Bologna	false	2.91	1.08	3.00
Faux Products Tartar Control Toothpaste	false	2.76	0.87	3.00
Red Wing AA-Size Batteries	true	11.30	5.54	5.00
Even Better Blueberry Yogurt	true	10.76	5.38	4.00
Bird Call 200 MG Ibuprofen	true	7.14	2.21	3.00
Denny 75 Watt Lightbulb	true	7.68	3.07	2.00
Super Vegetable Oil	false	4.44	1.51	2.00
Cormorant Bees Wax Candles	false	5.43	1.68	3.00
Golden Chicken TV Dinner	true	1.18	0.58	2.00
Landslide Decaf Coffee	true	6.80	2.29	3.00
Club Head Cheese	false	2.96	1.28	2.00
Best Choice Salted Pretzels	false	8.10	3.32	3.00
Hermanos Canned Peanuts	true	5.06	1.97	2.00
Gorilla Sour Cream	true	7.98	3.43	3.00
Lake Cole Slaw	false	3.36	1.04	2.00
Genteel Extra Lean Hamburger	true	5.70	2.05	3.00
Pearl White Zinfandel Wine	false	5.76	1.84	3.00

16. Interactive Sales Report

Page 1 of 20

Product Transaction Summary Report	
Product Family	Food
Product Department	Produce
Product Category	Vegetables

05. Accounts Report

Page 1 of 47

Accounts | 10/27/15

#	Name	Phone	Address
1	Burnaby, Canada		
1	Souza-Quick Construction Group	099-555-8963	3752 Hamilton Ct
2	Saffin-Cleary Engineering, Ltd	367-555-1366	3753 Forest Way
3	L & E Gillmore Communications Partners	393-555-9511	1983 Santa Cruz
4	N & O Garner Electronics, Inc	472-555-1111	9052 Montgomery Avenue
5	Orona-Warmack Transportation, Inc	259-555-5742	105 Schenone Court
6	D & Z Mclaughlin Electronics, Ltd	602-555-9145	661 Miller Ave.
7	Q & P Dodd Telecommunications Company	910-555-8441	5684 San Marino Ct.

Figure 2-10 Simple Dashboard Canvas with three reports

12. Click 👁 to preview the dashboard.

    The end user view of the dashboard appears.

13. Click 👁 to return to the designer
14. Click 💾▾, then select **Save Dashboard**.
15. In the Save As window, change the default name, New Dashboard to `Sales Dashboard` and locate a folder, such as the /Dashboards folder.
16. Click **Save**.

2.3.1 Adding New Content

In addition to pre-existing reports and Ad Hoc views, you can create content for your dashboard directly from the Dashboard Designer, including:

- Charts
- Crosstabs
- Tables
- Text
- Web page links
- Images

2.3.1.1 Adding Charts, Crosstabs, and Tables

The Dashboard Designer includes an on-board Ad Hoc Editor, which allows you to create charts, crosstabs, and tables for your dashboard without leaving the designer environment.

Any chart, crosstab, or table you create within the Dashboard Designer is available only on the current dashboard; otherwise, they function like standard Ad Hoc Editor-created versions of these elements. They are saved as an Ad Hoc view and placed in a dashlet on your dashboard.

To add a new chart, crosstab, or table to your dashboard:

1. In the **New Content** section of the **Available Content** panel, click and drag the type of element you want to add to your dashboard (Chart, Crosstab, or Table) onto the dashboard canvas.

   The designer's Ad Hoc Editor opens, and the **Select Data** dialog appears.
2. Browse to or search for the data source you want to use.
   - Click ⊢ for a tree view of the files.
   - Click ☰ for a list view of the files.
   - Use the text search field to locate a specific data source.
3. Depending on your selected data source, the remaining steps may vary. Follow the displayed instructions. for more information about this process, see **4.1.1, “Ad Hoc Sources: Topics, Domains, and OLAP Connections,” on page 90**.
4. When you complete the data source selection process, click **OK**. The Ad Hoc Editor opens.

   The on-board Ad Hoc Editor works just like the standard editor. For information on working with the editor, see **Chapter 4, “Working with the Ad Hoc Editor,” on page 89**.
5. When you finish creating your view, click 💾 to save.
6. In the **Save to Dashboard** dialog, enter a dashlet name and click **Save**. The dashlet is added to your dashboard.

2.3.1.2 Adding Text

You can add a text field dashlet for titles and instructional text.

To add a text dashlet:

1. In the **New Content** section of the **Available Content** panel, click and drag the **Text** item onto your dashboard. The Dashlet Text window opens.
2. Enter the text you want to appear on your dashboard.
3. Click **OK**. The dashlet is added to your dashboard.

   Edit the dashlet name and font appearance by opening the Properties dialog. See **“Dashlet Properties” on page 30** for more information.

2.3.1.3 Adding a Web Page

You can add a dashlet to display a web page on your dashboard.

To add a web page dashlet:

1. In the **New Content** section of the **Available Content** panel, click and drag the **Web Page** item onto your dashboard. The Dashlet URL window opens.
2. Enter the URL you want to appear on your dashboard.

3. Click **OK**. The dashlet is added to your dashboard.

   Edit the dashlet name by opening the Properties dialog. See **"Dashlet Properties" on page 30** for more information.

2.3.1.4 Adding an Image

You can add a dashlet to display an image, such as a corporate logo, on your dashboard.

To add an image dashlet:

1. In the **New Content** section of the **Available Content** panel, click and drag the **Image** item onto your dashboard. The Dashlet URI window opens.
2. Enter the URI for the image you want to appear on your dashboard. Use the `repo:` syntax for images in your repository.
3. Click **OK**. The dashlet is added to your dashboard.

   Edit the dashlet name by opening the Properties dialog. See **"Dashlet Properties" on page 30** for more information.

2.3.2 Adding Controls to a Dashboard

The Interactive Sales Report was designed to be run with input controls. When you add a report that has input controls to a dashboard, the controls don't appear until you explicitly add them, one-by-one. The controls can either be added directly to the dashboard as a dashlet or accessed from the toolbar as a pop-up window. When the report runs, dashboard users provide input using the control. Data based on the user input appears in the dashboard. For example, using the input control, you select Mexico. The report on the dashboard shows orders from Mexican companies.

To add controls as a dashlet:

1. If the Sales Dashboard created in **Creating a Dashboard** is not open, locate the /Dashboards folder in the repository. Right-click the dashboard name and select **Open in Designer** from the context menu.

   The Sales Dashboard appears in the designer, as shown in **"Simple Dashboard Canvas with three reports"**, and the input controls available appear in the **Filters** section of the **Available Content** panel.
2. In the **Filters** section, expand the 16. Interactive Sales Report folder.

   The input controls associated with the Interactive Sales Report appear.
3. Drag the Country input control onto the canvas, and place it above the Product Results by Store Type dashlet.

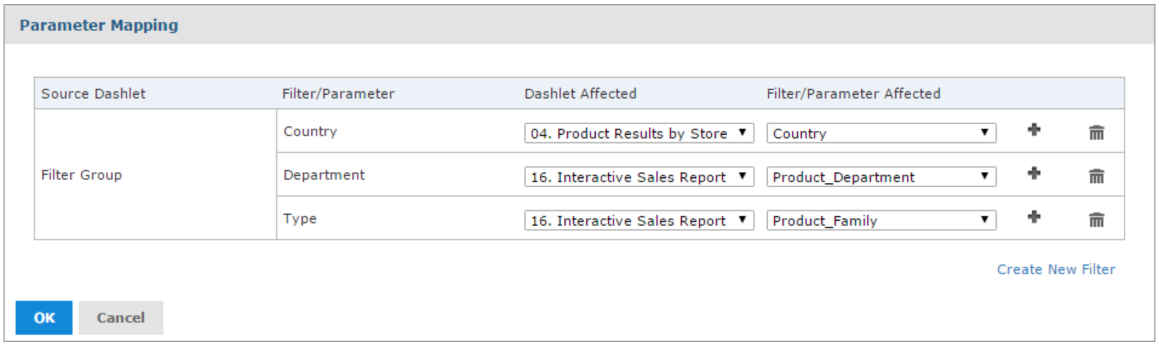
   The Country input control and its label appear above the Product Results by Store type report on the canvas.
4. Drag the Product Family and Product Department controls onto the Country input control dashlet. These input controls are added to the same dashlet. Resize the dashlets as needed to view all of the input controls.
5. Click 💾▾ and select **Save Dashboard**, then click 👁 to preview the dashboard.
6. Click in the **Country** text box to display the available countries. In this input control, you have the following options:
   - The three countries: **Canada**, **Mexico**, and **USA**.
   - ✓ **All**, which selects all available values in the input control.
   - ✕ **None**, which deselects all available values in the input control.
   - ⇄ **Invert**, which deselects any selected values, and selects the unselected values.

7. Use the options to select **Mexico** from the values list, and click **Apply** at the bottom of the dashlet. The data displayed in the Interactive Sales Report changes, but is not updated in the other reports, as they do not have an input control named Country.
8. Click 👁 to return to the Dashboard Designer.

You can also change the labels, or display names, of individual input controls and filters within a dashlet.

To rename an input control or filter:

1. With the Sales Dashboard open in the Dashboard Designer, right-click Product Family in the input control dashlet, and select **Properties**. The **Filter Properties** dialog opens.
2. Change the **Filter Label** from `Product Family` to `Type`, and click **OK**.
3. Right-click the Product Department input control and select **Properties**. Change the **Product Department** filter label to `Department`, and click **OK**. The input control labels are updated.



Parameter Mapping

Source Dashlet	Filter/Parameter	Dashlet Affected	Filter/Parameter Affected		
Filter Group	Country	04. Product Results by Store ▼	Country ▼	✚	🗑
	Department	16. Interactive Sales Report ▼	Product_Department ▼	✚	🗑
	Type	16. Interactive Sales Report ▼	Product_Family ▼	✚	🗑

Figure 2-11 Parameter Mapping for the Sales Dashboard

To add controls as a pop-up window:

1. With the Sales Dashboard open in the Dashboard Designer, click ⚙ to open Dashboard Properties.
2. Select **Show filter dashlet as pop-up** and click **OK**.

   The input controls dashlet disappears from the dashboard and ▼ appears on the toolbar. This button is also displayed when you're viewing the dashboard.
3. Click 💾▾ and select **Save Dashboard**, then click 👁 to preview the dashboard.
4. Click ▼ to view the filter pop-up window.

   Like the dashlet, selecting the options in the pop-up changes the data displayed in the Interactive Sales Report.
5. Click ▼ to close the filter pop-up window.
6. Click 👁 to return to the Dashboard Designer.

2.3.3 Refining a Dashboard's Layout

After completing the layout, refine the look and feel of the dashboard.

To refine the dashboard's layout:

1. If it isn't open, locate the Sales Dashboard you created in **Creating a Dashboard**, typically in the /Dashboards folder.
2. Right-click the dashboard name, select **Open in Designer** from the context menu
3. When the dashboard opens in the designer, click **Preview**.

   The end user's view of the dashboard appears.
4. In the **Country** field, select a new value.

   The reports do not update because the dashboard includes the **Apply** button.
5. Return to the Dashboard Designer and on the Country filter dashlet, right-click and select **Properties**.
6. Deselect **Show Apply button**, and select **Show Reset button**.
7. Click **Apply**.

   The dashlet's **Apply** button disappears, and the **Reset** button appears.
8. Reposition the **Reset** button to the right side of the dashlet using the Position of dashlet buttons drop-down menu. Click **OK**.
9. Click **Preview**. The end user view of the dashboard appears.
10. Change the value in the Country input control. The dashboard reflects the change immediately.
11. Return to the Dashboard Designer and click **Save > Save Dashboard**. The dashboard is saved to the repository.

2.4 Specifying Parameters in Dashlets

You can add parameters to dashlet names, text dashlets, web page dashlets, image dashlets, and hyperlinks in chart dashlets. When you add a parameter, you must map a filter in the dashboard to the parameter. See **Mapping parameters in Parameter Mapping** for more information.

Simple parameters:

For text dashlets, web links, and dashboard names, use the syntax `$P{parameter_name}` to directly pass the parameter value to the web page. Examples:

- In a text dashlet, enter the parameter in the Text box:

  ```
  Product Family: $P{Product Family}
  ```
- In a web page dashlet or a web link in a chart dashlet hyperlink, enter the parameter in the **Web Page Address (URL)** text box:

  ```
  http://en.wikipedia.org/wiki/$P{Customer Country}
  ```
- In a repository or web URI for an image dashlet, enter the parameter, enter the parameter in the **Web Address/Repository URI** text box:

  ```
  repo:/public/Samples/Reports/$P{Customer Country}/flag.png
  ```
- Dashlet name: Enter the parameter in the **Dashlet Name** box:

  ```
  Unit Sales for $P{Product Family}
  ```

For repository hyperlinks in a chart dashlet, use the syntax `?filter_in_target=$P{dashlet_param}`. For example, the following link sets the c_country_1 input control in the 05. Unit Sales Trend report to the value of a Store Country parameter in the dashlet.

```
repo:/public/Samples/Reports/05._Unit_Sales_Trend?c_country_1=$P{Store Country}
```

To find the correct name for a filter in a target resource, open the Dashboard Designer in a new window, and add the resource to the canvas. Then expand the resource's folder in the Filters panel and hover over the filter whose name you want to see.

Parameters in multi-valued input controls:

For multi-valued input controls, you can define a separator for the input control values:

- For text dashlets, web links, and dashboard names, use the syntax $P{parameter_name ? "separator"}. The following example in a text box displays results like **Product Family: Drink + Food:**

  ```
  Product Family: $P{Product Family ? " + "}
  ```

- For repository links in a chart or image dashlet, use the syntax `?$P{"filter_in_target=", dashlet_param, ? "&"}`. For example, the following link allows multi-select with the c_country_1 input control in the 05. Unit Sales Trend report:

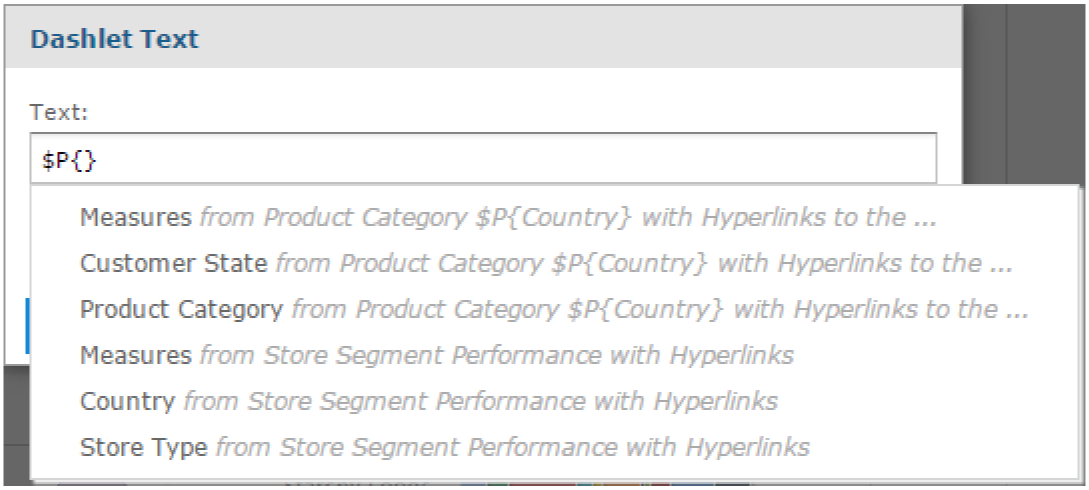
  ```
  repo:/public/Samples/Reports/05._Unit_Sales_Trend?$P{"c_country_1=", Store
  Country, "&"}
  ```

When `Mexico` and `USA` are selected, this expands to the following link:

```
repo:/public/Samples/Reports/05._Unit_Sales_Trend?c_country_1=Mexico&c_country_1=USA
```

2.4.0.1 Auto-complete parameters:

When you type the `$` symbol into a dashlet's text box, a drop-down list of available parameters is displayed. Select a parameter from the list or continue typing to narrow down the list of parameters. The auto-complete parameters feature is available for text fields, image link, and web links, but it is not available for the dashlet name field.



Dashlet Text

Text: $P{}

Measures from Product Category $P{Country} with Hyperlinks to the ...
Customer State from Product Category $P{Country} with Hyperlinks to the ...
Product Category from Product Category $P{Country} with Hyperlinks to the ...
Measures from Store Segment Performance with Hyperlinks
Country from Store Segment Performance with Hyperlinks
Store Type from Store Segment Performance with Hyperlinks

Figure 2-12 Text dashlet showing available parameters

2.4.0.2 Mapping parameters in Parameter Mapping

After you have created a parameter, create a mapping for it as follows:

Mapping parameters in dashboard names and text, image, or web page dashlets:

1. In the Dashboard Designer, click to open Parameter Mapping.

2. In the filter row where you want to add the new filter, click ✚. A row containing new affected dashlet and filter/parameter dropdown menus appears.
3. Using these new line-items, select the dashlet where you added a parameter, then select the parameter.
4. Click **OK** to apply and save or **Cancel** to discard your changes.

Mapping parameters in chart hyperlinks:

To have JasperReports Server create a parameter mapping automatically:

1. In the Dashlet Properties dialog box for your chart, use one of the names in the Available parameters list as the name of your parameter.
2. Click **OK** to close the dialog box and create the mapping.

To create a mapping in Parameter Mapping:

1. In the Dashlet Properties dialog box for your chart, click **Create Links in Parameter Mapping...** to open Parameter Mapping.
2. In the filter row where you want to add the new filter, click ✚. A row containing new affected dashlet and filter/parameter dropdown menus appears.
3. Using these new line-items, select the dashlet where you added a parameter, then select the parameter.
4. Click **OK** to apply and save or **Cancel** to discard your changes.

2.4.1 Creating a Web Page Dashlet

When working with Web Page dashlets in the Dashboard Designer, you can include a parameter reference in the dashlet's URL. The parameter references an existing filter in another dashlet, and uses that filter to display a specific web page relevant to the filtered information.

The following example takes you through the steps for creating a simple dashboard in the Dashboard Designer, adding a sample chart with geographic filters, and creating a Web Page dashlet that displays a Wikipedia page for the country the sample chart is filtered on.

First, create the simple dashboard:

1. Log in to JasperReports Server as superuser.
2. Click **Create > Dashboard**.

   The Dashboard Designer appears, displaying the list of available content and the canvas.
3. In the **Existing Content** section of the Available Content panel, find report 06. Sales Mix by Gender.
4. Click and drag the report onto the Dashboard Canvas. Note that the **Filters** section now includes the Sales Mix by Gender chart.

Next, add the filter input control to the dashboard:

1. In the **Filters** section, expand the 06. Sales Mix by Gender folder.

   The input controls associated with the Sales Mix by Gender Report appear.
2. Drag the Customer Country input control onto the canvas.

Now, add the Web Page dashlet with the parameter reference and add it to the filter:

1. In the **New Content** section of the **Available Content** panel, click and drag the **Web Page** item onto your dashboard.

   The Dashlet URL window opens.

2. Enter the following URL:

   `http://en.wikipedia.org/wiki/$P{Customer Country}`
3. Click **OK**.
4. Right-click the dashlet and select **Properties**.
5. Change the Dashlet Name to **Wiki**.
6. Click ⊶ to open Parameter Mapping.
7. In the **Customer Country** filter, click ✚.

   A row containing the new affected dashlet and filter/parameter dropdown menus appears.
8. In the Dashlet Affected column, select **wiki** from the new Select dashlet... dropdown menu.
9. In the Filter/Parameter Affected column, select Customer Country from the new Select parameter... dropdown menu.
10. Click **OK**.

Finally, preview the new dashboard functionality:

1. Click 👁 to preview the dashboard.
2. Click in the **Customer Country** text box to display the available countries.
3. Select **Mexico** from the values list, and click **Apply** at the bottom of the dashlet. The data in the Sales Mix by Gender report is updated to display only information about Mexico, and the wiki dashlet displays the Wikipedia page for Mexico.

   In Figure 2-13 you can see how the dashboard canvas looks at this point:

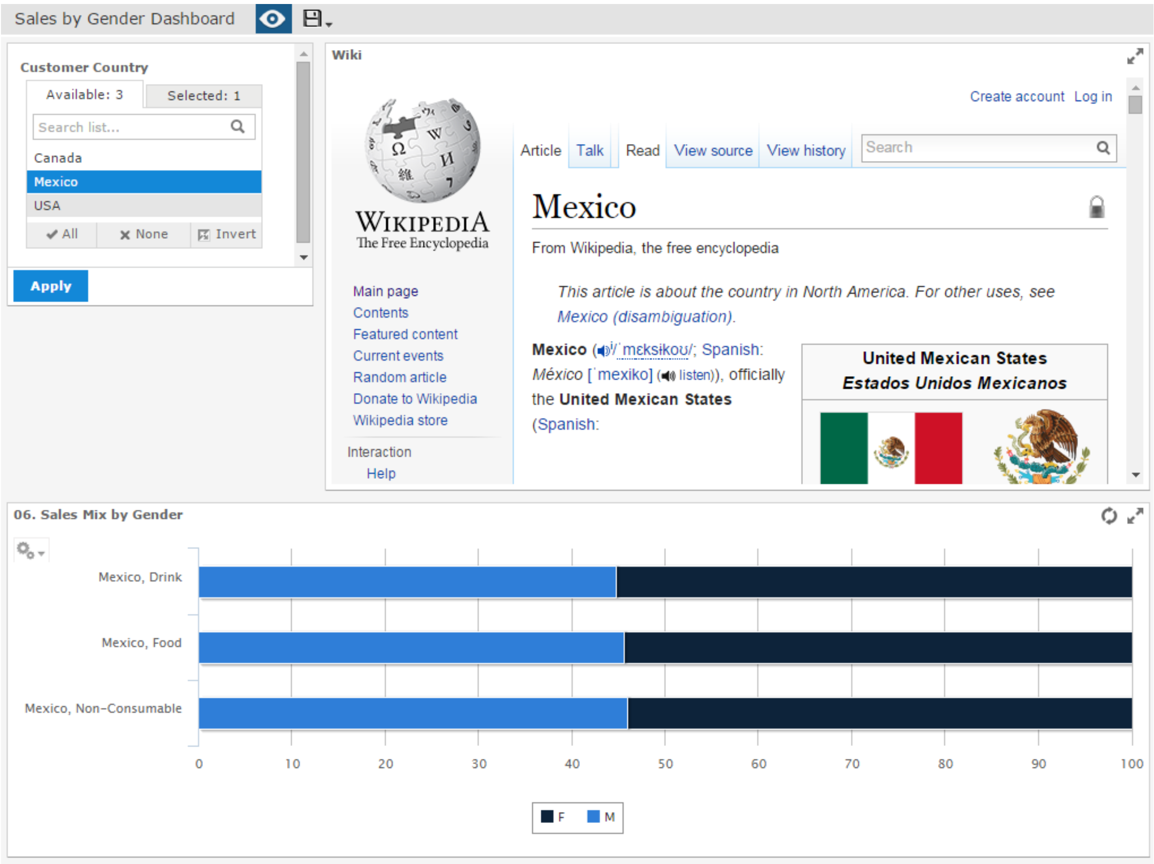


Figure 2-13 Dashboard with web page parameters

4. Click 👁 to return to the Dashboard Designer.
5. Click 💾▾, then select **Save Dashboard**.
6. In the Save As window, change the default name, New Dashboard to `Sales by Gender Dashboard` and locate a folder, such as the /Dashboards folder.
7. Click **Save**.

2.4.2 Adding a Hyperlink to a Chart Dashlet

When working with chart dashlets in the Dashboard Designer, you can specify a link that opens when a user clicks the dashlet. You can optionally include a parameter which references an existing filter in another dashlet, and uses that filter to display a specific web page relevant to the filtered information.

The following example takes you through adding a hyperlink to the chart dashlet in the Sales by Gender dashboard created in **2.4.1, "Creating a Web Page Dashlet," on page 43**. The example also shows how to add a parameter to a hyperlink.

First, open the dashboard:

1. If the Sales by Gender dashboard created in **2.4.1, "Creating a Web Page Dashlet," on page 43** (above) is not open, locate the /Dashboards folder in the repository. Right-click the dashboard name and select **Open**

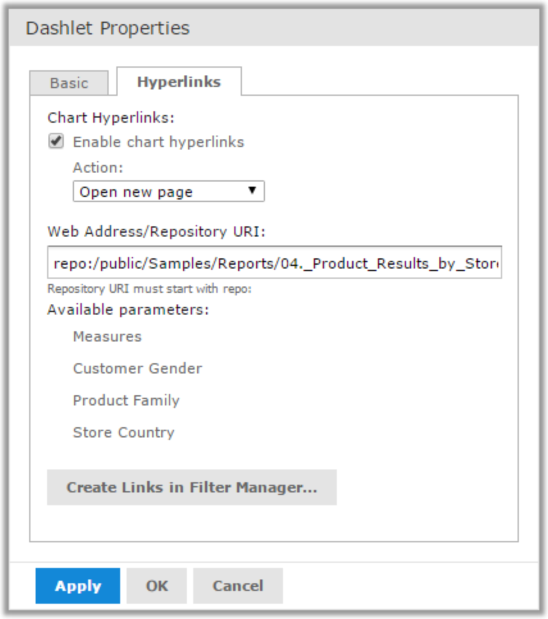
in Designer from the context menu.

The Sales by Gender Dashboard appears in the designer.

Create a chart hyperlink:

1. Right-click on the Sales Mix by Gender dashlet and select **Properties...**.

   The Dashlet Properties dialog box opens.
2. Click the Hyperlinks tab.



Dashlet Properties

Basic | Hyperlinks

Chart Hyperlinks:

☑ Enable chart hyperlinks

Action:

Open new page ▾

Web Address/Repository URI:

repo:/public/Samples/Reports/04._Product_Results_by_Stor

Repository URI must start with repo:

Available parameters:

Measures

Customer Gender

Product Family

Store Country

Create Links in Filter Manager...

Apply | OK | Cancel

Figure 2-14 Hyperlinks tab in Dashlet Properties

3. **Click Enable chart hyperlinks** to select it.
4. Choose **Open new page** from the **Action** menu.

   The Web Address/Repository URI entry bar is displayed.
5. Enter `repo:/public/Samples/Reports/04._Product_Results_by_Store_Type_Report` for the Web Address/Repository URI.

   To see the report's path, hover over the report's name in the Existing Content pane.
6. Click OK.

Preview the new dashboard functionality:

1. Click 👁 to preview the dashboard.
2. Click the chart dashlet.

   The 04. Product Results by Store Type Report opens in a new tab.
3. Click 👁 to return to the Dashboard Designer.

Add a parameter:

1. Right-click on the Sales Mix by Gender dashlet, select **Properties...**, and click the Hyperlinks tab.

   The Available Parameters area shows the parameters available in the chart.

2. Modify the URL to add a parameter that provides a value for an input control in the target report:

   - You need to know the correct name of the input control in your target report. In this case it is `sales__store__store_contact__store_country_1` in the 04. Product Results by Store Type Report.
   - Create a parameter. If you create a parameter name, it is helpful to use a name that is not used in Parameter Mapping. In this example, use `LinkCountry`. For more information about parameters, see **2.4, "Specifying Parameters in Dashlets," on page 41**.

     The link is as follows:

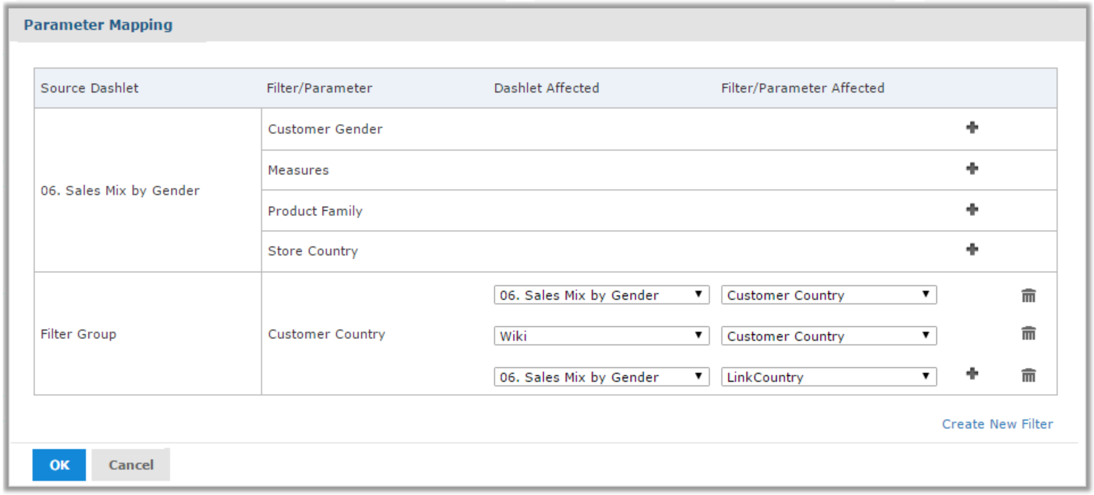
     ```
     repo:/public/Samples/Reports/04._Product_Results_by_Store_Type_Report?sales__
     store__store_contact__store_country_1=$P{LinkCountry}
     ```

If you use one of the names in the Available Parameters list, the mapping to the parameter is created for you. In this example, if you use `Store Country` instead of `LinkCountry`, you do not need to create links in Parameter Mapping.

This example does not show how to add support when more than one country is selected. For more information, see **2.4, "Specifying Parameters in Dashlets," on page 41**.

3. Click **Create Links in Parameter Mapping**. Click **Yes** when prompted.

   Parameter Mapping is displayed.



Source Dashlet	Filter/Parameter	Dashlet Affected	Filter/Parameter Affected
06. Sales Mix by Gender	Customer Gender		
	Measures		
	Product Family		
	Store Country		
Filter Group	Customer Country	06. Sales Mix by Gender	Customer Country
		Wiki	Customer Country
		06. Sales Mix by Gender	LinkCountry

Figure 2-15 Parameter Mapping

4. In the **Customer Country** filter group, click ✚.

   A new row with affected dashlet and filter/parameter drop-down menus appears.

5. In the Dashlet Affected column, select **06. Sales Mix by Gender** from the new Select dashlet... drooping menu.

6. In the Filter/Parameter Affected column, select **LinkCountry** from the new Select parameter... dropdown menu.

7. Click **OK**.

Finally, preview the new dashboard functionality:

1. Click 👁 to preview the dashboard.
2. Click in the **Customer Country** text box to display the available countries.
3. Select **Mexico** from the values list, and click **Apply** at the bottom of the dashlet.
4. Click the chart dashlet.

   The 04. Product Results by Store Type Report opens in a new tab.
5. To verify that the parameter has been passed correctly, click in ☰ the 04. Product Results by Store Type Report.

   The Input Controls dialog box opens, with Mexico set.
6. Return to the tab that contains your dashboard.
7. Click 👁 to return to the Dashboard Designer.
8. Click 💾▾, then select **Save Dashboard**.

2.5 Editing a Dashboard

You can edit a dashboard if you have the proper permissions.

To edit a dashboard:

1. Select **View > Repository** and search or browse for the Dashboard you want to modify.

   By default, the repository includes the /Dashboards folder where you can store dashboards.
2. Right-click the dashboard and select **Open in Designer** from the context menu. The designer appears, displaying the dashboard.
3. Edit the dashboard by adding, removing, resizing, or dragging content.

   For more information about working with dashboard content, see **"Creating a Dashboard" on page 35**.
4. When you are satisfied with the dashboard, hover your cursor over 💾▾ and select **Save Dashboard**. To create a new version of the dashboard, select **Save Dashboard As** and specify a new name.

2.6 Exporting Dashboards and Dashlets

You can create a snapshot of a dashboard or dashlet and save it on your computer. Exporting a dashboard or dashlet requires the following:

- The export button has been enabled for the dashboard or dashlet. See **"Dashboard Properties" on page 28** and **"Dashlet Properties" on page 30** for more information.
- PhantomJS is installed on the computer hosting JasperReports Server. For information on configuring PhantomJS for dashboards, see the System Configuration chapter in the JasperReports Server Administrator Guide.

To export a dashboard:

1. Select **View > Repository** and search or browse for the Dashboard you want to export.
2. Click the link to open the dashboard.

3. Hover your cursor over for the dashboard or individual dashlet and select the export format from the drop-down list.

   The available formats are:

   - PNG (Dashboard only)
   - PDF
   - Excel (Paginated) (Dashlet only)
   - Excel (Dashlet only)
   - CSV (Dashlet only)
   - DOCX
   - RTF (Dashlet only)
   - ODT
   - ODS (Dashlet only)
   - XLSX (Paginated) (Dashlet only)
   - XLSX (Dashlet only)
   - PPTX
4. Save the dashboard or dashlet in the export file format, for example PDF, or open it in the application.

If you perform the export while a dashlet is reloading its content, the dashlet will appear as a grayed-out box in the exported file.

2.6.1 Localizing Controls

You can design dashboard controls to accommodate different languages. First, use the $R syntax to define prompts and static lists of values. Next, attach resource bundles to the report that contain translations of the prompts and lists of values. Finally, add the report to the dashboard. For more information about how to localize input controls, see **"Localizing Reports" on page 190**.

2.7 Tips for Designing Dashboards

Charts and small crosstabs are best suited to dashboards. However, you can design table reports that work well in the dashboard. Such reports tend to be very narrow and are typically used with input controls to limit the number of rows they return.

Keep reports small because dashboards typically contain more than one. In particular, reports shouldn't be too wide, as horizontal room is always at a premium in a dashboard. The server strips margins from an Ad Hoc report when displaying the report on a dashboard.

2.7.1 Input Control Tips

When designing input controls for a dashboard, keep these guidelines in mind:

- To pass a value to an external URL, the **URL Parameter Name** you give to the input control must match the name of a parameter that the URL can accept. The value of the input control must also be a value the URL can accept. The target URL is likely to have additional requirements and limitations. For example, the name of the parameter may be case-sensitive; in this case, the value you enter in the **URL Parameter Name** field is also case-sensitive.

The input control must pass data that the URL can accept. Otherwise, the server may be unable to retrieve the correct data from the external URL.

2.7.2 Time-Date Wildcards

When creating dashlets, you can enter a wildcard into a text input field to display the current date or time:

- `$Date`: Displays the current date in the `Month Day, Year` format. For example, `January 15, 2016`. You can also customize how the date is displayed using the following parameters:
  - `$Date{MM-DD-YYYY}`. For example, January 15, 2016 is displayed as `01-15-2016`.
  - `$Date{DD/MM/YYYY}`. For example, January 15, 2016 is displayed as `15/01/2016`.
- `$Time`: Displays the current time in the `hh:mm am/pm` format, but without the leading `0` for hours. For example, `1:25 pm`. You can customize how the time is displayed using the following parameters:
  - `$Time{hh:mm}`. Displays the time in the 24 hour time format. For example, 1:25 pm is displayed as `13:25`.
  - `$Time{hh:mm a}`. Displays the current time in the `hh:mm am/pm` format with the leading `0` for hours. For example, `01:25 pm`.
  - `$Time{hh:mm:ss}`. Displays the current time in the 24 hour time format with seconds. For example, 1:25:30 pm is displayed as `13:25:30`.

The wildcards are available for text fields, image link, and web links, but it is not available for the dashlet name field.

When you type the `$` symbol into a dashlet's text box, a drop-down list with the time-date wildcards and other available parameters is displayed. Select the wildcard you want from the list or continue typing to narrow down the list.

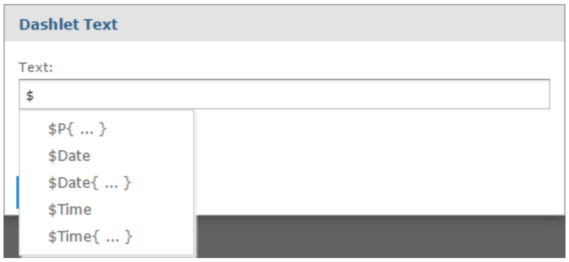


Figure 2-16 Time-Date Wildcard Auto-complete

2.8 Viewing Legacy Jaspersoft Dashboards

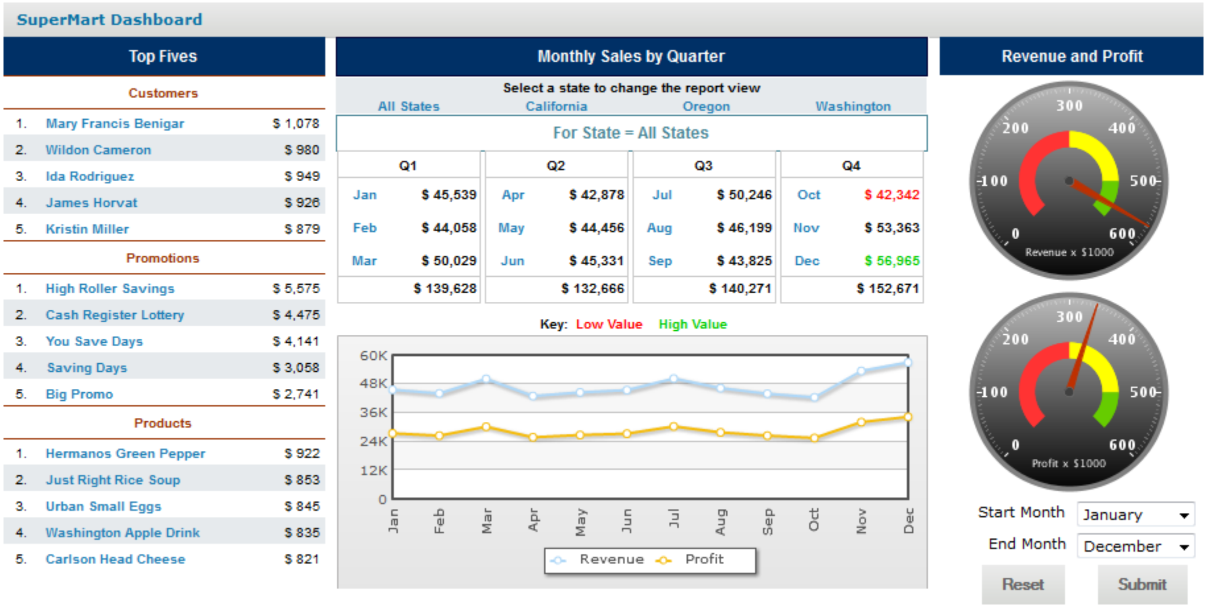
This section refers to legacy dashboards, those created in JasperReports Server version 5.6.2 and earlier.

Dashboards created prior to JasperReports Server version 6.0 open in their respective versions of the Dashboard Designer. The following procedure shows you some of the features of legacy dashboards. For more information about the features available for legacy dashboards, see the edition of the *JasperReports Server User Guide* corresponding to the version of JasperReports Server where the dashboards were created.

To view a legacy dashboard:

1. Open your dashboard for viewing. This example uses the SuperMart Performance dashboard, one of the samples in an earlier version of JasperReports Server. Although this dashboard may not be available in your version of JasperReports Server, other legacy dashboards are similar.

   The SuperMart dashboard includes three reports:



SuperMart Dashboard

Top Fives

Customers

1.	Mary Francis Benigar	$ 1,078
2.	Wildon Cameron	$ 980
3.	Ida Rodriguez	$ 949
4.	James Horvat	$ 926
5.	Kristin Miller	$ 879

Promotions

1.	High Roller Savings	$ 5,575
2.	Cash Register Lottery	$ 4,475
3.	You Save Days	$ 4,141
4.	Saving Days	$ 3,058
5.	Big Promo	$ 2,741

Products

1.	Hermanos Green Pepper	$ 922
2.	Just Right Rice Soup	$ 853
3.	Urban Small Eggs	$ 845
4.	Washington Apple Drink	$ 835
5.	Carlson Head Cheese	$ 821

Monthly Sales by Quarter

Select a state to change the report view

All States California Oregon Washington

For State = All States

Q1		Q2		Q3		Q4	
Jan	$ 45,539	Apr	$ 42,878	Jul	$ 50,246	Oct	$ 42,342
Feb	$ 44,058	May	$ 44,456	Aug	$ 46,199	Nov	$ 53,363
Mar	$ 50,029	Jun	$ 45,331	Sep	$ 43,825	Dec	$ 56,965
	$ 139,628		$ 132,666		$ 140,271		$ 152,671

Key: Low Value High Value

Revenue and Profit

Start Month January

End Month December

Figure 2-17 SuperMart Dashboard Example

2. When you hover over a report, controls appear for that report: ↻ ➦ Click **Refresh** to refresh the report content, and click **Open in a new window** to open the report in a new window.
3. Select new values from the **Start Month** and **End Month** drop-downs and click **Submit** to change the data displayed. All three reports update to display data for the months you indicate.
4. Click **Reset** to set the input controls to the last values saved and return the dashboard to its initial view.
5. When done, click **View > Repository** to go to the repository page.

Keep these points in mind when viewing a dashboard that has input controls:

- An input control may appear as a text field, a drop-down, a check box, a multi-select list box, or a calendar icon.
- If one of the frames in a dashboard does not refer to an input control, that frame does not update when you change that input control's value. Only reports that refer to the input control reflect the change.

If a dashboard includes a **Print View** button, click it to display the dashboard without JasperReports Server's header and footer. Depending on your web browser, this may also open your browser's Print window.

CHAPTER 3 RUNNING REPORTS AND THE REPORT VIEWER

JasperReports Server makes it easy to run reports. When you run a report, it opens in the interactive Report Viewer. With the Viewer, you can personalize and refine the displayed report data. If the report has input controls, you run the report with one set of data and then another.

This chapter contains the following sections:

- **Overview of The Report Viewer**
- **Running or Creating a Simple Report**
- **Running a Flash Chart**
- **Running a Report with Input Controls or Filters**

The tutorials in this chapter and throughout this guide assume you've installed the sample data provided with the server.

3.1 Overview of The Report Viewer

The Report Viewer allows you to view a report, export content to various output formats, and apply formatting, sorting, and filters to control how the data is displayed.

This section describes the functions available in the Report Viewer. You can find more detailed information about using this functionality throughout this chapter.

To open a report in the Report Viewer:

1. Locate your report in the library or repository.
2. Click the report name, or right-click the report name and select **Run**. In the repository, you can also click the report row and select **Run** from the tool bar. The report opens in the Report Viewer.

3.1.1 The Report Viewer Tool Bar

The Report Viewer tool bar contains a number of controls for working with your report. These controls are described in **Table 3-1**.

Table 3-1 Report Viewer Tool Bar Icons

Icon	Name	Description
⟳	Refresh report with latest data	Click to refresh the report data from the data source.
◀◀	First	Click to jump to the first page of the report.
◀	Previous	Click to go to the previous page in the report
Page 1 of 47	Current Page	Shows the number of the currently displayed report page.
▶	Next	Click to go to the next page in the report.
▶▶	Last	Click to jump to the last page of the report.
−	Zoom Out	Click to zoom out on the report.
+	Zoom In	Click to zoom in on the report
100% ▾	Zoom Options	Click this icon to open the Zoom Options drop-down menu.
search report 🔍 ▾	Search	Enter your search term here to find a text string in your report. Click the drop-down menu to toggle search preference settings.
◀	Search Previous	Click to jump to the previous instance of the search term.
▶	Search Next	Click to jump to the next instance of the search term.
Back	Back	Exits the Report Viewer and takes you to the previous screen.

Icon	Name	Description
💾	Save	Place the cursor over this icon to open a menu of save options.
⇨	Export	Click to export the View into one of the available formats.
↶	Undo	Click to undo the most recent action.
↷	Redo	Click to redo the most recently undone action.
↺	Undo All	Click to revert the report to its most recently saved state.
☰	Input Controls	Click to see the input controls applied to this report. For more information, refer to **"Simple Input Controls" on page 69**.
🔖	Bookmarks	Click to display the Table of Contents pane. This pane is available only on reports with enabled Bookmarks.

3.1.2 Column Menu

Reports that contain table components are enabled for user interactivity. Table components are defined in Jaspersoft Studio or from Ad Hoc Views. When a table is enabled for interactivity, column formatting, filtering, and sorting are managed from a menu displayed by clicking the column you want to apply changes to. These menu icons are described in **Table 3-6**.

Table 3-2 Column Formatting Icons

Icon	Name	Description
▥▾	Formatting/Show column/Hide column	Select **Formatting** to open the Format Column box. Select **Show column** or **Hide column**.

Icon	Name	Description
▽	Column filters	Click to open the Filter column box.
⬆	Sort ascending	Click to sort fields in the selected column in ascending order.
⬇	Sort descending	Click to sort fields in the selected column in ascending order.
▼	Column size	Click and drag this icon to make columns wider or narrower.

3.1.3 Data Snapshots

Some reports have an optional data snapshot feature enabled. A data snapshot is a cached copy of the data included in a specific report. Data snapshots allow you to access a report's data (including input control settings) without having to retrieve it from the data source, which in some cases can save a significant amount of time.

When a report is opened in the Report Viewer, data is retrieved from the data snapshot. If the snapshot does not exist, then a live query is made to the data source. A snapshot is created when a report is saved from the viewer, or via the scheduler. The Report Viewer UI displays a date and time stamp that indicates when the report data was last refreshed with live source data.

The system administrator can enable or disable the data snapshot feature.

It should be noted that a report can have only one snapshot. For instance, if you edit and save a report that already has a snapshot associated with it, a new snapshot overwrites the previously-created snapshot.

3.2 Running or Creating a Simple Report

You can view and work on a report in the Report Viewer in a number of ways:

- Running an instance of an existing report
- Creating a new report from an existing Ad Hoc view

3.2.1 Running a Simple Report

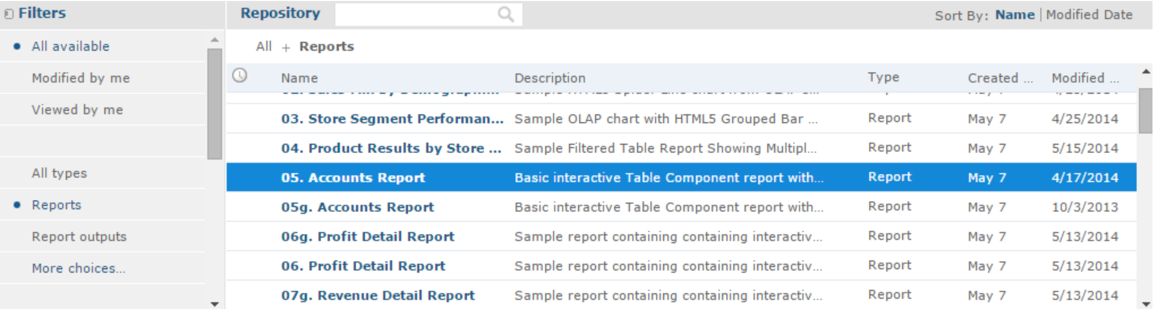
This section describes how to run a tabular report that lists account data.

To run a report:

1. Log into the server as an administrator, such as jasperadmin.

2. On the Home page, click **View list** in the **Reports** block.

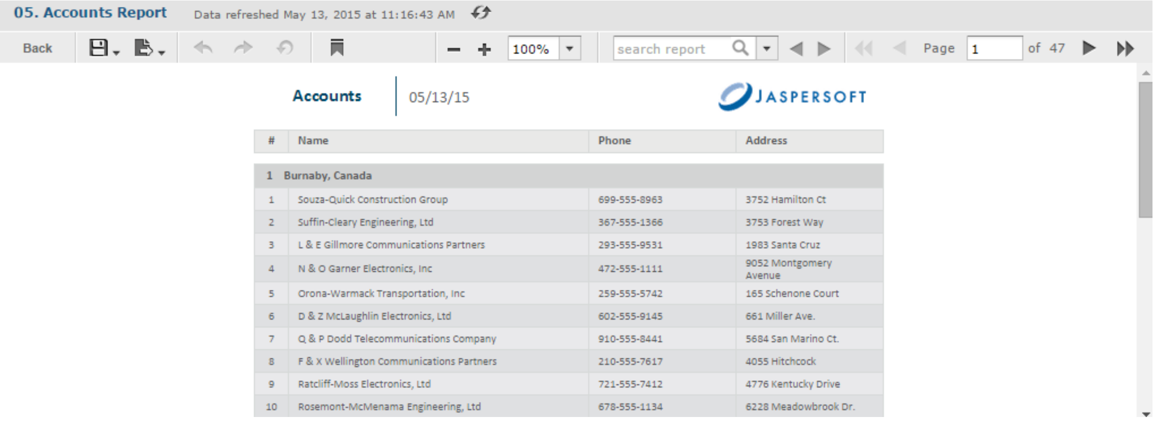
   The search results appear, listing your own files and other files that your user account has permission to view. If you log in as jasperadmin, 05. Accounts Report appears in the search results.



Name	Description	Type	Created ...	Modified ...
03. Store Segment Performan...	Sample OLAP chart with HTML5 Grouped Bar ...	Report	May 7	4/25/2014
04. Product Results by Store ...	Sample Filtered Table Report Showing Multipl...	Report	May 7	5/15/2014
05. Accounts Report	Basic interactive Table Component report with...	Report	May 7	4/17/2014
05g. Accounts Report	Basic interactive Table Component report with...	Report	May 7	10/3/2013
06g. Profit Detail Report	Sample report containing containing interactiv...	Report	May 7	5/13/2014
06. Profit Detail Report	Sample report containing containing interactiv...	Report	May 7	5/13/2014
07g. Revenue Detail Report	Sample report containing containing interactiv...	Report	May 7	5/13/2014

Figure 3-1 Search Results Listing

3. To run a report, click the name of a report in the repository. For example, click **05. Accounts Report**. The report appears, as shown in **Figure 3-2**.



Accounts | 05/13/15

#	Name	Phone	Address
1	Burnaby, Canada		
1	Souza-Quick Construction Group	699-555-8963	3752 Hamilton Ct
2	Suffin-Cleary Engineering, Ltd	367-555-1366	3753 Forest Way
3	L & E Gillmore Communications Partners	293-555-9531	1983 Santa Cruz
4	N & O Garner Electronics, Inc	472-555-1111	9052 Montgomery Avenue
5	Orona-Warmack Transportation, Inc	259-555-5742	165 Schenone Court
6	D & Z McLaughlin Electronics, Ltd	602-555-9145	661 Miller Ave.
7	Q & P Dodd Telecommunications Company	910-555-8441	5684 San Marino Ct.
8	F & X Wellington Communications Partners	210-555-7617	4055 Hitchcock
9	Ratcliff-Moss Electronics, Ltd	721-555-7412	4776 Kentucky Drive
10	Rosemont-McMenama Engineering, Ltd	678-555-1134	6228 Meadowbrook Dr.

Figure 3-2 Output of the Accounts Report

If you are running a report with multiple pages, the first page of the report appears before the entire report is loaded. You can begin scrolling through report pages as they load, as indicated in the pagination controls in the upper left corner of the Report Viewer.

If you want to cancel loading the report before it is complete, click the **Cancel Loading** button that appears next to the pagination controls.

3.2.2 Creating a Report

You can create a report directly from the Jaspersoft Server Home page. This method allows you to select an existing Ad Hoc view and generate a report from it, without going through the Ad Hoc Editor.

To create a report from the Home page:

1. On the Home page, click **Create** in the **Reports** block. The Create Report wizard opens.

2. Select the Ad Hoc view you want to use as the basis for your report.
3. Select a report template. To use a template other than the default, select Custom Report Template, click Browse and select the desired template. See **3.2.3, “Report Templates,” on page 58** for more information.
4. Click **OK**. If asked, enter the input controls needed. See **“Using Input Controls” on page 149**.

You can now begin working with your report.

3.2.3 Report Templates

When you create a report, the Create Report wizard displays layout options for generating and exporting the report:

- **Default Report Template** applies basic layout options to your report. This is usually the Actual Size template.
- **Custom Report Template** allows you to browse to an existing template. JasperReports Server includes a number of templates are available by default, including:
  - A4 Landscape
  - A4 Portrait
  - Actual Size
  - Letter Landscape
  - Letter Portrait.

  Other report templates may be available. Report templates can be created in Jaspersoft Studio and uploaded to JasperReports Server.
- **Report Generator** allows you to create a highly customized report design. This option is not often enabled. See your JasperReports Server administrator for more information.

Most commonly, you will choose **Default Report Template**.

3.2.3.1 Using Report Templates for PDF

If you are exporting your report to PDF, choose your option based on the size of the output.

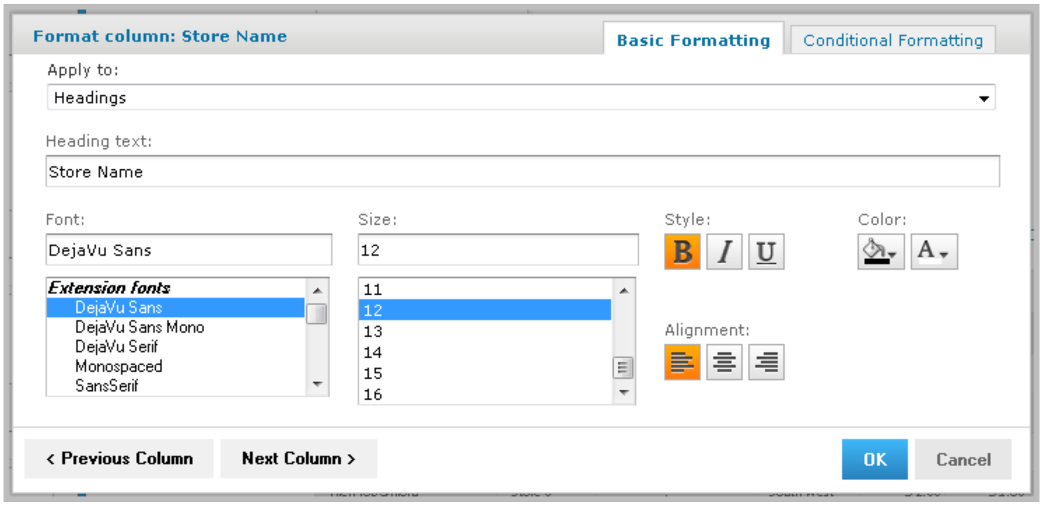
- For most PDF exports, you can use Actual Size, which supports a maximum size of 14400px by 14400px.
- For reports with an output height exceeding 14400 px, use a paginated report template that is wide enough for your report. For example, if you have a long report with width less than 842px, you can use the paginated A4 Landscape theme. A report designer can create additional custom templates in Jaspersoft Studio. contact your administrator for more information.
- Reports with output width exceeding 14400 px will be truncated in PDF. Redesign your report or use a different export format.

3.3 Getting New Perspectives on Data

The report shown in **Figure 3-2** was created using the Ad Hoc Editor. As this type of report runs, you can interact with it in the Report Viewer to visualize the data in different ways. Column formatting allows you to highlight certain columns and fields, and filtering and sorting report output on-the-fly can provide timely views of the data that answer your questions. For example, suppose you’re running the Accounts Report and want to know how many accounts have offices nearby. Highlighting the phone number column with red text and filtering it to show only accounts in your area code would reveal this data.

3.3.1 Column Formatting

You can customize the basic formatting of column headings and fields, using the Format Column dialog. Hover over ▥ and click **Formatting...** The Format column dialog appears.



Format column: Store Name

Apply to: Headings

Heading text: Store Name

Font: DejaVu Sans

Size: 12

Figure 3-3 Format Column Dialog

You can alter a column’s basic formatting or apply conditional formatting to a column.

In longer reports, columns have floating headers. If your report extends past your browser frame, use the vertical scroll bar to move up and down the list. If you do not see a vertical scroll bar, try increasing the width of your browser window.

This section discusses how to apply formatting to column headings and values. For information on conditional formatting, see **“Conditional Formatting” on page 60**.

Column formatting options include:

- Text
- Font type, size, and style
- Background color
- Font color
- Text alignment

To customize your column formatting:

1. Run your report, so it opens in the Report Viewer.
2. Click the column you want to format.
3. Hover over ▥ and click **Formatting...**
4. Click the **Basic Formatting** tab, and change the following options if needed:
   - **Apply to** – Select the part of the column you want to apply the formatting to.
   - **Heading text** – Type new heading text to replace the current text.
   - **Font** – Scroll through the menu to select a font.
   - **Size** – Scroll through the menu to select a font size.

  - **Style** – Click to select Bold, Italic, or Underlined text.
  - **Background Color** – Click to open the background color picker, then click to select the background color.
  - **Font Color** – Click to open the font color picker, then click to select the text color.
  - **Alignment** – Click to select Left, Center, or Right alignment.
5. If needed, click **Previous Column** or **Next Column** to change the formatting for an adjacent column.
6. Click **OK**.

3.3.2 Conditional Formatting

The Report Viewer allows you to format column headings and fields, to highlight data that meets specific criteria. For instance, if you want to call out fields for store sales above $100,000, you can do so by applying text and background formatting to those stores that meet those numbers.

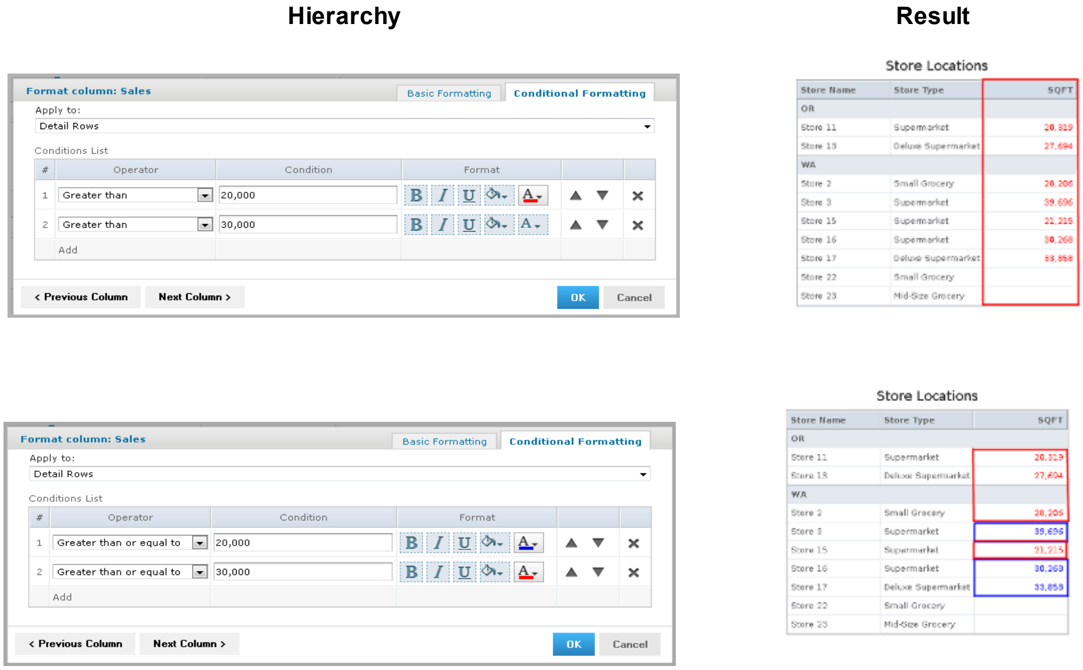
With conditional formatting, you can apply the formatting options listed in **"Column Formatting" on page 59**. However, it is a slightly more complex process than applying formatting options to entire columns. This section describes those complexities, including:

- Condition hierarchy
- Condition button states
- Applying conditional formatting

3.3.2.1 Condition Hierarchy

If you have multiple conditions applied to a single field, their order will affect how they function. Conditions are read and applied from bottom to top, and the topmost condition overrides the one(s) below.

For example, imagine you have more than one condition applied to the same format element: red text for all stores over 20,000 square feet, and blue text for all stores over 30,000 square feet. As shown in **"Condition Hierarchy Example" on page 61**, placing the formatting rule for stores over 20,000 above the rule for stores over 30,000 causes the topmost rule to override the one below:



Hierarchy

Format column: Sales

Apply to: Detail Rows

Conditions List

#	Operator	Condition	Format
1	Greater than	20,000	B I U A
2	Greater than	30,000	B I U A

< Previous Column | Next Column > | OK | Cancel

Result

Store Locations

Store Name	Store Type	SQFT
OR		
Store 11	Supermarket	20,319
Store 13	Deluxe Supermarket	27,694
WA		
Store 2	Small Grocery	20,206
Store 3	Supermarket	39,696
Store 15	Supermarket	21,215
Store 16	Supermarket	30,268
Store 17	Deluxe Supermarket	33,858
Store 22	Small Grocery	
Store 23	Mid-Size Grocery	

Format column: Sales

Apply to: Detail Rows

Conditions List

#	Operator	Condition	Format
1	Greater than or equal to	20,000	B I U A
2	Greater than or equal to	30,000	B I U A

< Previous Column | Next Column > | OK | Cancel

Store Locations

Store Name	Store Type	SQFT
OR		
Store 11	Supermarket	20,319
Store 13	Deluxe Supermarket	27,694
WA		
Store 2	Small Grocery	20,206
Store 3	Supermarket	39,696
Store 15	Supermarket	21,215
Store 16	Supermarket	30,268
Store 17	Deluxe Supermarket	33,858
Store 22	Small Grocery	
Store 23	Mid-Size Grocery	

Figure 3-4 Condition Hierarchy Example

3.3.2.2 Condition Button States

Because conditions higher up in the hierarchy can affect those below, the font style selection buttons each have three states:

- **Unchanged**, which means it inherits the previous condition-based style, if any.
- **Set**, which means the style is applied to text that meets the condition.
- **Not Set**, which means the style is not applied to the text that meets the condition, and is removed if a conflicting condition lower in the conditional formatting hierarchy has marked that style as “Set”.

By default, the buttons are in the “Unchanged” state. Clicking the buttons toggles you through the three states.

See **“Style Button States”** for examples of the style button states.

Table 3-3 Style Button States

	Unchanged	Set	Not Set
Bold	B	B	B
Italic	I	I	I
Underline	U	U	U

The background and font color pickers have buttons for similar states, but these states behave slightly differently:

- **Unchanged**, which means the field inherits the previous condition-based color, if any.
- **Set**, which means the color is applied to text or background of the field that meets the condition.
- **No Fill (background only)**, which means no color is applied to the background that meets the condition. Regardless of conditions lower in the hierarchy, the background inherits the table's default color.

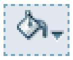
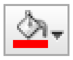
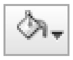
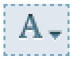
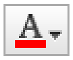
Both have two buttons at the top of the window, along with the color selection boxes.

You control these states through the background color picker and the font color picker windows, using the following buttons:

- **No Fill (background only)**, which applies the No Fill state described above.
- **Reset**, which returns the text or background to the Unchanged state.
- The **color selection boxes**, which apply the Set state.

See **"Color Picker Button States"** for examples of the color picker button states.

Table 3-4 Color Picker Button States

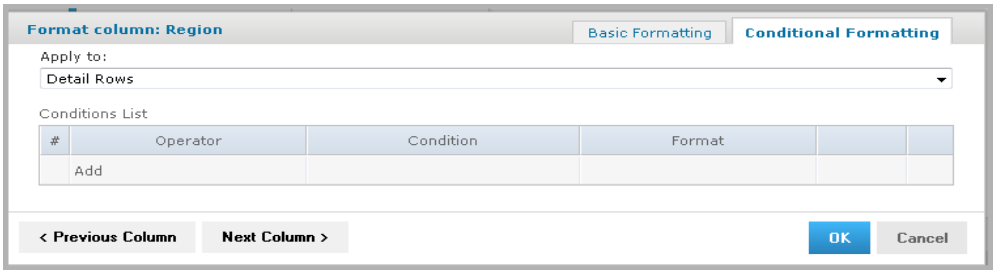
	Unchanged	Set	No Fill
Background Color			
Text Color			N/A

3.3.2.3 Applying Conditional Formatting

You apply conditional formatting much like you do standard column formatting, as described in **"Column Formatting" on page 59** with the extra step of creating the condition by which the formatting is applied.

To create a condition:

1. Run your report, so it opens in the Report Viewer.
2. Click the header or field of the column you want to format.
3. Move your mouse over ▥▾ and click **Formatting...**
4. Click the **Conditional Formatting** tab. The Conditional Formatting options appear:



Apply to: Detail Rows

Figure 3-5 Conditional Formatting Tab

5. In the **Apply to** box, select the part of the column you want to apply the formatting to.
6. Click **Add**. This adds a line item in the Conditions List.
7. Fill in the following information:
   - **Operator**: Use the drop down menu to define how the condition is compared to the column data.
   - **Condition**: Enter the condition criteria.
   - **Format**: Select the formatting applied to fields meeting the defined condition. Take care setting the button states, as described in **"Condition Button States" on page 61**.
8. Repeat if needed to add multiple conditions to a column.

   If you have multiple conditions, you may want to reorder them, to ensure they do not conflict with each other. Use the ▲ and ▼ to move conditions in the hierarchy.
9. If needed, click **Previous Column** or **Next Column** to change the conditional formatting for an adjacent column.
10. Click **OK**. The condition is applied to the column.

3.3.3 Interactively Filtering Report Output

If the report output contains more information than you want, interactively filter it to display just what you need. You conditionally filter report output by first selecting the column to use as a basis for filtering. Next, you enter a filter condition, then a value for comparison. The server compares each field of the column to the value that meets the condition. In **Table 3-5** you can see the conditions available for each type of column: numeric, date, and text.

Table 3-5 Interactive Filtering Conditions

Numeric	Date	Text
Equals	Equals	Equals
Does not equal	Is not equal to	Is not equal to
Greater than	Is between	Contains
Greater than or equal to	Is not between	Does not contain
Less than	Is on or before	Starts with

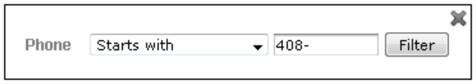
Numeric	Date	Text
Less than or equal to	Is before	Does not start with
Is between	Is on or after	Ends with
Is not between	Is after	Does not end with

To interactively filter report data:

1. As you run the type of report shown in **Figure 3-2**, click the column you want to use for filtering the report. Continuing with the example in **"Running or Creating a Simple Report" on page 56**, click the **Phone** column in the Accounts report.
2. Click the ▼ .

   The **Filter column** dialog appears, as shown in **Figure 3-6**. By default, **Show all rows** is selected.
3. To build your filter, click the radio button to select **Show only rows where**.

   the comparison operator drop-down and value entry box become active.
4. Select a comparison operator from the drop-down. For example, select **Starts with** to compare phone numbers starting with certain numbers.
5. Enter a value for comparison with the data in the column. For example, enter the area code: `408-`

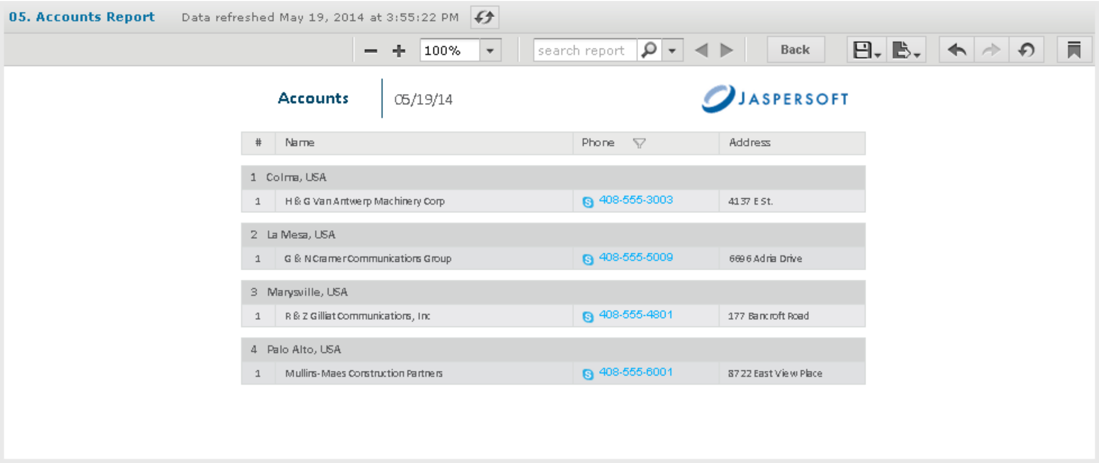


Phone Starts with 408-

Figure 3-6 Filter Column Dialog

6. Click **OK**.

   The view of the report changes to show the filtered output. For example, now the Accounts Report only shows accounts in the 408 area code.



05. Accounts Report Data refreshed May 19, 2014 at 3:55:22 PM

Accounts 05/19/14

#	Name	Phone	Address
1	Colma, USA		
1	H & G Van Antwerp Machinery Corp	408-555-3003	4137 E St.
2	La Mesa, USA		
1	G & N Cramer Communications Group	408-555-5009	6696 Adria Drive
3	Marysville, USA		
1	R & Z Gilliat Communications, Inc	408-555-4801	177 Bancroft Road
4	Palo Alto, USA		
1	Mullins-Maes Construction Partners	408-555-6001	8722 East View Place

Figure 3-7 Filtered Report Shows Only Accounts in Area Code 408

A small star icon appears in the heading of the filtered column, to the right of the heading text.

7. To clear the filter indicator and once again display all the accounts, reopen the Filter column dialog and select **Show all rows**, then click **OK**.

3.3.4 Interactively Sorting a Report

You can also sort data in report output interactively. For example, you can sort the data alphabetically in ascending or descending order within each city listed in the Accounts Report.

To interactively sort report output:

1. As you run the type of report shown in **Figure 3-2**, click the column you want to use for sorting the report. For example, click the **Name** column in the Accounts report.
2. Click ⬆ (sort ascending) or ⬇ (sort descending). The report refreshes, and data appears sorted by the selected column in the selected order. In this example, for instance, if you select ⬆ , the grouped rows of data are sorted alphabetically in ascending order within each group. Row 1 of the Burnaby, Canada group is now D & Z McLaughlin Electronics, Ltd and row 1 of the Cliffside, Canada group is Abbott-Liff Electronics Holdings.

   The heading text in the column used to sort report data appears red, and the up or down arrow icon in the column header indicates that report output now appears in ascending or descending order, respectively.

   To change the sort order of the groups themselves, you have to modify the report query.

3.3.5 Moving, Resizing, and Hiding Columns

Columns are easily moved, resized, and hidden in your report.

- To move a column, click the column you want to move, then drag the column left or right into the new position. The ▼ indicates where the column is placed.
- To resize a column, click the column you want to resize, then drag the ▼ until the column is the size you want.
- To hide a column, click the column you want to hide, then move your mouse over the ▦▾ and select **Hide column.**

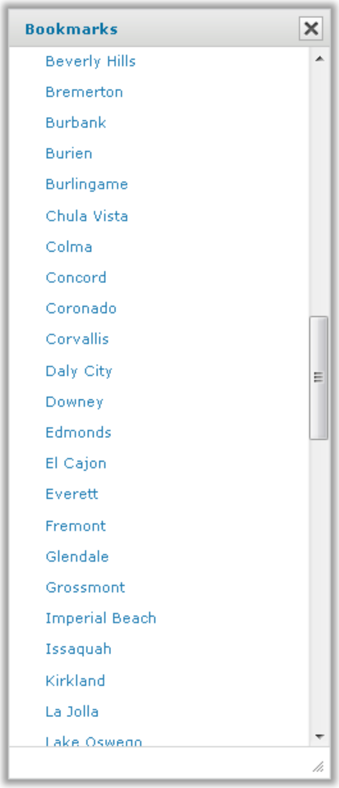
3.3.6 Setting Output Scale

You can determine the display size for any report by using the report scaling options, located in the Report Viewer tool bar.

- Click ➕ to zoom in on the report.
- Click ➖ to zoom out on the report.
- Click 100% ▾ to open the Zoom Options drop-down menu, and select the percentage by which you want to increase or decrease the size of the displayed report.

3.3.7 Using the Bookmarks Panel

When working with a report that contains bookmarks, they are displayed in a floating panel. Using this panel, you can jump to designated sections of the report.



Bookmarks

Beverly Hills
Bremerton
Burbank
Burien
Burlingame
Chula Vista
Colma
Concord
Coronado
Corvallis
Daly City
Downey
Edmonds
El Cajon
Everett
Fremont
Glendale
Grossmont
Imperial Beach
Issaquah
Kirkland
La Jolla
Lake Oswego

Figure 3-8 The Bookmarks Panel

- To display the Bookmarks panel, click 🔖 in the Report Viewer tool bar.
- To jump to a bookmarked section of the report, click the name of the section in the Bookmarks panel.

3.4 Navigating the Report

If your report has multiple pages, you can use the pagination controls to move through the report quickly.

To navigate the published report:

- Use ◀ at the top of the Report Viewer to navigate to the previous page.
- Use ▶ to navigate to the next page.
- Use ▶▶ to go to the end of the report.
- Use ◀◀ to go to the beginning of the report.
- If you know the number of the page you want to view, enter the page number in the Current Page indicator box.

3.5 Exporting the Report

To export the report:

1. To view and save the report in other formats, click the **Export** button.
2. Select an export format from the drop-down. The export options are listed in **Table 3-6**.

Table 3-6 Export File Types

Option	Format Name	Usage
PDF	Adobe Acrobat	Choose a report template based on report size. Use the Actual Size report template for reports with dimensions less than or equal to 14400px by 14400px. See **3.2.3, "Report Templates," on page 58** for more information.
Excel (Paginated)	XLS	Not recommended for exporting most tables or crosstabs. Repeats headers and footers on each page.
Excel	XLS	Ignores page size and produces spreadsheet-like output.
CSV	Comma Separated Values	Characters outside the Latin 1 character set can cause the Excel spreadsheet to look unacceptable. Try saving the file and importing it using Excel's Import functionality.
DOCX	Word	Do not export reports having more than 63 columns. In Microsoft Word, you cannot create tables having more than 63 columns.
RTF	Rich Text Format	Creates a large output file and, therefore, takes longer to export than PDF, for example.
ODT	OpenDocument Text	For best results, minimize the number of rows and columns and make sure they don't overlap.
ODS	OpenDocument Spreadsheet	Same as ODT.
XLSX (Paginated)	Microsoft Open XML Format Spreadsheet	Not recommended for exporting most tables or crosstabs. Repeats headers and footers on each page.
XLSX	Microsoft Open XML Format Spreadsheet	Ignores page size and produces spreadsheet-like output.
PPTX	Microsoft PowerPoint Presentation	Each page of report becomes a slide in the PowerPoint presentation.

3. Save the report in the export file format, for example PDF, or open the report in the application.

3.6 Running a Flash Chart

The JasperReports Server commercial editions support Flash charting, and include the Maps Pro, Charts Pro, and Widgets Pro component libraries.

Using the libraries, you can create visually appealing, animated, and interactive reports:

- Maps Pro – Color-coded maps covering all countries and regions of the globe.
- Charts Pro – Standard and stacked charts with animation and interactivity.
- Widgets Pro – Non-standard charts such as gauges, funnels, spark lines, and Gantt charts.

These components are based on Fusion libraries and generate Flash output that is embedded in the HTML and PDF output. When a report containing a Maps, Charts, or Widgets Pro element is exported in a format other than HTML or PDF, the space used by the element remains blank.

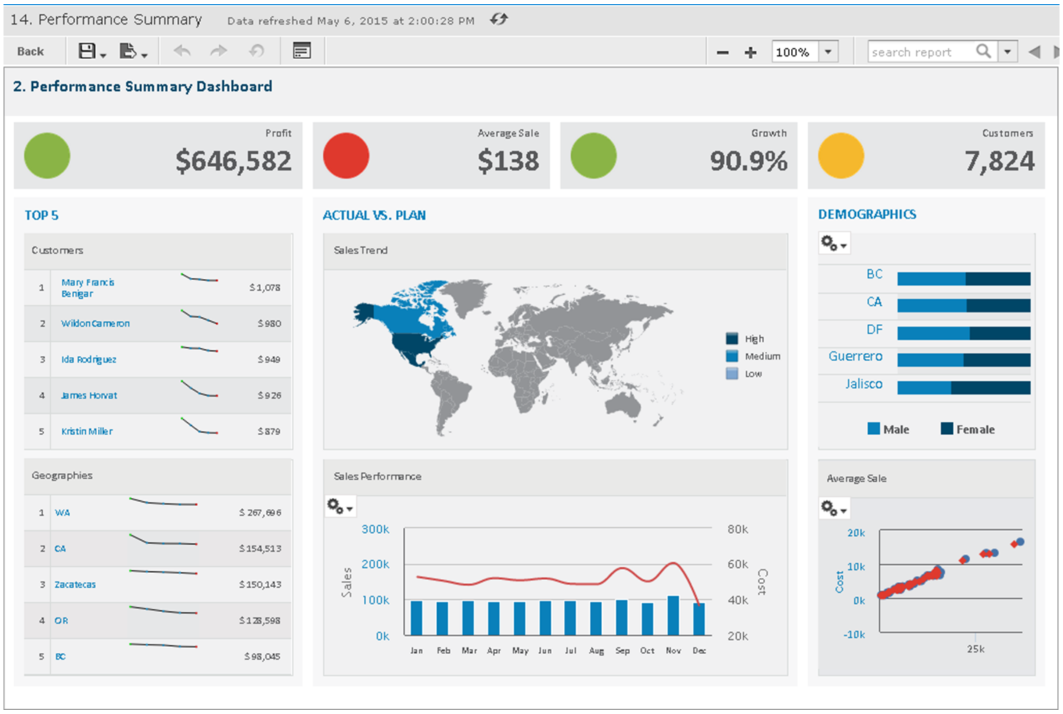
To view a Maps, Charts, or Widgets Pro element in the server, install Flash Player. You might need to install plug-ins or enable Flash on the browser. To view Flash elements in PDF output, use a Flash-enabled PDF viewer such as Adobe Reader.

You can configure the server to default to HTML5 when rendering Pro Charts. See the *JasperReports Server Administrator Guide* for more information.

Flash charts are created in Jaspersoft Studio Professional as JRXML reports and uploaded to the repository as a report unit. JasperReports Server cannot create Flash charts. The sample report **14. Performance Summary** includes flash elements.

To find and run a Flash chart example:

1. In the repository, locate the sample report **14. Performance Summary**.
2. Click the report name to run the report, which launches as part of the Performance Summary Dashboard.



14. Performance Summary Data refreshed May 6, 2015 at 2:00:28 PM

2. Performance Summary Dashboard

Profit $646,582

Average Sale $138

Growth 90.9%

Customers 7,824

TOP 5

Customers		
1	Mary Francis Benigar	$1,078
2	Wildon Cameron	$980
3	Ida Rodriguez	$949
4	James Horvat	$926
5	Kristin Miller	$879

Geographies		
1	WA	$ 267,696
2	CA	$154,513
3	Zacatecas	$150,143
4	OR	$128,598
5	BC	$98,045

Figure 3-9 **Performance Summary Dashboard with Flash Map**

Note the Sales Trend map in the center of the dashboard.

3. To interact with the map, mouse-over any of the countries to see the full country name and, when data exists, the value for that country.
4. On the Sales Trend map, click **Canada** to launch the Interactive Sales Report, displaying the information for Canada only. From there, you can modify the underlying report as needed.

To upload JRXML reports, see **"Adding Reports Directly to the Repository" on page 163**.

3.7 Running a Report with Input Controls or Filters

An input control are graphical widgets that filter the data that appears in a report. The perfect input control limits the data to what you want to see—and nothing more. When you run a report based on a Domain Topic that defines a filter, the server can render the filter as an input control. The JasperReports Server interface uses "input controls," "filters," and "options" interchangeably.

If your system administrator has enabled the data snapshot feature (described in **"Data Snapshots" on page 56**), it is important to note that the default input controls - that is, the input controls as defined when the original Jaspersoft Studio- or Ad Hoc View-based report is run - will overwrite any changes made to them the next time you run a report. For instance, suppose you run a report, update the input controls, then save the report. At a later date, you run a report from the Jaspersoft Studio or Ad Hoc View source again. That new report will replace the report you ran earlier, and your input control changes will be lost.

To avoid this, save a version of the report with your selected data preloaded. That way, when subsequent reports are run from the same source, they will not overwrite your report. See **"Saving Input Control Values" on page 72** for more information.

3.7.1 Simple Input Controls

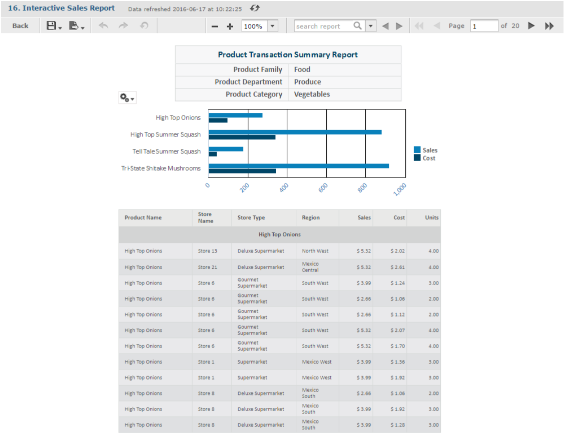
The 16. Interactive Sales Report example has several input controls:

- Product Family
- Product Department
- Product Category
- Product Name
- Country
- Gender

Using input controls, you run the report with one set of data and then another. When saved, an instance of the report with alternate input controls is called a **Report Version**, and is labeled as such in the repository.

To run a report with simple input controls:

1. In the repository, locate and run the report 16. Interactive Sales Report.



16. Interactive Sales Report Data refreshed 2016-06-17 at 10:22:25

Product Transaction Summary Report	
Product Family	Food
Product Department	Produce
Product Category	Vegetables

Product Name	Store Name	Store Type	Region	Sales	Cost	Units
High Top Onions						
High Top Onions	Store 13	Deluxe Supermarket	North West	$ 5.32	$ 2.02	4.00
High Top Onions	Store 21	Deluxe Supermarket	Mexico Central	$ 5.32	$ 2.61	4.00
High Top Onions	Store 6	Gourmet Supermarket	South West	$ 3.99	$ 1.24	3.00
High Top Onions	Store 6	Gourmet Supermarket	South West	$ 2.66	$ 1.06	2.00
High Top Onions	Store 6	Gourmet Supermarket	South West	$ 2.66	$ 1.12	2.00
High Top Onions	Store 6	Gourmet Supermarket	South West	$ 5.32	$ 2.07	4.00
High Top Onions	Store 6	Gourmet Supermarket	South West	$ 5.32	$ 1.70	4.00
High Top Onions	Store 1	Supermarket	Mexico West	$ 3.99	$ 1.36	3.00
High Top Onions	Store 1	Supermarket	Mexico West	$ 3.99	$ 1.92	3.00
High Top Onions	Store 8	Deluxe Supermarket	Mexico South	$ 2.66	$ 1.06	2.00
High Top Onions	Store 8	Deluxe Supermarket	Mexico South	$ 3.99	$ 1.92	3.00
High Top Onions	Store 8	Deluxe Supermarket	Mexico South	$ 3.99	$ 1.28	3.00

Figure 3-10 Interactive Sales Report

2. In the Filters panel, use the Country menu to select USA only.

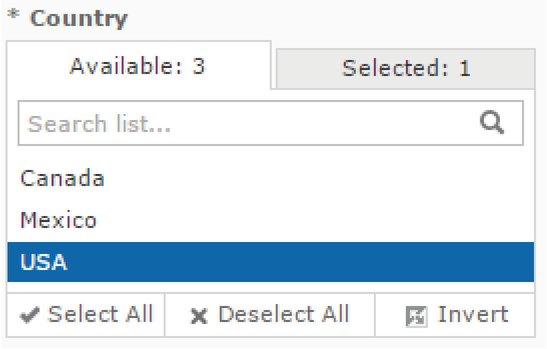


Figure 3-11 Input Control Selection - Country

3. Click **Close**, then click **Apply** at the bottom of the panel. The report shows data for the USA only.
4. Click 💾, then select **Save As**. You are prompted to name the new report.
5. Enter a new name, then click **Save**.

3.7.2 Multi-select Input Controls

The 1. Geographic Results by Segment Report has an example of a multi-select input control for Product Name.

To run a report with a multi-select input control:

1. In the repository, locate and run the report 1. Geographic Results by Segment Report.

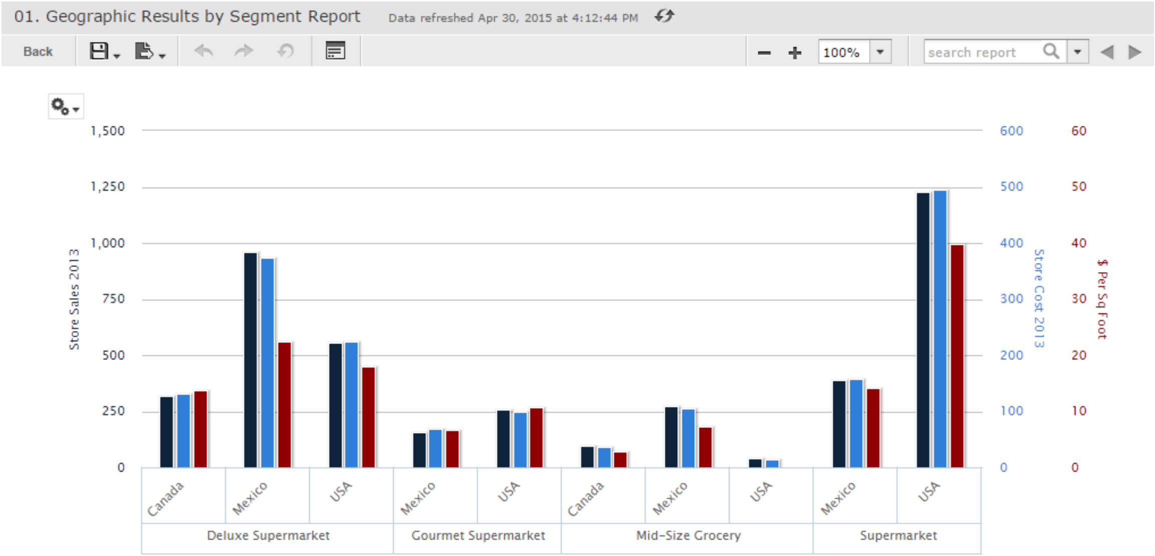


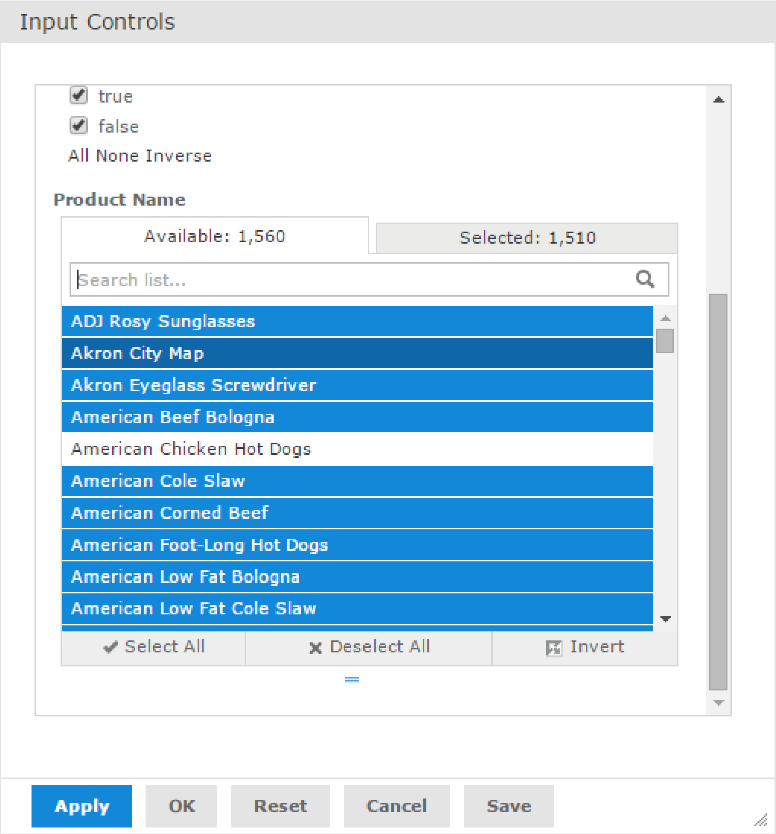
Figure 3-12 Geographic Results by Segment Report

2. On the tool bar, click ▤ .
3. Scroll to the bottom of the Input Controls dialog to view the Product Name input control.
4. Enter **chicken** in the Search list 🔍.

   Only products containing "chicken" are shown.
5. Click the first item in the list, then scroll down and Shift-click to select all the items with "chicken".
6. Click in the Search list again and delete the word **chicken**.

   All items are displayed, with ones that include "chicken" selected.
7. Click the **Selected** tab to see that only items that are selected are displayed. Then click the **Available** tab to view all the items.
8. Click **Invert** ▨ to invert the selection.

   Now all items are selected except those items that contain "chicken".
9. Click **OK** at the bottom of the panel. The report shows data for products that do not contain the word "chicken" only.



☑ true
☑ false

Product Name

Available: 1,560 | Selected: 1,510

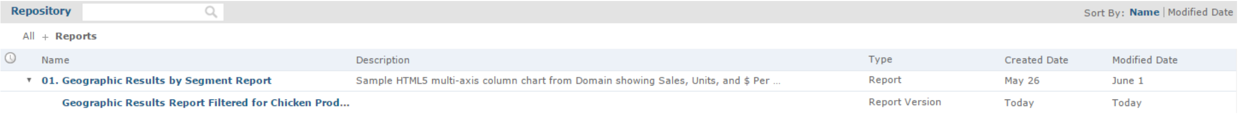
- ADJ Rosy Sunglasses
- Akron City Map
- Akron Eyeglass Screwdriver
- American Beef Bologna
- American Chicken Hot Dogs
- American Cole Slaw
- American Corned Beef
- American Foot-Long Hot Dogs
- American Low Fat Bologna
- American Low Fat Cole Slaw

Figure 3-13 Multi-Input Control Selection

3.7.3 Saving Input Control Values

You can save your selected input control values to use at another time. You will have the original report and a copy of it JasperReports Server saves a version of the report with the selected values as a child of the original report. This new version of the report appears as a child of the original report in the repository, as shown in **Figure 3-14**. Click ▶ next to the original report in the repository to see all versions of it.

Your saved input control values also appears in a drop-down list when you open the input controls dialog.



Name	Description	Type	Created Date	Modified Date
▾ 01. Geographic Results by Segment Report	Sample HTML5 multi-axis column chart from Domain showing Sales, Units, and $ Per ...	Report	May 26	June 1
Geographic Results Report Filtered for Chicken Prod...		Report Version	Today	Today

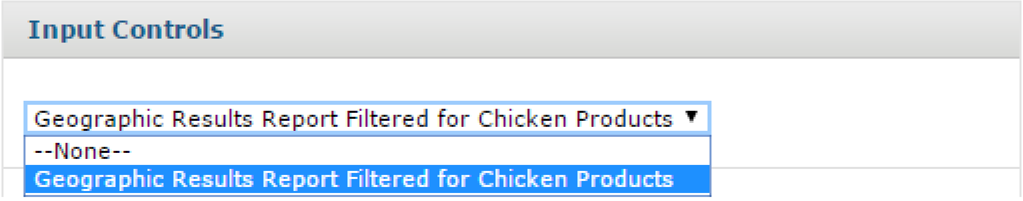
Figure 3-14 Filtered Version of Geographic Results Report in Repository

To save the input control values:

1. In the repository, locate and run the report 1. Geographic Results by Segment Report.
2. On the tool bar, click .
3. Select all of the chicken products, as described in **"Multi-select Input Controls" on page 70**.

4. Click save at the bottom of the dialog box.
5. Enter "Geographic Results Report Filtered for Chicken Products" as a name for the input control values and click **Save**.

   JasperReports Server saves the input control values as an option. A new drop-down box appears at the top of the input control.



Input Controls

Geographic Results Report Filtered for Chicken Products ▼
--None--
Geographic Results Report Filtered for Chicken Products

Figure 3-15 Saved Input Controls Option

6. Select Geographic Results Report Filtered for Chicken Products from the list of options and click **OK**. The report shows data for chicken-related products only.

3.7.4 Cascading Input Controls

Cascading input controls in a report reduce a large number of choices to a manageable number. A single value chosen for a cascading input control determines which other values appear as choices for input. For example, the choice of a country determines which states or regions are listed as choices. For more information, see **"Selecting a Data Source for Running the Complex Report" on page 186**.

To run a report with cascading input controls:

1. In the repository, locate and run the report 16. Interactive Sales Report.

   The report runs, and appears with the **Options** panel open on the left side of the Report Viewer
2. In the Options panel's **Country multi select** drop-down, select a different country, for example Canada.

   The other drop-downs in the Filters panel are automatically updated with Canadian data.

   Cascading input controls are implemented as queries that access the database to retrieve the new values. The server displays an activity monitor while the query is running, and in the case of long queries, you can click Cancel and select different values.
3. In the Product Name section, select a product to display in the report.
4. Click **Apply** to run the report with the chosen values.

3.8 Running a Report Book

Report books are multiple reports bundled into a single object, created in Jaspersoft Studio. You can run and view report books from the Library page or the repository, much like you would a standard report. However, report books contain some elements that are not found in standard reports.

To run a report book in the Report Viewer:

1. In the repository, click the name of the report book you want to run. For example, click the sample **17. Report Workbook**. The report appears, as shown below.

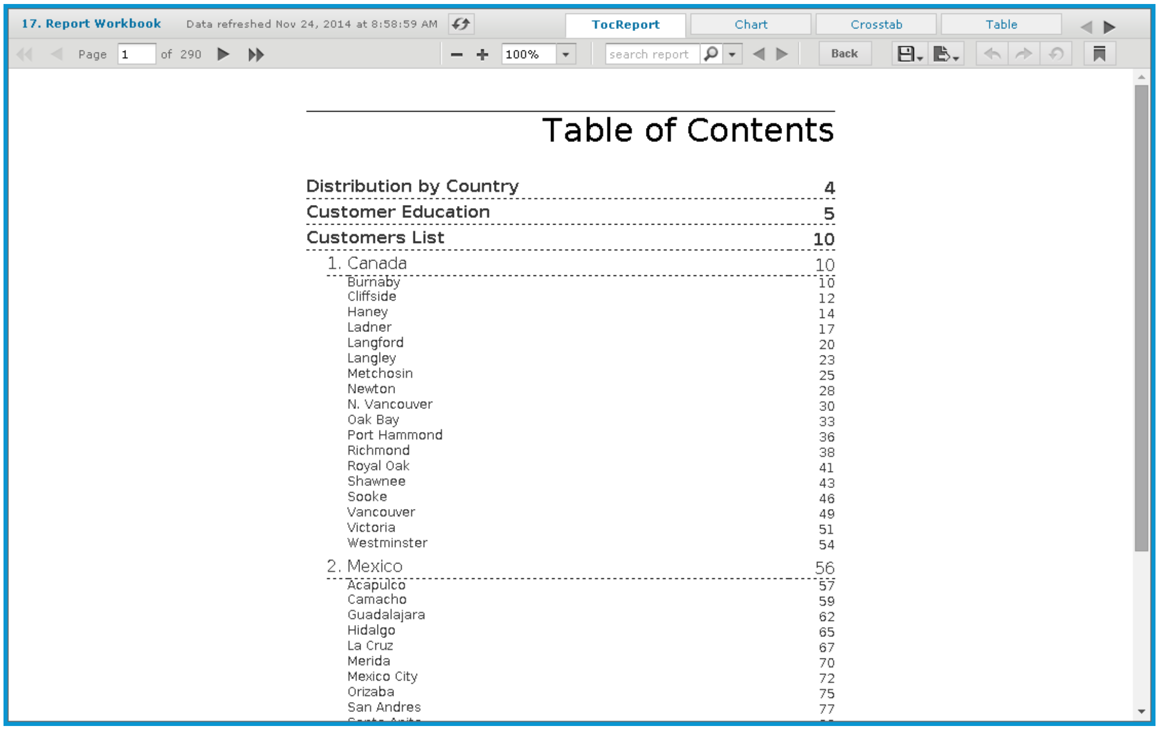


Table of Contents

Distribution by Country	4
Customer Education	5
Customers List	10
1. Canada	10
Burnaby	10
Cliffside	12
Haney	14
Ladner	17
Langford	20
Langley	23
Metchosin	25
Newton	28
N. Vancouver	30
Oak Bay	33
Port Hammond	36
Richmond	38
Royal Oak	41
Shawnee	43
Sooke	46
Vancouver	49
Victoria	51
Westminster	54
2. Mexico	56
Acapulco	57
Camacho	59
Guadalajara	62
Hidalgo	65
La Cruz	67
Merida	70
Mexico City	72
Orizaba	75
San Andres	77

Figure 3-16 Sample report book as seen in the Report Viewer.

The sample report book contains three bundled reports:

- **Distribution by Country**, a chart-type report.
- **Customer Education**, a crosstab-type report.
- **Customers List**, a table-type report.

Each of these reports can be accessed by a tab, along with the Table of Contents page, located at the top of the Report Viewer.

2. Click the **Chart** tab to open the **Customer Distribution by Country** report. Note that you can interact with this report as you would with a standard chart-based report in the viewer.
3. Click the **TocReport** tab to return to the table of contents page.
4. Scroll down and click the **Acapulco** entry in the table of contents. The first page of the Acapulco table entries is displayed.
5. Click to open the bookmarks pane.
6. Click the **Customer Education** entry in the bookmarks pane. The Customer Education crosstab report opens in the viewer.
7. Close the bookmarks pane by clicking the x in the upper right corner of the pane.

Chapter 3 Scheduling Reports and Dashboards

Using the JasperReports Server scheduler, you can run reports and take snapshots of dashboards repeatedly and unattended during off hours or at other times. The reports and dashboards are saved in a file format of your choice to the repository, the host machine, or an FTP server. Scheduling these jobs to run in the background or during off hours can reduce the performance impact on the server.

You can view a list of scheduled reports or dashboards on the **View > Schedules** page.

This chapter contains the following sections:

- **Creating a Schedule**
- **Viewing the List of Scheduled Jobs**
- **Changing Schedules**
- **Pausing a Job**
- **Deleting a Job**
- **Running a Job in the Background**
- **Event Messages**

3.9 Overview of the Scheduler

Using the scheduler, you set up a schedule for generating reports or dashboard with parameters, output options, and notifications:

- Schedule – When to run the scheduled job, and how often
- Parameters – If the report or dashboard was designed with input controls, which parameters the scheduled job will use
- Output Options – The name of the output file, the output format and locale, and where the output file is stored
- Notifications – Email options for sending the output to recipients and for sending administrative messages

The permissions of the user who schedules a job determines the data that is exposed. For example, Gloria only has access to inventory data from the Southeast US region. A report that she schedules only shows data from that region, even when the report is viewed by users in other regions. Other users schedule the report themselves to see the data for their own regions.

Sensitive data could be exposed to unauthorized users if you schedule a job as an administrative user with no data restrictions because the output will contain all requested data in the data source. Any user who receives the output can view all the data regardless of the user's access restrictions.

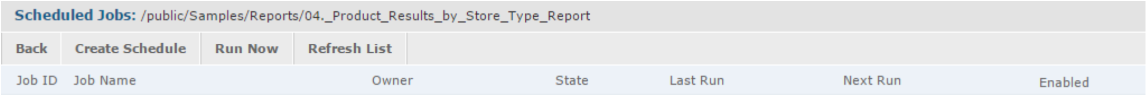
Scheduling a dashboard requires having PhantomJS installed on the computer hosting JasperReports Server. For information on configuring PhantomJS for dashboards, see the System Configuration chapter in the JasperReports Server Administrator Guide.

3.10 Creating a Schedule

To create a schedule:

1. Locate the report or dashboard you want to schedule in the repository.

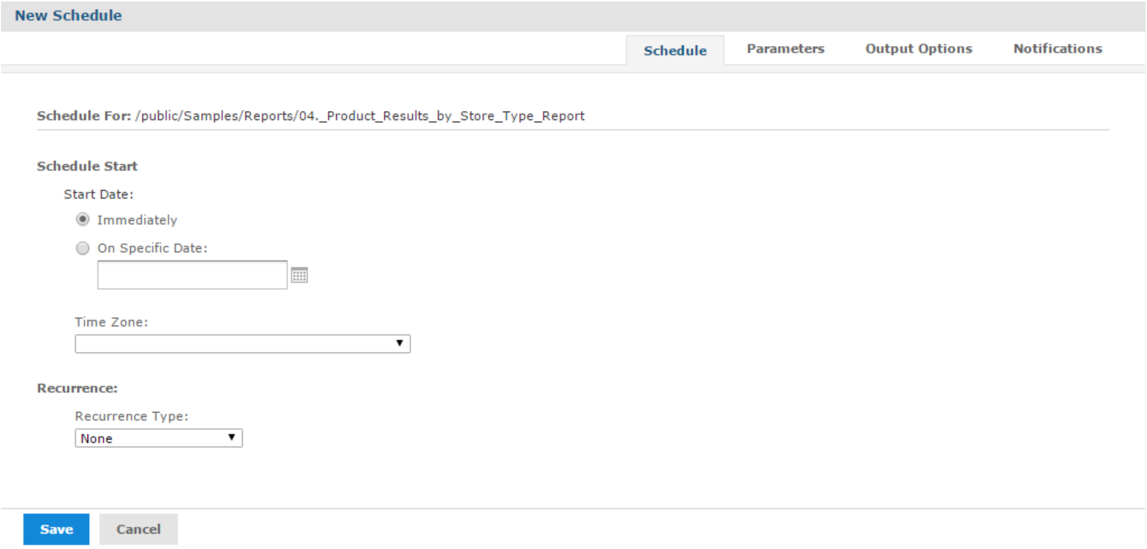
2. Right-click the report or dashboard and select **Schedule...** from the context menu, or if it already has a schedule, click the schedule icon 🕓. The Scheduled Jobs page appears.



Scheduled Jobs: /public/Samples/Reports/04._Product_Results_by_Store_Type_Report

Figure 3-17 Scheduled Jobs Page

3. Click **Create Schedule**. The Schedule tab of the scheduler appears.



New Schedule

Schedule For: /public/Samples/Reports/04._Product_Results_by_Store_Type_Report

Schedule Start

Start Date:

● Immediately

○ On Specific Date:

Time Zone:

Recurrence:

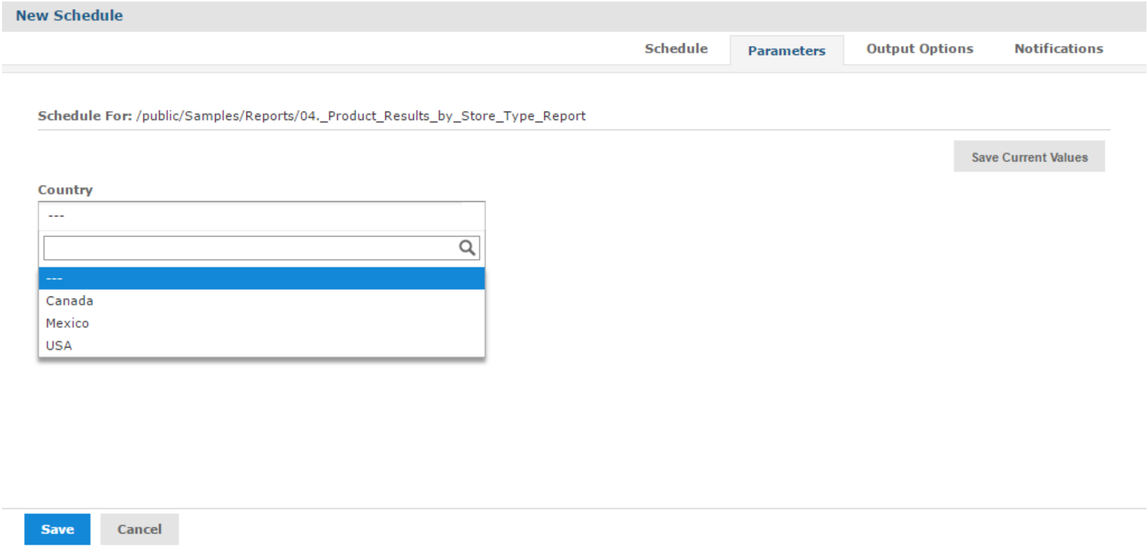
Recurrence Type: None

Figure 3-18 New Schedule Tab

4. Set a start date, choosing whether to run immediately or on a specific date. If a specific date is selected, click the calendar icon 📅 to select a start date and time.
5. Specify the time zone for the schedule. The default time zone is the time zone of the server, the time zone you entered at log in. If you're in a different time zone, set this field accordingly.
6. Choose a recurrence setting, as described in **"Running a Job Repeatedly" on page 82**. If you select Simple or Calendar Recurrence, additional controls appear on the page.
   - **None**: Run the job once.
   - **Simple**: Schedule the job to recur at a regular interval, specified in minutes, hours, days, or weeks.
   - **Calendar**: Schedule the job to recur on days of the week, days of the month, specific dates, or date ranges.

> If you set up a job with simple recurrence to start immediately, the job schedule will change after export/import or a server restart. This happens because the job does not retain the previous run history and therefore starts immediately after import or restart. If you have a large number of scheduled jobs, all scheduled jobs with simple recurrence will attempt to start at the same time after export/import or restart. This can impact server performance. In addition, some scheduled jobs may be locked out and they will continue to try to run.
>
> To ensure a recurring schedule does not change after export/import or restart, either use simple recurrence with a specific start time, or set up calendar recurrence.

7. If the report or dashboard you are scheduling has input controls that prompt for user input, click the Parameters tab.



Schedule For: /public/Samples/Reports/04._Product_Results_by_Store_Type_Report

Figure 3-19 Set the Parameter Values Page for Scheduling a Report

Saved values, if there are any, appear in a drop-down list at the top of the page, as shown in **Figure 3-19**. In the **Use saved values** drop-down, you can set the input controls defined for the report or dashboard you're scheduling. You can set the input values for the scheduled job, and click **Save Current Values** to save the input value as a named set of values.

For more information about using saved values and saving input values, see **"Running a Report with Input Controls or Filters" on page 69**.

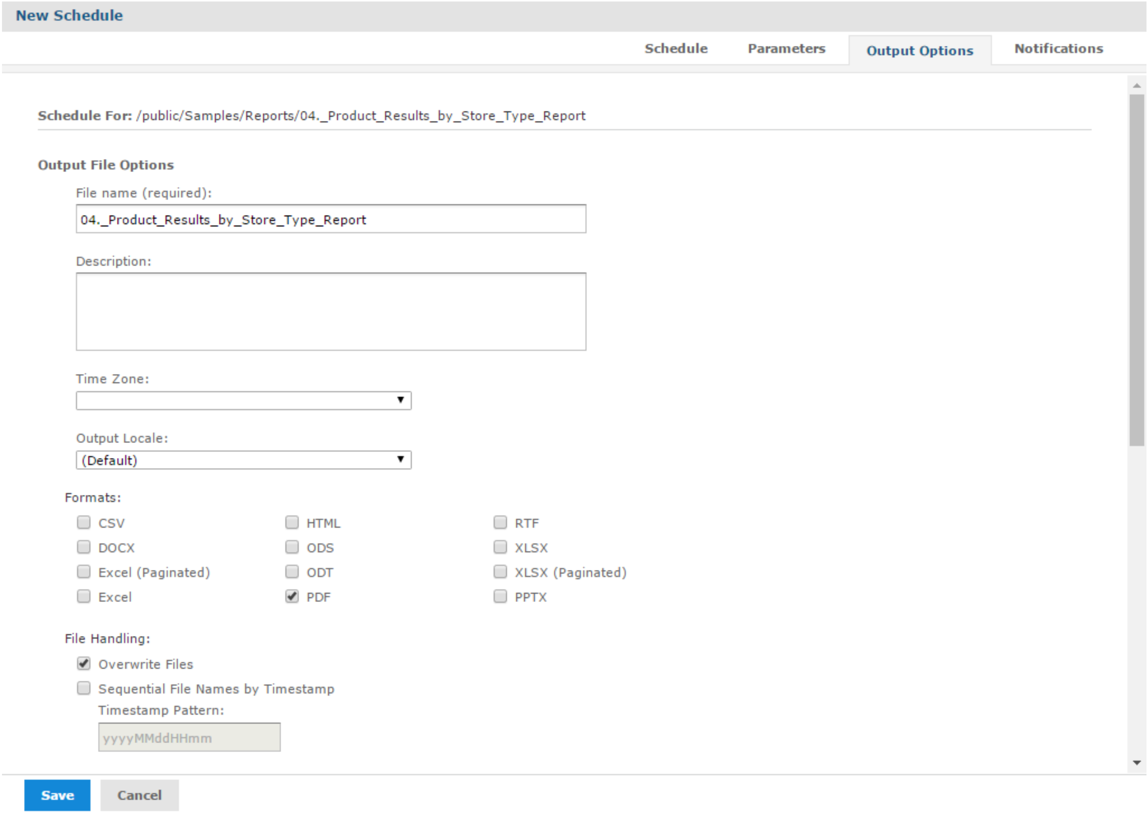
8. Choose a set of saved values, or set the input controls.
9. Click the Output Options tab and set the output format and location, as described in **Setting Output Options**.
10. Click the Notifications tab and set up email notifications, as described in **Setting Up Notifications**
11. Click **Save**. The Save dialog box appears.
12. In the **Scheduled Job Name** field, enter a name for the job, for example, Weekly Report. The description is optional.
13. Click **Save** to save the schedule. The job appears in the list of saved jobs for the report or dashboard.

3.10.1 Setting Output Options

On the Output Options tab, you can change these settings:

- **File name** – The base name for the output file in the repository. This name must be unique; if you attempt to create a schedule with the same file name as an existing output file, you will not be able to save.
- **Description** – The optional description of the file that appears to users who view the repository.
- **Time Zone**– The output time zone for generating the report.
- **Output Locale** – The locale settings for generating the report.

  The report must support locales, for example a report based on a Domain with language bundles (see **"Security and Locale Information for a Domain" on page 274**).
- **Formats** – The available output formats. Select one or more formats; the default format is PDF. When you select more than one, each format is stored as a separate file in the selected output destination.
- **Canvas Size** – Sets the size of the canvas when exporting a dashboard.
- **File Handling** – Select one or more checkboxes to specify handling for multiple output files with the same name:
  - **Overwrite Files** – Overwrites old output files with newer ones of the same name.
  - **Sequential File Names by Timestamp** – Appends a timestamp to the names of files created by the job. Useful for the output of recurring jobs or for time-sensitive reports where the output must be dated. When the timestamp is used, the output filename is `<basename>-<timestamp>.<extension>`. Note that, depending on the frequency of the schedule and the format of the timestamp, it may be possible to have two output files with the same name; in this case, modify the timestamp or use the **Overwrite Files** checkbox to specify the behavior you want.
  - **Timestamp Pattern** – When **Sequential File Names by Timestamp** is selected, a required pattern for the timestamp, based on the `java.text.SimpleDateFormat` is required. Valid patterns for output files can contain only letters, numbers, dashes, underscores, and periods. The default pattern is `yyyyMMddHHmm`, for example `200906150601`.
  - For more information about the valid patterns for this field, refer to: http://download.oracle.com/javase/6/docs/api/java/text/SimpleDateFormat.html



Schedule For: /public/Samples/Reports/04._Product_Results_by_Store_Type_Report

Output File Options

File name (required): 04._Product_Results_by_Store_Type_Report

Description:

Time Zone:

Output Locale: (Default)

Formats:

□ CSV	□ HTML	□ RTF
□ DOCX	□ ODS	□ XLSX
□ Excel (Paginated)	□ ODT	□ XLSX (Paginated)
□ Excel	☑ PDF	□ PPTX

File Handling:

☑ Overwrite Files

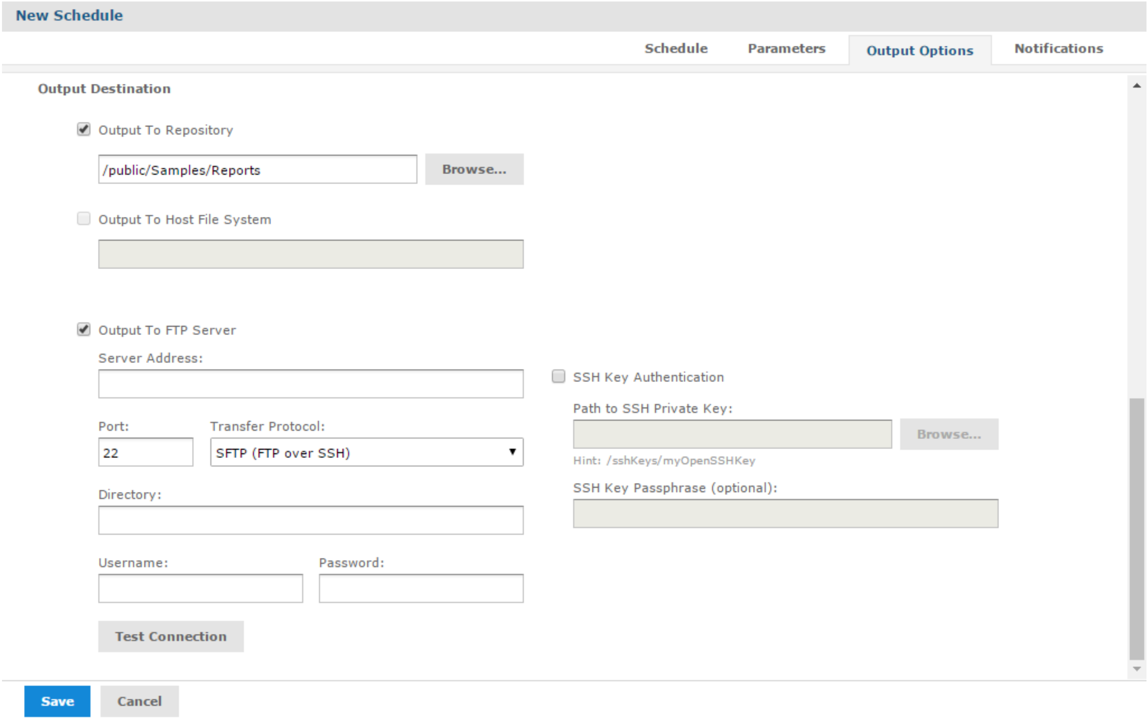
□ Sequential File Names by Timestamp

Timestamp Pattern: yyyyMMddHHmm

Figure 3-20 Output Page for Scheduling a Report – Output File Options

- **Output Destination** – To save the output file, select one or more checkboxes to specify the output location. If you do not want to save the report output (for example, if you only want to email the report) leave all checkboxes blank.
  - **Output To Repository** – If checked, saves the report output to the specified location in the repository. You must have write permission to the output folder. JasperReports Server validates the path when you click **Save** and will display an error message if the location doesn't exist. You can change the repository location that appears here by default. See JasperReports Server Administrator Guide for more information.
  - **Output To Host File System** – If checked, saves the report output to the specified folder on the Server host machine; check with your administrator for the correct location to enter. **Output To Host File System** must be configured by an administrator; if the check box is grayed out, saving to the host file system has not been enabled for your system. See the *JasperReports Server Administrator Guide* for more information
  - **Output To FTP Server** – If checked, saves the report output to the specified FTP server. You must have write permission to the selected directory on the FTP server. Enter the following properties of your FTP server:
    - **Server Address** – The host, IP address or URL of the FTP server.
    - **Port** – Specifies the FTP connection port. For FTP, the default port is 21; for FTPS, the default port is 990; for SFTP, the default port is 22.
    - **Transfer Protocol** – Specifies the file transfer protocol that the server uses. You can select one of the following:

  - **FTP**.
  - **FTPS (FTP over SSL)**.
  - **SFTP (FTP over SSH)**. If selected, the **SSH Key Authentication** checkbox and SSH private key fields appear.
- **SSH Key Authentication**. Check if the server requires an SSH private key for SFTP transfers.
- **Path to SSH Private Key**. When **SSH Key Authentication** is checked, the location of the SSH private key in the repository is required.
- **SSH Key Passphrase**. When **SSH Key Authentication** is checked, enter the passphrase for the SSH private key, if it has one.
- **Directory** – The directory on the FTP server where the report output is saved.
- **Username** – The username for access to the FTP server.
- **Password** – The password for the username.



☑ Output To Repository

/public/Samples/Reports

□ Output To Host File System

☑ Output To FTP Server

Server Address:

Port: 22

Transfer Protocol: SFTP (FTP over SSH)

Directory:

Username:

Password:

□ SSH Key Authentication

Path to SSH Private Key:

Hint: /sshKeys/myOpenSSHKey

SSH Key Passphrase (optional):

Figure 3-21 Output Page for Scheduling a Report – Output Destination

When you click **Save**, the job appears in the list of scheduled jobs.

3.10.2 Setting Up Notifications

On the Notifications page, you can set up email notifications to the recipients of the report or dashboard and to administrators.

Notifications are sent to all users with the organization ROLE_ADMINISTRATOR role, who belong to the same organization as the user who creates the scheduled job.

Send report when scheduler runs:

Enter one or more email addresses to send the output after the job has run. You can configure the following:

- **To** – One or more email addresses separated by commas for sending email notification.

By default, the mail server is not configured by the JasperReports Server installer. To send email notifications, the administrator must configure the mail server, as described in the *JasperReports Server Installation Guide*.

- **CC** – One or more email addresses separated by commas for sending an email notification on the CC line.
- **BCC** – One or more email addresses separated by commas for sending blind carbon-copy email notification; the addresses in this field are not revealed to the other recipients.
- **Subject** – The subject line of the notification email.
- **Message** – Content of the notification email.
- Choose one of the radio buttons to specify how email recipients access the output:
  - **Include reports/dashboards as repository links in email body** – Sends a link to the output in the repository. Not available unless **Output to Repository** is selected on the Output Options tab.
  - **Include report/dashboard files as attachments** – Sends the output as attachments to the notification email. If you have selected multiple output formats, each one is attached as a separate file to the email notification.
  - **Include report/dashboard files as ZIP attachment** – Zips all outputs into a single archive file before attaching to the email.
  - **Include HTML report in email body** – Displays the report directly in the email body. This option is available only when **Include report files as attachments** or **Include report files as ZIP attachment** is checked and HTML is selected as one of the options on the Output File Options page. When this option is selected, you cannot include a message in the email body. This option is not available for dashboards.

Be careful when sending reports containing sensitive data by email, either in the email body or as an attachment.

- **Do not send emails for empty reports** – A check box option that, if checked, prevents the server from attaching empty report output files to email notifications. This applies to parametrized reports where there is no data that matches the parameters. This option is not available for dashboards.

Send job status notifications:

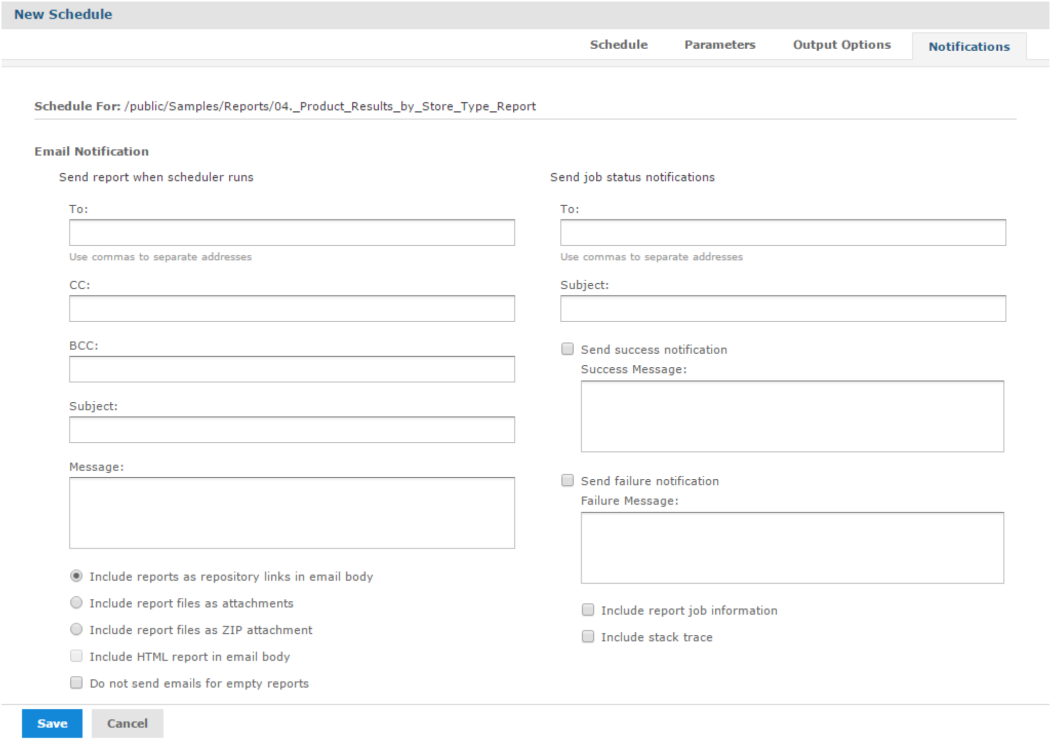
Enter one or more email addresses to send notification of job success or failure to administrators:

- **To** – One or more email addresses separated by commas for sending email notification.

By default, the mail server is not configured by the JasperReports Server installer. To send email notifications, the administrator must configure the mail server, as described in the *JasperReports Server Installation Guide*.

- **Subject** – The subject line of the notification email.
- **Send success notification** – Check box option that, when checked, sends a notification when the scheduled job runs.
  - **Success Message** – The message in the body of the notification email sent on success.

- **Send failure notification** – Check box option that, when checked, sends a notification when the scheduled job fails to run.
  - **Failure Message** – The message in the body of the notification email sent on failure.
- **Include report/dashboard job information** – A check box option that, if selected, includes the report or dashboard's label, ID, description, and job status in the notification email.
- **Include stack trace** – A check box option that, if selected, includes the stack trace for failed jobs in the body of the email.



Schedule For: /public/Samples/Reports/04._Product_Results_by_Store_Type_Report

Email Notification

Send report when scheduler runs

To:

Use commas to separate addresses

CC:

BCC:

Subject:

Message:

● Include reports as repository links in email body

○ Include report files as attachments

○ Include report files as ZIP attachment

□ Include HTML report in email body

□ Do not send emails for empty reports

Send job status notifications

To:

Use commas to separate addresses

Subject:

□ Send success notification

Success Message:

□ Send failure notification

Failure Message:

□ Include report job information

□ Include stack trace

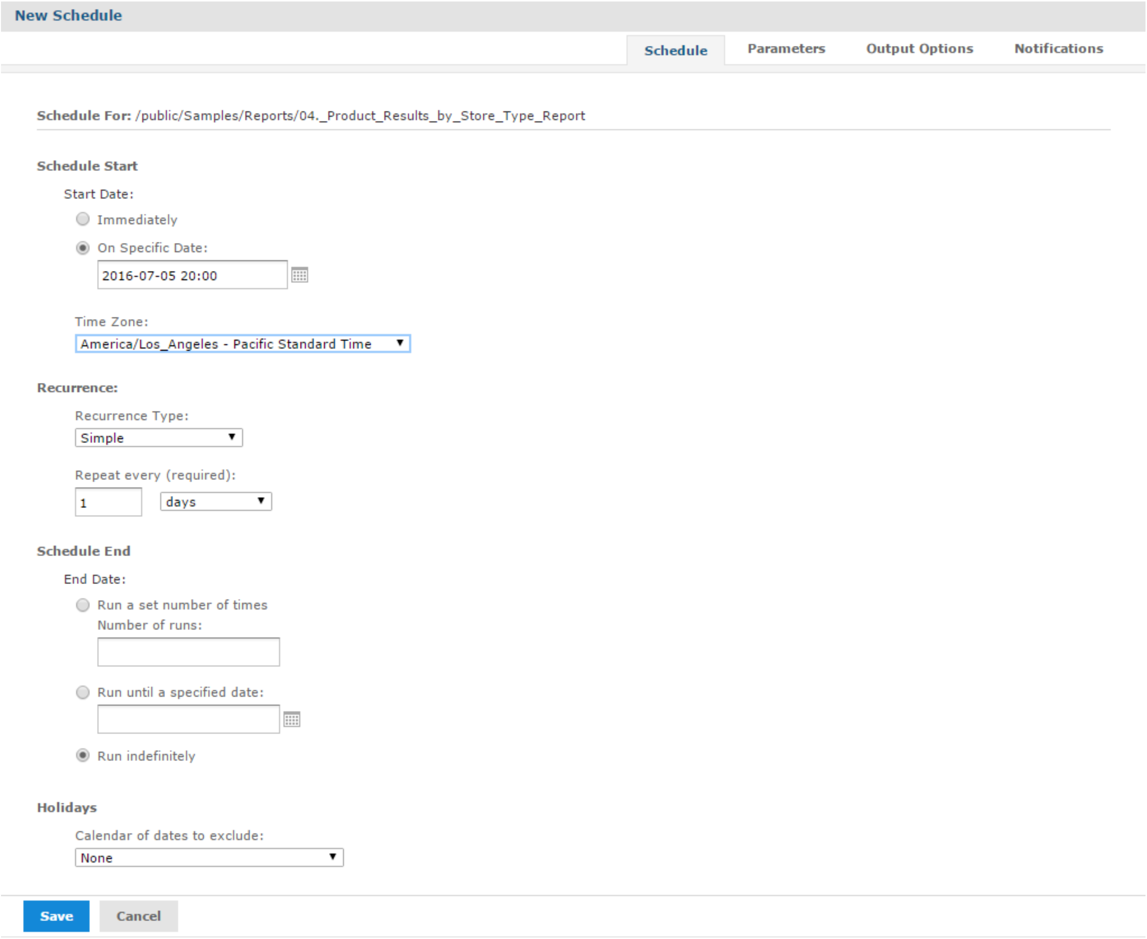
Figure 3-22 Notifications Page for Scheduling a Report

3.10.3 Running a Job Repeatedly

To run jobs automatically on a regular basis, select simple or calendar recurrence on the Schedule tab:

- **Simple** recurrence repeatedly runs the job at a regular interval set in minutes, hours, days, or weeks.
- **Calendar** recurrence involves more settings: time of day, days of the week, or days of the month, and months of the year.

In **Figure 3-23** you can see an example of how to set simple recurrence.



New Schedule

Schedule For: /public/Samples/Reports/04._Product_Results_by_Store_Type_Report

Schedule Start

Start Date:

○ Immediately

● On Specific Date:

2016-07-05 20:00

Time Zone:

America/Los_Angeles - Pacific Standard Time

Recurrence:

Recurrence Type:

Simple

Repeat every (required):

1 days

Schedule End

End Date:

○ Run a set number of times

Number of runs:

○ Run until a specified date:

● Run indefinitely

Holidays

Calendar of dates to exclude:

None

Figure 3-23 Simple Recurrence Settings

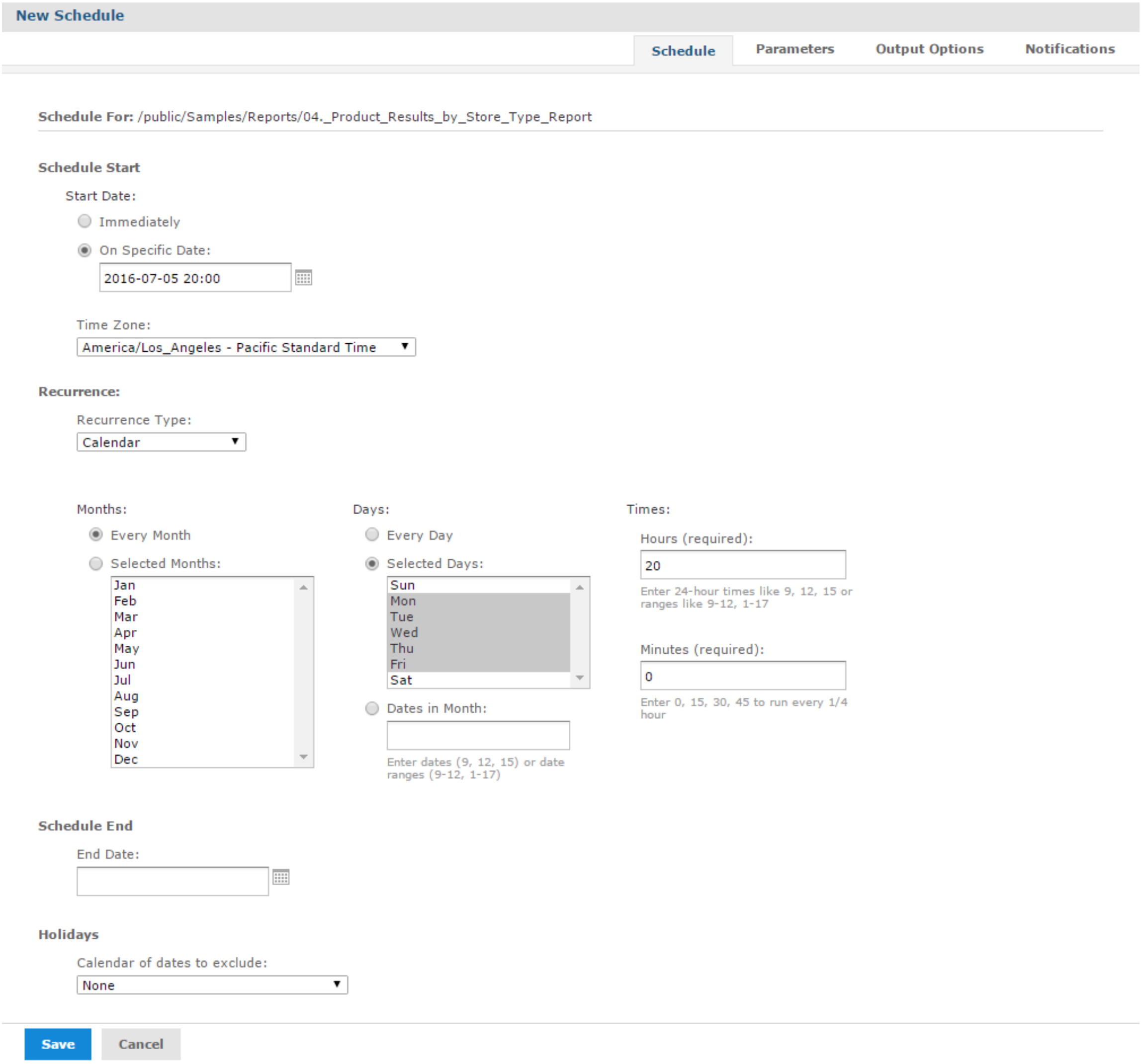
Simple recurrence options are:

- **Repeat every** – The interval between jobs, in minutes, hours, days, or weeks.
- **Run a set number of times** – Runs the specified number of times.
- **Run until a specified date** – Runs until a calendar date is reached. Click the calendar icon, ▦, to select the date.
- **Run indefinitely** – Runs at the specified times until you delete the job.
- **Holidays** – A holiday calendar specifies a list of days when the scheduled job will not run. To use a holiday calendar, select it from the drop-down list. Only one holiday calendar can be selected at a time. Holiday calendars are configured by an administrator; if no calendars are available in this list, this option has not been configured for your system.

If your server recognizes Daylight Savings Time (DST), jobs scheduled using simple recurrence may seem to occur one hour later (when DST ends) or one hour earlier (when DST begins). If you want jobs to recur at the same time of day and respect DST adjustments, use calendar recurrence.

If you set up a job with simple recurrence to start immediately, the job schedule will change after export/import. This happens because the imported job does not retain the previous run history and therefore starts immediately after successful import. To ensure a recurring schedule does not change after export/import, either use simple recurrence with a specific start time, or set up calendar recurrence.

In **Figure 3-24** you'll see an example of calendar recurrence settings.



Schedule For: /public/Samples/Reports/04._Product_Results_by_Store_Type_Report

Schedule Start

Start Date:

○ Immediately

● On Specific Date: 2016-07-05 20:00

Time Zone: America/Los_Angeles - Pacific Standard Time

Recurrence:

Recurrence Type: Calendar

Months:

● Every Month

○ Selected Months: Jan, Feb, Mar, Apr, May, Jun, Jul, Aug, Sep, Oct, Nov, Dec

Days:

○ Every Day

● Selected Days: Sun, Mon, Tue, Wed, Thu, Fri, Sat (Mon, Tue, Wed, Thu, Fri selected)

○ Dates in Month:

Enter dates (9, 12, 15) or date ranges (9-12, 1-17)

Times:

Hours (required): 20

Enter 24-hour times like 9, 12, 15 or ranges like 9-12, 1-17

Minutes (required): 0

Enter 0, 15, 30, 45 to run every 1/4 hour

Schedule End

End Date:

Holidays

Calendar of dates to exclude: None

Figure 3-24 Calendar Recurrence Settings

Calendar recurrence options are:

- **Months** – The months during which the job runs.
  - Every Month
  - Selected Months

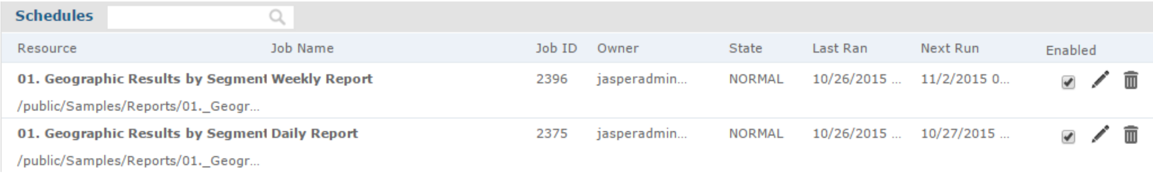
- **Days** – The days when the job runs.
  - Every Day
  - Selected Days
  - Dates in Months – Enter dates or date ranges separated by commas, for example: `1, 15`.
- **Times** – The time of day in minutes and hours when the job should run. The hours use 24-hour format. You can also enter multiple minutes or hours, and ranges, separated by commas. For example, entering `0,15,30,45` for the minutes, and `9-17` for the hours, runs the report every 15 minutes from 9:00 a.m. to 5:45 p.m. Enter an asterisk (`*`) to run the job every minute or every hour.
- **End Date** – Calendar recurrence runs until a calendar date is reached. Click ▦ to select the date.
- **Holidays** – A holiday calendar specifies a list of days when the scheduled report will not run. To use a holiday calendar, select it from the drop-down list. Only one holiday calendar can be selected at a time. Holiday calendars are configured by an administrator; if no calendars are available in this list, this option has not been configured for your system.

  Administrators see the chapter on scheduling in the *JasperReports Server Web Services Guide* for more information on configuring calendars.

3.11 Viewing the List of Scheduled Jobs

View a list of all scheduled jobs:

All scheduled jobs that the user has defined appear on the Click **View > Schedules** page.



Schedules

Resource	Job Name	Job ID	Owner	State	Last Ran	Next Run	Enabled
01. Geographic Results by Segment /public/Samples/Reports/01._Geogr...	**Weekly Report**	2396	jasperadmin...	NORMAL	10/26/2015 ...	11/2/2015 0...	☑
01. Geographic Results by Segment /public/Samples/Reports/01._Geogr...	**Daily Report**	2375	jasperadmin...	NORMAL	10/26/2015 ...	10/27/2015 ...	☑

Figure 3-25 The Schedules Page

Typical users see only the jobs they have defined; administrators see the jobs defined by all users. In **Figure 3-25**, jasperadmin has scheduled two jobs for the Geographic Results by Segment report.

You can use the Schedules page's search field to find the scheduled job you want.

The Schedules page shows the name of the scheduled report or dashboard, the repository URI of the job, the internal ID number of the job, the user (owner) who created the job, and the state of the job. Job states are:

- **NORMAL** – The job is scheduled.
- **EXECUTING** – The server is generating the output.
- **COMPLETE** – The server has finished running the job and placed output to the repository.
- **PAUSED**– The job has been disabled. Click **Enabled** to resume the schedule.
- **ERROR** – The scheduler encountered an error while scheduling or triggering the job. This doesn’t include cases where the job is successfully triggered, but an error occurs while it runs.
- **UNKNOWN** – The scheduler encountered an error with the job trigger.

The Schedules page includes these controls:

- **Enabled** checkbox – When checked, the job will run at the scheduled times. When unchecked, the job is paused.

- ✎ – Edits the schedule.
- 🗑 – Deletes the scheduled job.

  When the server receives a request to delete a job that is running, the server completes running the job before deleting it.

View scheduled jobs for an individual report or dashboard:

Scheduled jobs appear in the repository with a 🕓 icon beside the report or dashboard's name. To view the list of scheduled jobs for a report or dashboard, locate it in the repository, click on the 🕓 icon or right-click the report or dashboard, and select **Schedule** from the context menu. The Scheduled Jobs page appears. The Scheduled Jobs display information that's similar to the Schedules page.

You can filter search results for scheduled jobs in the repository using the following options: **Any schedule** (scheduled and unscheduled reports), **Scheduled**, **Scheduled by me**, or **Not scheduled**. See **1.5.2, "Filtering Search Results," on page 18** for more information.

Buttons on the Scheduled Jobs page include:

Button	Description
Back	Returns to the repository.
Create Schedule	Opens the Schedule tab to define a new job.
Run Now	Opens the scheduler and allows you to run the job immediately. See **"Running a Job in the Background" on page 87**.
Refresh List	Refreshes the list of jobs, for example to see if a job has finished running.

3.12 Changing Schedules

If the start date for a schedule has not yet passed, you can edit the schedule. Once the start date for a schedule has passed, create a new schedule rather than changing the start date.

To edit a schedule:

1. Click **View > Schedules**.
2. Click ✎ in the row of the job you want to change.
3. Make the changes on the Schedule, Parameters, Output, and Notifications pages.
4. Click **Save**. The update occurs immediately.

3.13 Pausing a Job

To stop a job from running without deleting it, disable the job.

To pause a scheduled job:

1. Click **View > Schedules**.
2. In the row of the job you want to stop, uncheck **Enabled**.

To resume a paused job:

1. Click **View > Schedules**.
2. In the row of the job you want to resume, check **Enabled**. When a stopped job is re-enabled, it waits until the next scheduled time to run.

3.14 Deleting a Job

To delete a scheduled job:

1. Click **View > Schedules**.
2. In the row of the job you want to delete, click 🗑.

3.15 Running a Job in the Background

Running a job in the background generates a report or exports a dashboard without interrupting your workflow. You can keep working in the server as the job runs. When the job completes, you can export the job directly to any format and save it in the repository. You can the share the output with others by sending it by email.

Running a job in the background is equivalent to scheduling it to run immediately without recurrence.

To run a job in the background:

1. Locate the report or dashboard you want in the repository.
2. Right-click the report or dashboard and select **Run in Background** from the context menu.
3. Set the output format and location, as described in **Setting Output Options**. By default, the output is saved in the repository.
4. (Optional) If the job you are running has input controls that prompt for user input, click the Parameters tab. Choose a set of saved values, or set the fields one at a time.
5. (Optional) Click the Notifications tab and set up email notifications, as described in **Setting Up Notifications**
6. Click **Save**. The job begins to run immediately.

3.16 Event Messages

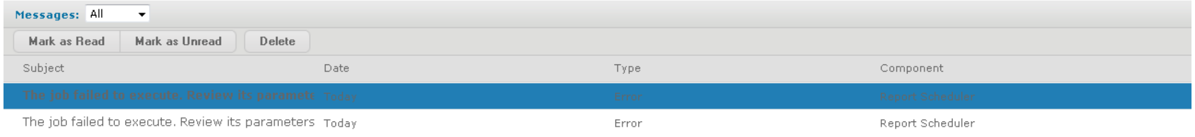
When an event occurs (for example, a scheduled report returns errors), JasperReports Server sends the owner of the job a notification message. You can browse these messages to troubleshoot report scheduling problems in the server. For example, you can determine that a job fails because its data source configuration uses incorrect credentials.

A common cause of the error message indicating that the job failed to execute is an incorrectly configured mail server. The mail server must be manually configured after installation in order for users to send email notifications.

The Messages page displays the list of events logged for the current user.

To open the Messages page:

1. On any page, click **View > Messages**. The Messages page appears.
2. To view a message, click its name. The message opens in the Message Detail page.
3. To activate the buttons on the Messages page, click in a blank area of the message row that you want to manage. The buttons become enabled.



Subject	Date	Type	Component
The job failed to execute. Review its paramets	Today	Error	Report Scheduler
The job failed to execute. Review its parameters	Today	Error	Report Scheduler

Figure 3-26 **Message Management Buttons**

4. Use the buttons on the Messages page to manage the list of messages.

CHAPTER 4 WORKING WITH THE AD HOC EDITOR

This section describes functionality that can be restricted by the software license for JasperReports Server. If you don't see some of the options described in this section, your license may prohibit you from using them. To find out what you're licensed to use, or to upgrade your license, contact Jaspersoft.

The Ad Hoc Editor is the interactive designer for creating and editing an Ad Hoc view, where you can explore and analyze data from your Topic, Domain, or OLAP data source. Ad Hoc views can also be used to create content for reports.

This chapter discusses the Ad Hoc Editor and Ad Hoc views, and includes the following sections:

- **Overview of the Ad Hoc Editor**
- **Working with Tables**
- **Working with Charts**
- **Working with Standard Crosstabs**
- **Working with OLAP Connection-based Crosstabs**
- **Calculated Fields and Measures**
- **Using Filters and Input Controls**
- **Creating a View from a Domain**
- **Working with Topics**

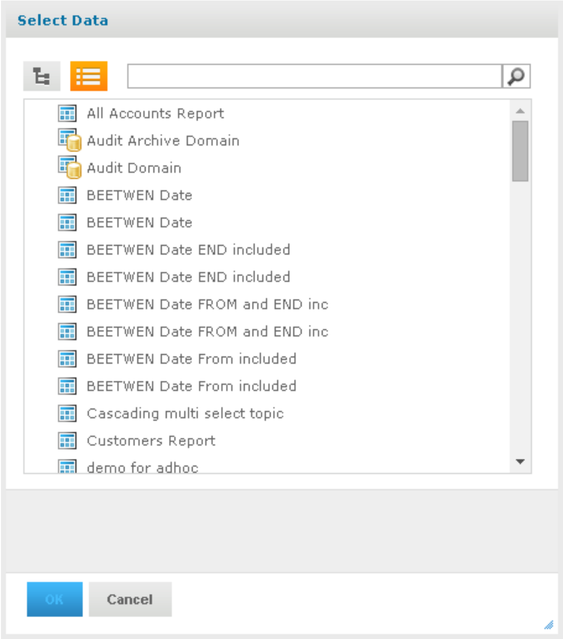
After you create an Ad Hoc view, you - and other users with the proper permissions - can run the report, then further refine the displayed information and personalize the look of the report in the report viewer. For more information on that process, see **"Running Reports and the Report Viewer" on page 53**.

4.1 Overview of the Ad Hoc Editor

The Ad Hoc Editor supports the creation of views for various types of reports: tables, crosstabs, and charts. You intuitively interact with the editor to create these views by simply dragging and dropping elements. You can add and summarize fields, define groups, label and title the report, and format data for each field. You can also use the editor to explore and analyze data interactively.

To open the Ad Hoc Editor:

1. Click **Create > Ad Hoc View**. This opens the **Select Data** dialog.
2. Choose your data source from the list and click **OK**.



All Accounts Report
Audit Archive Domain
Audit Domain
BEETWEN Date
BEETWEN Date
BEETWEN Date END included
BEETWEN Date END included
BEETWEN Date FROM and END inc
BEETWEN Date FROM and END inc
BEETWEN Date From included
BEETWEN Date From included
Cascading multi select topic
Customers Report
demo for adhoc

Figure 4-1 List of Data Sources in the Select Data Dialog

The Ad Hoc Editor contains the following panels, from left to right:

- **Data Source Selection,** which contains the fields, dimensions, and measures available in the source Domain, Topic, or OLAP connection.
- **Ad Hoc View**, the main view design panel.
- **Filters**, which defines a subset of data to retrieve from the data source.

We'll discuss how to use these panels to create an Ad Hoc view later in this section.

4.1.1 Ad Hoc Sources: Topics, Domains, and OLAP Connections

The following repository objects provide a prepared connection to a data source for Ad Hoc view creation:

- **Topics**, JRMXL files created externally and uploaded to JasperReports Server as a basis for Ad Hoc views.
- **Domains**, virtual views of a data source that present the data in business terms, allow for localization, and provide data-level security.
- **OLAP Connections**, multi-dimensional views of data that allow users to analyze a large number of aggregate data levels.

You can also open and edit an existing Ad Hoc view to create a new Ad Hoc view.

4.1.1.1 Topics

Generally, an administrator or Jaspersoft Studio user creates a Topic as a JRXML file. The JRXML topic is then associated with a data source in the server. A Topic can also be created from a Domain in the server. Both types of topics appear in the Select Data wizard when you create an Ad Hoc view.

Using a Topic as your source generates an empty view, which allows you to begin adding data to your view right away, without choosing, pre-filtering, or changing display names of the data (all of which are required steps when creating a Domain-based view).

To begin designing a Topic-based view:

1. Launch the Ad Hoc Editor by clicking **Create > Ad Hoc View.**
2. In the Select Data wizard, click and navigate to **Ad Hoc Components > Topics**.
3. Expand the Topics folder and select a topic.
4. Select the type of view you intend to create: table, chart, or crosstab. For an overview of view types, see **"Ad Hoc View Types" on page 92**.

You can now begin working on your view in the Ad Hoc Editor.

4.1.1.2 Domains

Administrators create Domains that typically filter the data, create input controls, and manage the list of available fields and measures. A Domain specifies tables in the database, join clauses, calculated fields, display names, and default properties, all of which define items and sets of items for creating Ad Hoc views.

To begin designing a Domain-based view:

1. Launch the Ad Hoc Editor by clicking **Create > Ad Hoc View.**
2. In the Select Data wizard, click and navigate to **Domains**.
3. Expand the Domains folder and select a domain.
4. Click **Choose Data...**, and click the options on the left of the window to perform the following tasks:
   - Click **Fields** to select fields of data to use in the view.
   - Click **Pre-filters** to create filters to limit the data available in the Ad Hoc Editor.
   - Click **Display** to change the fields' display names.
   - Click **Save as Topic** to save the customized topic for later use.
5. Select the type of view you want to create: table, chart, or crosstab. For an overview of view types, see **"Ad Hoc View Types" on page 92**.

You can now begin working on your view in the Ad Hoc Editor.

4.1.1.3 OLAP Connections

Administrators create OLAP client connections that expose transactional data and define how the data can be seen as a multidimensional cube. An OLAP connection can expose multiple cubes in a single OLAP connection.

With OLAP connections, you can create chart and crosstab views only.

To begin designing an OLAP connection-based view:

1. Launch the Ad Hoc Editor by clicking **Create > Ad Hoc View.**
2. In the Select Data wizard, click and navigate to **Analysis Components > Analysis Connections** and select a sample project and a connection.

You can now begin working on your view in the Ad Hoc Editor.

For more information about OLAP-based view functionality, refer to the following sections:

- **"Working with OLAP Connection-based Crosstabs" on page 121**
- **"Working with Topics" on page 157**

4.1.2 Ad Hoc View Types

The Ad Hoc Editor allows you to select from three view types:

- **Tables**, which are used to view values in the database and to summarize the values in columns.
- **Charts**, which compare one or more measures across multiple sets of related fields.
- **Crosstabs**, which aggregate data across multiple dimensions.

This section provides an overview of each view type. The design and content tasks for working with each type of view are discussed in more detail in the following sections:

- For more information on table views, see **Working with Tables**.
- For more information on chart views, see **"Working with Charts" on page 103**.
- For more information on crosstab views, see **"Working with Standard Crosstabs" on page 118**.

4.1.2.1 Tables

The architecture of a table view consists of columns, rows, and groups.

Columns in a table correspond to the columns in the data source. They are included by adding fields or measures to the table in the Ad Hoc view.

Rows correspond to rows in the database. The information in each row depends on what columns are included in the table.

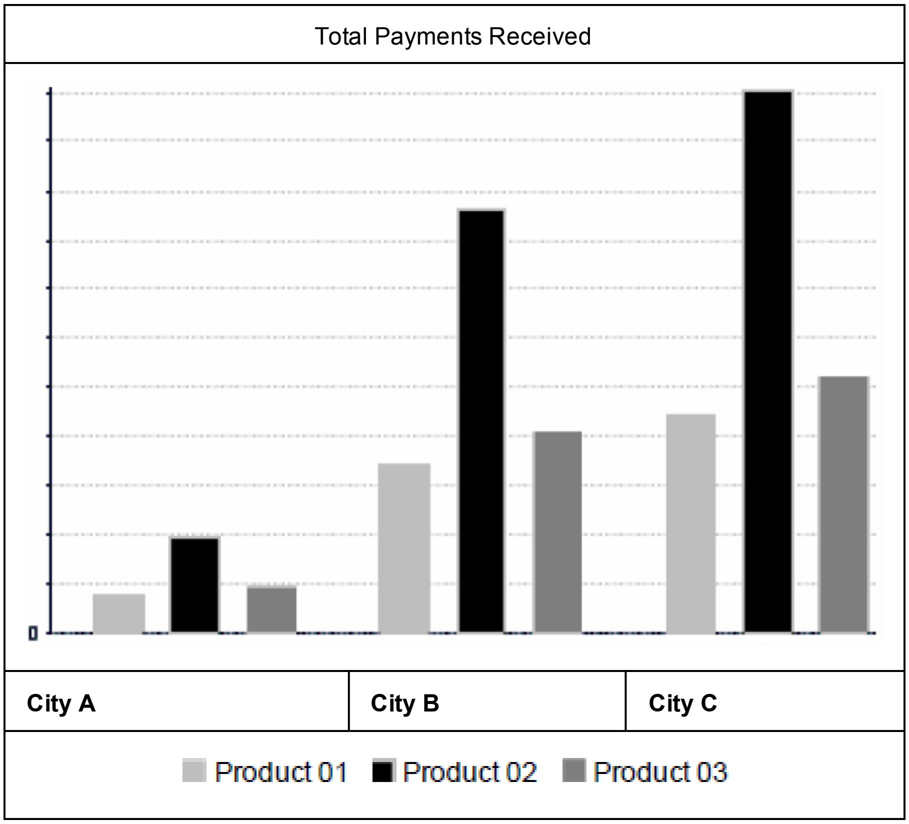
Using groups, rows can be grouped by identical values in any field with intermediate summaries for each grouped value. For example, a table view of product orders might contain columns to show the dates and amounts of each order, and its rows might be grouped by city and product.

	Date Placed	**Date Filled**	**Payment Received**
City A			
Product 01			
	Date	Date	Amount
	Date	Date	Amount
Product 01 totals:		**Count**	**Sum**
Product 02			
	Date	Date	Amount
	Date	Date	Amount
Product 02 totals:		**Count**	**Sum**
Product 03			
	Date	Date	Amount
	Date	Date	Amount
Product 03 totals:		**Count**	**Sum**
City A totals:		**Count**	**Sum**
City B			
Product 01			
	Date	Date	Amount
	Date	Date	Amount
	Date	Date	Amount
	...	...	...

For more information on working with tables, see **"Working with Tables" on page 98**.

4.1.2.2 Charts

Charts summarize data graphically. Types of charts include bar chart, line chart, and pie chart, among others. With the exception of time series and scatter charts, each type of chart compares summarized values for a group. For example, the **Chart** tab might show the data in a bar chart that compared the sum of Payments Received for each of the products in each of the cities.



Time series and scatter charts use time intervals to group data.

For more information on working with charts, see **"Working with Charts" on page 103**.

4.1.2.3 Crosstabs

Crosstabs are more compact representations than tables; they show only aggregate values, rather than individual database values. Columns and rows specify the dimensions for grouping; cells contain the summarized measurements. For instance, the example above could be displayed in a crosstab with columns grouped by sales manager and year.

		Tom			**Harriet**			**Manager Totals**
		2012	**2013**	**Year Totals**	**2012**	**2013**	**Year Totals**	
City A	**Product 01**	Payment Received	Payment Received	**Payment Received**	Payment Received	Payment Received	**Payment Received**	**Payment Received**
	Product 02	Payment Received	Payment Received	**Payment Received**	Payment Received	Payment Received	**Payment Received**	**Payment Received**
	Product 03	Payment Received	Payment Received	**Payment Received**	Payment Received	Payment Received	**Payment Received**	**Payment Received**
	Product Totals	**Payment Received**	**Payment Received**	**Payment Received**	**Payment Received**	**Payment Received**	**Payment Received**	**Payment Received**
City B	**Product 01**	Payment Received	Payment Received	**Payment Received**	Payment Received	Payment Received	**Payment Received**	**Payment Received**
	...	...	...	**...**	...	...	**...**	**...**
City Totals		**Payment Received**	**Payment Received**	**Payment Received**	**Payment Received**	**Payment Received**	**Payment Received**	**Payment Received**

For more information on working with crosstabs, see **“Working with Standard Crosstabs” on page 118**.

OLAP connection-based crosstabs behave differently than those created from Topics or Domains. See **“Working with Topics” on page 157**.

4.1.3 The Data Source Selection Panel

The Data Source Selection panel contains a list of available fields in the chosen Topic or Domain. If you are using a Domain, fields may appear in nested sets. Use the arrow beside the set name to expand or collapse a set of fields.

Available fields may be divided into two sections in the panel, **Fields** and **Measures**. You can use the search field in each section to locate a specific field or measure.

To hide this panel, click the icon in the top left corner; this is helpful when arranging content in a large Ad Hoc view. Click the same icon on the minimized panel to expand it.

For more information on working with fields, see **“Using Fields in Tables ” on page 98**, **“Using Fields and Measures in Charts” on page 103**, and **“Using Fields in Crosstabs” on page 118**.

4.1.4 The Ad Hoc View Panel

The Ad Hoc View panel provides tools that allow you to control what data is included in a view, and how it is organized.

Along the top of the panel, there is a tool bar and two drop-down menus.

The first drop-down menu allows you to reset the view type. Use this menu to select **Table, Chart,** or **Crosstab**. You can change the view type in the Ad Hoc Editor during the design or when you reopen the design for editing.

The next drop-down menu contains options for displaying a subset of the available data (**Sample Data**), all available data (**Full Data**), or none of the available data (**No Data**) in the view. Using the sample data can make the design process quicker by loading less data. Use the subset for initial design; use the full set for refining layout elements such as column width. This drop-down menu is not available for charts, which use all available data.

By default, the editor displays only a smaller, sample set of the data in the table. Use the drop-down menu to select **Full Data** to view the full set of data.

Depending on its configuration, JasperReports Server may load a Topic, Domain, or OLAP connection's entire result set into memory when you edit the view, or run a report from it. If the data policies and other options that control JasperReports Server's memory are disabled, ensure that each Topic, Domain, or OLAP connection returns a manageable amount of data, given the environment's load capacity. Alternately, you can change the server's configuration.

The tool bar at the top of the panel provides access to many functions of the Ad Hoc Editor. The toolbar is described in **Table 4-1 on page 96**.

Table 4-1 Ad Hoc Editor Tool Bar Icons

Icon	Name	Description
👁	Display Mode	Click this icon to hide the editor interface. This mode provides a subset of the editor's full feature set.
💾	Save	Place the cursor over this icon to open a menu of save options.
⇨	Export	Place the cursor over this icon to open a menu of export options.
↶	Undo	Click this icon to undo the most recent action.
↷	Redo	Click this icon to redo the most recently undone action.
↺	Undo All	Click this icon to revert the view to its state when you last saved.
⟳	Switch Group	Click this icon to change the way groups are displayed. For more information, refer to **"Creating a View from a Domain" on page 152**.
A Z ↕	Sort	When working with tables, click this icon to view the current sorting and to select fields for sorting data. For more information, refer to **"Sorting Tables" on page 101**
☰	Input Controls	Click this icon to see the input controls applied to this view. For more information, refer to **"Using Input Controls" on page 149**.
☑	Page Options	Place the cursor over this icon to open a menu of page-level options. You can: • Change whether to display the Layout Band. • Change whether to display the title area. • In tables, you can also hide or show the detail rows when the data is summarized. This option is available only if the table includes grouped columns. • In crosstabs, you can merge or unmerge cells with the same data.

Icon	Name	Description
	View SQL/MDX Query	For more information on viewing SQL queries, see **"Viewing the SQL Query" on page 97** For more information viewing MDX queries, see **"Viewing the MDX Query" on page 123**

4.1.4.1 The Layout Band

Directly beneath the tool bar is the Layout Band. Here there are two fields. These fields have different labels and functions, depending on the type of view you are creating:

- For tables, these fields are **Columns** and **Groups**.
- For charts, these fields are **Columns** and **Rows**.
- For crosstabs, these fields are **Columns** and **Rows**.

You can drag and drop fields and measures into these boxes to populate your view.

4.1.4.2 Managing Canvas Options

When working with a table or chart view, the Canvas Options selector appears below the Layout Band, to the left of the view title. This tool allows you to control the level of detail displayed in your table or chart. Click to display the options available for your Ad Hoc view type.

For tables, the Canvas Options selector includes the following options:

- **Detailed Data** displays table detail only.
- **Totals Data** displays table totals only.
- **Details and Totals** displays both details and totals.

For charts, the Canvas Options selector includes the following options:

- **Chart Types** displays the available chart types.
- **Chart Format** displays formatting options.

4.1.4.3 Viewing the SQL Query

You may want to look at the SQL query for your view, to verify what data users are hitting. If you have the proper permissions, you can do this in the Ad Hoc Editor with the **View Query** button.

The query is read-only, but can be copied onto a clipboard or other document for review.

To view the SQL query:

- In the tool bar, click .

The View Query window opens, displaying the SQL query.

4.1.5 The Filters Panel

The Filters panel displays any filters defined for the view. You can set the filter values and see the resulting change in the Ad Hoc View panel. To hide the Filters panel, click the icon in the top left corner of the panel. Click the same icon on the minimized panel to expand it again.

For more information on working with filters, see **"Using Filters and Input Controls" on page 145**.

4.1.6 Saving an Ad Hoc View, Previewing and Creating a Report

After you create a compelling table, chart, or crosstab, you can save the view in the repository for future use.

To save an Ad Hoc view:

1. Hover your mouse over 💾 .
2. Select **Save Ad Hoc View** or **Save Ad Hoc View As**.
3. Name the view as needed, and click **Save**. The view is saved in the repository.

You can also save an Ad Hoc view as a report. Typically, a report is created when you want to:

- See data in the interactive report viewer.
- Perform additional formatting of the table data.
- Embed the data content in a dashboard.

Create a report from the view by selecting **Save Ad Hoc View and Create Report**; you can also create and run a report directly from the repository. For more information, see **"Running or Creating a Simple Report" on page 56**. When you run the report, it is displayed as a JasperReport.

4.1.6.1 Dependent Reports

When you create a report from an Ad Hoc view, the report is considered "dependent" on that view.

When you save an Ad Hoc view its dependent reports are not updated. For example, if you open an Ad Hoc view in the Editor and add a column to it, that column will not show up in previous reports created from that view. In such a case, you may want to save the updated view with a different file name to avoid confusion.

4.2 Working with Tables

The following sections explain how to populate, edit, and format your table-type view.

4.2.1 Using Fields in Tables

Insert data into your table by adding fields. All available fields are listed in the **Data Source Selection** panel, on the left side of the Ad Hoc Editor.

The available fields are divided into two sections in the panel:

- **Fields**, which can be added to the table as columns or groups.
- **Measures**, which are specialized fields that contain data values.

To add fields and measures as columns to a table:

1. In the **Data Source Selection panel**, click to select the field or measure you want to add to the table. Use Ctrl-click to select multiple items.
2. Drag the selected item into the **Columns** box in the Layout Band.

   The field is added to the view as a column in the table.

To remove a field or measure from a table:

- In the Layout Band, click the **x** next to the field or measure's name.

4.2.1.1 Groups

Groups allow you to create detailed data rows. For example, if you have a table that lists the suppliers for a national restaurant chain, you can group the suppliers by the State field. The suppliers' names are then rearranged so that all suppliers located in Maine, for instance, are located under a "Maine" header row; suppliers in Maryland are together under a "Maryland" header row, and so on.

You can use multiple fields to make more specific nested groups. By adding a group based on the "City" field to the table described above, the restaurant suppliers are arranged by City within the State groups. Under the "Maine" header row, new header rows for Augusta, Bangor, and Portland are added, and the names of the Maine-based suppliers appear under their respective cities. Under the "Maryland" header row, header rows for Annapolis, Baltimore, and Silver Spring are added, and the names of Maryland-based suppliers appear under those headers, and so on.

Only fields can be applied to a table as a group; measures cannot.

Data is grouped in the table according to the order they have defined. You can change the order by dragging the groups into position if needed.

To create a group:

1. In the **Data Source Selection** panel, click to select the field you want to add to the table as a group.
2. Drag the field to the **Groups** box in the Layout Band.

   The Ad Hoc view refreshes and displays the data grouped under a new header row.

You can also add a group to the table by right-clicking a field and selecting **Add as Group**.

To remove a group:

- In the Layout Band, click the **x** next to the field's name in the **Groups** box.

To move a the grouping order up or down in a table:

- In the Layout Band, drag the name of the group you want to move into its new position.

4.2.1.2 Summaries

You can display summary data for any column in your table. Summary data may be in the form of various functions, such as:

- Sum
- Count
- Distinct Count
- Average

For example, in a table with a list of stores, grouped by City and Country, you can display the number of stores in each City, and in each Country, using this function.

By default, the summary function for each field is defined by the data source, OLAP, or domain definition.

To add a summary to a specific column:

- In the table, right-click the column you want to calculate a summary for, and select **Add Summary**.

  The summary information is added to the group header, or is added to the bottom of a column if no groups are included in the table.

To remove a summary from a specific column:

- In the table, right-click the column with the summary you want to remove, and select **Remove Summary**. The summary information is removed from the table.

To add or remove summaries from all columns:

- Click ⚙ and select **Detailed Data**.

4.2.1.3 Column and Header Labels

You can edit a column or header label directly in the Ad Hoc Editor.

To edit a column or header label:

1. On the Ad Hoc view panel, right-click the column or group header you want to rename.
2. Select **Edit Label** from the context menu. The Edit Label window opens.
3. In the text entry box, delete the existing name and enter the new name.
4. Click **Submit**.

If space is at a premium, you can remove labels from the view. When you delete a label, it still appears when you look at the view in the Ad Hoc Editor, but does not appear when you run the report.

To delete a column or header label:

1. On the Ad Hoc view, right-click the column or header label you want to remove.
2. Select **Delete Label** from the context menu.

To re-apply a label:

1. Right-click the column or header label you want to replace.
2. Select **Add Label** from the context menu. The Edit Label window opens.
3. Enter the label name, if needed.
4. Click **Submit**.

4.2.1.4 Managing Column Size and Spacing

You can change the size of, and spaces between, columns to manage the appearance of your table or use space more efficiently.

To resize a column:

1. In the Ad Hoc View panel, click to select the column you want to resize.
2. Move the cursor to the right edge of the column.
3. When the cursor changes to the resize icon (↔), click and drag the column edge right or left until the column is needed size.

Spacers can be added to a table to arrange columns farther apart, or add margins to a table.

To change the spacing between columns:

1. In the **Data Source Selection** panel, in the Measures section, click **Spacer**.
2. Drag the spacer into the **Columns** box in the Layout Band between names of the two columns you want to move apart.

   A spacer column, labeled ◆, appears in the table.

3. Repeat this action to add space as needed between columns.
4. To remove a spacer, right-click the spacer column and select **Remove from Table**.

To use spacers to create table margins:

1. In the **Data Source Selection** panel, click to select **Spacer**.
2. Drag the spacer into the **Columns** box in the Layout Band.
3. Repeat until the margins are as wide as needed.
4. Repeat the steps above, adding the spacer to the right edge of the table.

4.2.1.5 Reordering Columns

You can move columns to the right or left to reorder data in your table.

To reorder a column:

1. In the Ad Hoc View panel, right-click the column you want to move.
2. Select **Move Right** or **Move Left** from the context menu.

4.2.1.6 Sorting Tables

In the Ad Hoc Editor, you can sort the rows of a table by any field, using a number of different methods.

To sort a table:

1. Click ⇅ . The Sort window appears. If the table is already sorted, the window shows the fields used.
2. To add a field to sort on, double-click the field in **Available Fields**. The Available Fields panel now lists only fields not currently in Sort On.
3. Select one or more fields to sort by. You can also use Ctrl-click to select multiple fields.
4. Click ▶ .
5. To arrange the sorting precedence of the fields, select each field in the Sort window and click **Move to top**, **Move up**, **Move down**, or **Move to bottom**: ⤒ , ▲ , ▼ , and ⤓ .
6. To remove a field, select it and click ◀ .
7. Click **OK**. The table updates to display the rows sorted by the selected fields.

You can also sort a table using the following methods:

- Right-click a field in the Fields section of the **Data Source Selection** panel, and select **Use for Sorting** from the context menu. In this case, the table is sorted by a field that isn't in the table; you may want to note the sorting fields in the title.
- Right-click a column header on the Canvas of the **Ad Hoc View** panel, and select **Use for Sorting** from the context menu.

If a column is already being used and you want to stop using it or change the sorting, right-click the column and select **Change Sorting** from the context menu.

4.2.1.7 Adding a Title

1. Above the table, click the text **Click to add a title**.
2. Enter the new table title in the text entry box.

4.2.1.8 Changing the Data Format

You can change the formatting for columns containing numeric data, such as dates and monetary amounts.

To change the data format for a column:

1. In the Ad Hoc view, right-click the column.
2. Select **Change Data Format** from the context menu.
3. Select the format you want to use. These options vary, depending on the type of numeric data contained in the column.

   The data in the column now appears in the new format.

4.2.1.9 Changing the Data Source

You may need to select a new data source for your table. This is a simple task, but you should keep in mind that **all view data and formatting are lost when you select a new Topic, Domain, or OLAP connection**. Any changes to the view are also lost if you navigate to another page using the browser navigation buttons, the main menu, or the **Search** field. To preserve changes, accept the current Topic or click **Cancel**.

To change the table's data source:

1. At the top of the **Data Source Selection** panel, click ▤▾ and select **Change Source.**
2. Select a different Topic, Domain, or OLAP connection.
3. Click **Table** to apply the new data source.

Click **Cancel** to return to the editor without changing the Topic.

4.2.1.10 Showing and Hiding Detail Rows

You can simplify or expand the information in your table by hiding or showing detail rows.

To hide detail rows in a table:

1. Place the cursor over ☑ .
2. Select **Hide Detail Rows** to show only the summarized totals for each group.

   The Ad Hoc Editor applies a summary to each field depending on its datatype.

To show detail rows in a table:

1. Place the cursor over ☑ .
2. Select **Show Detail Rows** to display the detailed information for each group.

   The Ad Hoc Editor displays the complete information in each row.

4.2.1.11 Controlling the Data Set

You can control the data displayed in the grid using the Grid Detail Selector: ⚙▾

Your options are:

- **Detailed Data**, which displays table detail only. For instance, in a table listing sales in dollars for all stores in a region for a given month, the amount sold by each store that month is displayed.
- **Totals Data**, which displays the table totals only. In the table described above, the total amount of all sales at all regional stores that month is displayed.
- **Details and Totals**, which displays both the individual store sales numbers, as well as the total sales numbers at the bottom of the store sales column.
  Click to select the option you want to apply to your table.

4.3 Working with Charts

Ad Hoc charts are a flexible, interactive way to explore your data graphically. You can choose different levels of aggregation for rows and columns, change a field from a column to a row, pivot the entire chart, hide chart values, and zoom in to see chart details.

The following sections explain how to populate, edit, and format an Ad Hoc chart. Many tasks related to working with charts are identical (or very similar) to those for tables and crosstabs. For any tasks not discussed in this section, see the information in **“Working with Tables” on page 98**.

4.3.1 Using Fields and Measures in Charts

You must add at least one measure to view a chart. Before any measures are added to the chart, the Ad Hoc Editor displays a placeholder with the legend displaying a single entry: Add a measure to continue. As you add measures, the editor displays the grand total of each measure in the chart.

The initial display reflects only the measures you add; it does not change when you add fields or dimensions. For example, for each measure you add to a bar chart, you see a bar with the total value of the measure, regardless of how many fields you add. This means you can add, remove, and arrange measures and fields without waiting for the display to update. Once you have the fields and measures you want, you can use the sliders on the right to select the level of detail you want. See **Figure 4-2** for more information.

All available fields are listed in the Data Selection panel, as either standard fields or measures.

- Standard fields can be added to a column or a row.
- Measures contain summarized values. They are typically numeric fields that determine the length of bars, size of pie slices, location of points (in line charts), and height of areas. They can be added to rows or columns, but must all be in the same target — that is, you can add one or more measures to the chart as columns, or add one or more measures to the chart as rows, but you cannot have one measure as a column and another as a row in the same chart.

When creating a chart, keep in mind that row and column groups are arranged in hierarchies, with the highest member of the hierarchy on the left. For an Ad Hoc view based on an OLAP data source, you can change the order of distinct dimensions by dragging, but you cannot change the order of levels within a dimension. For an Ad Hoc view based on a non-OLAP data source, you can drag the field headings to rearrange the hierarchy; the highest level in a group should appear to the left; the lowest level in a group should appear to the right. For example, it doesn’t make sense to group first by postal code then by country, because each postal code belongs to only one country.

To add a field or measure to a row or column:

1. In the **Data Selection** panel, select the field you want to add to the chart as a group. Use Ctrl-click to select multiple items.
2. Drag the selected item into the **Columns** or **Rows** box in the Layout Band.

4.3.1.1 Setting Levels

When you add a field or dimension to a column or row, a multi-level slider located at the top of the Filters pane allows you to set the level of aggregation to use for viewing the data. The number of fields or dimensions in the row or column determines the number of levels on the slider. Measures are not reflected in the slider.

You cannot adjust levels on time series charts.

The following figure shows the effect of the slider on a chart with one level of aggregation for both rows and columns.

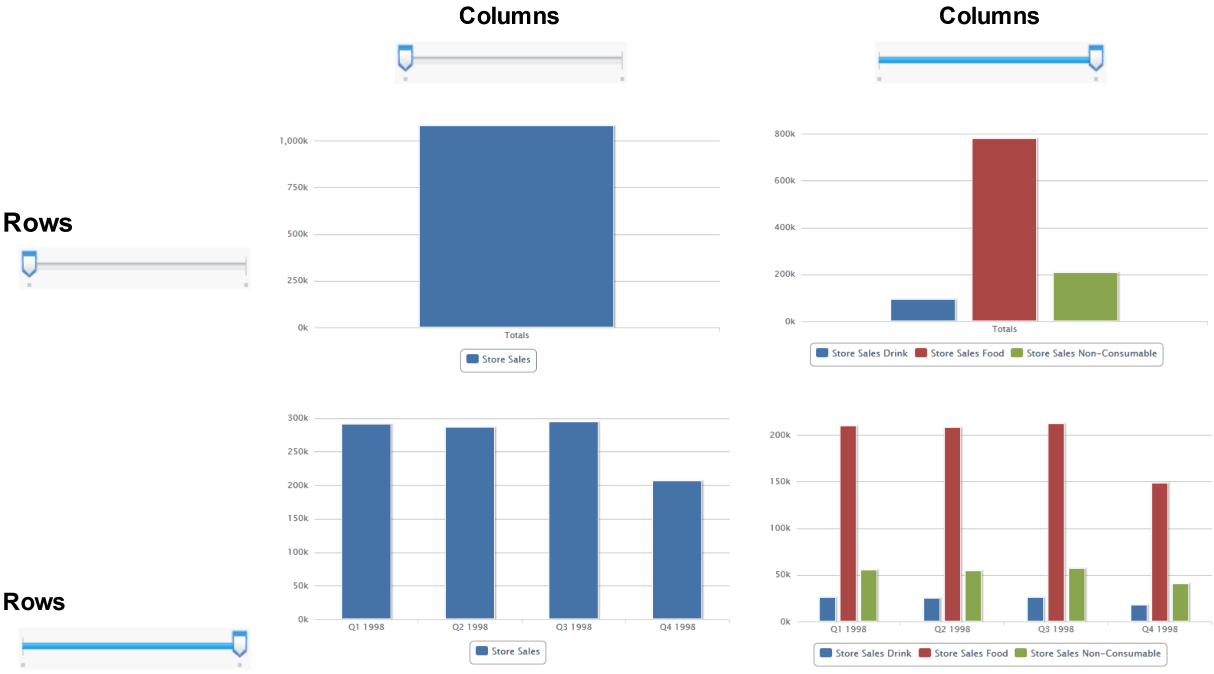


Figure 4-2 Effect of the Slider on a Chart

To recreate this view:

1. Select **Create > Ad Hoc View**.
2. In the Select Data wizard, select **foodmart data for crosstab** and click **OK**.
3. Drag the following from the Fields panel to the Layout Band:
   - Store Sales from **Measures** to **Columns**. The view changes to show a column with the total. No slider is added for measures.
   - Product Family from **Fields** to **Columns**. The **Data Level** area is shown in the **Filters** panel, with a **Columns** slider added.
   - Date from **Fields** to **Rows**. A **Rows** slider is added to the **Data Level** area in the **Filters** panel.
4. Use the sliders to see how the view changes.

The sliders help you explore your data visually in a number of ways:

- The slider reflects the hierarchy of the row or column groups, as determined by the order in which fields are arranged in the Layout Band.

- Hovering over a setting on the slider shows the name of the field or dimension corresponding to that setting.
- When you pivot a chart, slider settings are preserved and applied to the new target. For example, if you have the **Row** slider set to Month, the **Column** slider is set to Month when you pivot. See **“Pivoting a Chart” on page 105** for more information.
- When you remove the currently selected level from a row or column, the slider is reset to the total; when you remove a field that is not selected, the level remains the same. When you add a field or dimension to a row or column, the number of levels of the slider changes to reflect your addition. When you change the order of the fields in a row or column, the level on the slider changes to reflect the new level of the field corresponding to the selection.

4.3.1.2 Changing Date Grouping

If your chart includes data based on a date field, you can change the level of aggregation for the time data. To select the unit of time to chart:

- Right-click on the date field in the Layout Band and select **Change Grouping**. Then select the time period you want from the cascading submenu. The view updates to reflect the new date grouping.

Time Series charts can use only day, or smaller, intervals.

4.3.1.3 Changing the Summary Function of a Measure

You can get a new view of your data by changing the summary function of a measure, for example, from sum to average. To select a new summary function for a measure:

- Right-click on the measure in the Layout Band and select **Change Summary Function**. Then select the function you want from the cascading submenu. The view updates to reflect the new summary function.

4.3.1.4 Pivoting a Chart

You can pivot a chart in two ways:

- Pivot the entire chart by clicking . The row and column groups switch places; slider levels are maintained. The following figure shows the effect of pivoting a basic column chart.

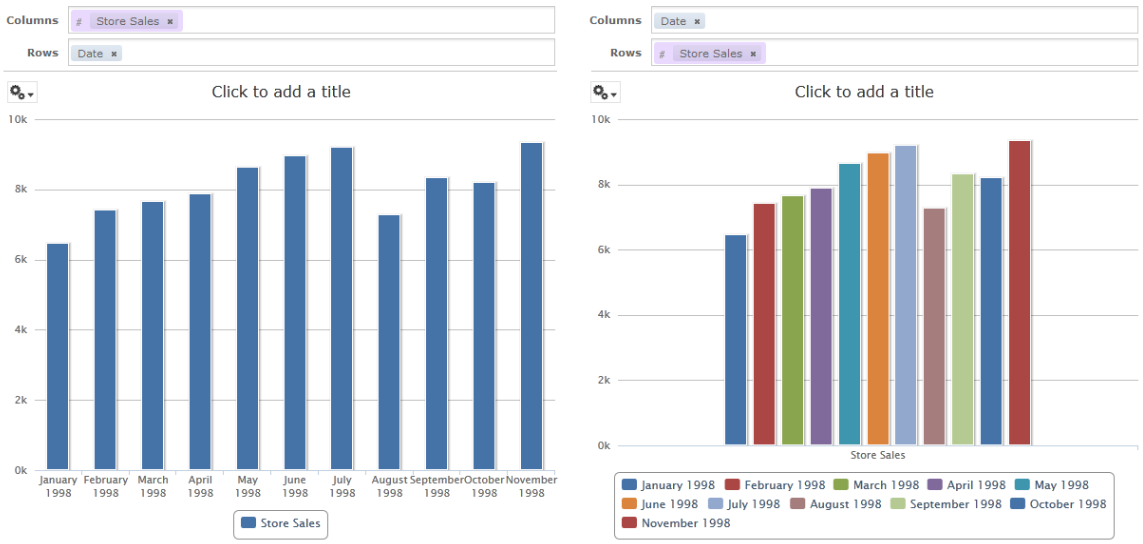


Figure 4-3 Effect of Pivoting a Chart

- Pivot a single group:
  - To pivot a single row group, right-click it and select **Switch To Column Group**. You can also move any field or dimension by dragging. You cannot drag a measure to a different group.
  - To pivot a single column group, right-click it and select **Switch To Row Group**. You can also move any field or dimension by dragging. You cannot drag a measure to a different group.

4.3.2 Selecting a Chart Type

Choose from multiple chart types to best represent your information, including:

- Column, which compares values displayed as columns.
- Bar, which compares values displayed as bars.
- Line, which compares values displayed as points connected by lines.
- Area, which compares values displayed as shaded areas.
- Spider, which compares three or more values on a series of spokes. Spider charts can use columns, lines, or areas to display values.
- Dual- and Multi-Axis, which display values using two or more measures.
- Time Series, which compares time intervals displayed as points connected by lines. The Time Series chart type is only available for non-OLAP charts.
- Scatter, which compares values as individual points arrayed across both axes of a chart.
- Bubble, which compares three measures displayed as circles of varying sizes arrayed across both axes of a chart.
- Pie, which compares values displayed as slices of a circular graph.
- Range, which displays values as heat and tree maps. Only the Heat Map chart is available for OLAP data sources.

By default, the Ad Hoc Editor creates a column chart. You can select a different type of chart at any time.

If you recently upgraded to JasperReports Server 5.5 or later you may find that existing scatter charts did not update, and are incompatible with the newer software. You can use the Ad Hoc views for those scatter charts to generate new reports, but you can't edit the views for scatter charts created in earlier versions of JasperReports Server.

To select a new chart type:

1. In the Ad Hoc View panel, click the ⚙ icon to show the Canvas Options menu.
2. Select **Chart Types...** from the menu. The **Select Chart Type** window is displayed.

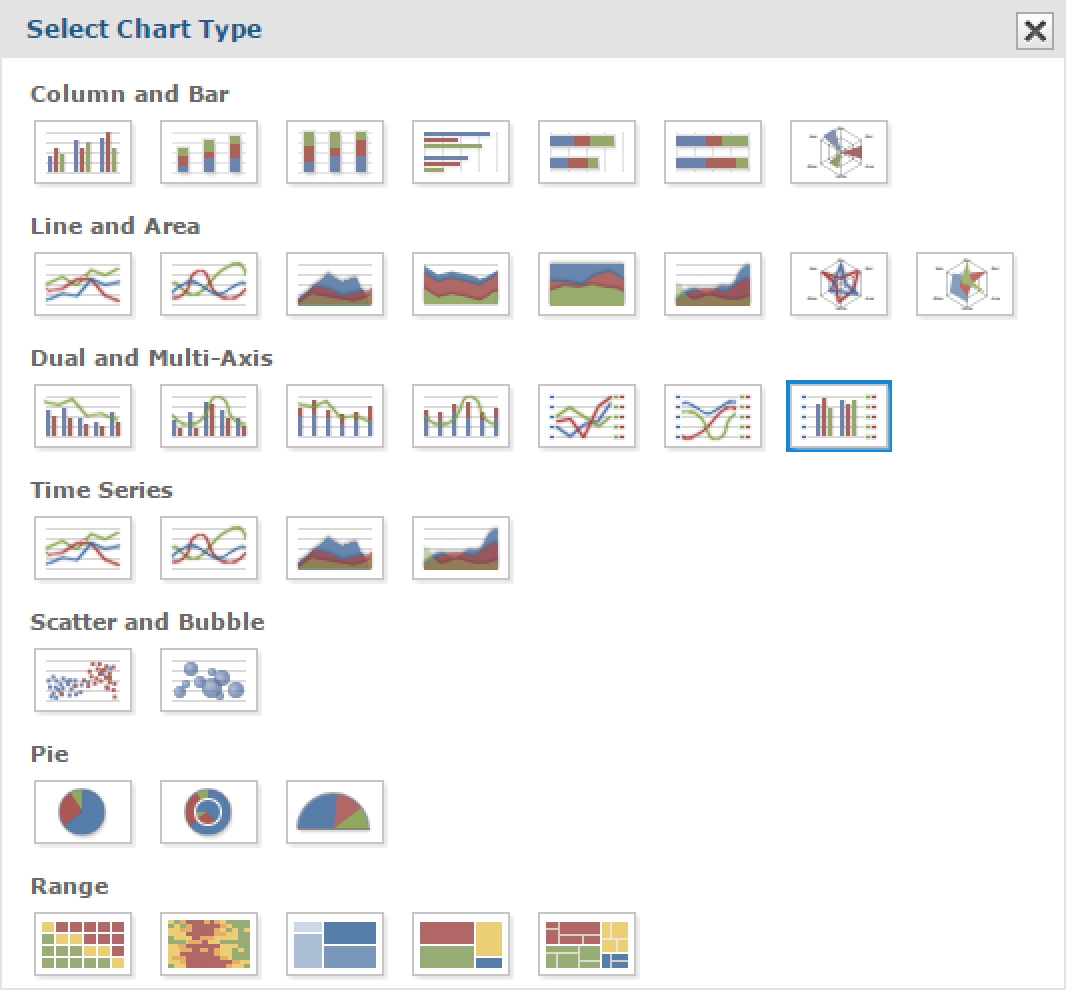


Figure 4-4 Select Chart Type Window

3. Click the type of chart you want to apply to your report. The selected chart type is outlined in blue.
4. Leave the **Select Chart Type** window open to rapidly switch between chart types, or click the ☒ icon at the top right to close it.

The following table describes the available chart types, and rules (if any) affecting their use:

Table 4-2 HTML5 Chart Types

Icon	Description	Rules
Column charts	**Column charts compare values displayed as columns.**	
	Column. Multiple measures of a group are depicted as individual columns.	
	Stacked Column. Multiple measures of a group are depicted as portions of a single column whose size reflects the aggregate value of the group.	
	Percent Column. Multiple measures of a group are depicted as portions of a single column of fixed size.	
	Spider Column. Multiple measures of a group are depicted as portions of individual "columns" along a spoked chart.	
Bar charts	**Bar charts compare values displayed as bars.**	
	Bar. Multiple measures of a group are depicted as individual bars.	
	Stacked Bar. Multiple measures of a group are depicted as portions of a single bar whose size reflects the aggregate value of the group.	
	Percent Bar. Multiple measures of a group are depicted as portions of a single bar of fixed size.	
Line charts	**Line charts compare values displayed as points connected by lines.**	
	Line. Displays data points connected with straight lines.	
	Spline. Displays data points connected with a fitted curve.	
	Spider Line. Displays data points connected with straight lines on a spoked chart.	

Icon	Description	Rules
Area charts		
	Area. Displays data points connected with a straight line and a color below the line; groups are displayed as transparent overlays.	
	Stacked Area. Displays data points connected with a straight line and a solid color below the line; groups are displayed as solid areas arranged vertically, one on top of another.	
	Percent Area. Displays data points connected with a straight line and a solid color below the line; groups are displayed as portions of an area of fixed sized, and arranged vertically one on top of the another.	
	Area Spline. Displays data points connected with a fitted curve and a color below the line; groups are displayed as transparent overlays.	
	Spider Area. Displays data points connected with straight lines and a solid color between the line and the center of a spoked chart; groups are displayed as transparent overlays.	
Dual- and Multi-Axis charts	**You can compare two or more measures, using one charting time or multiple charting types, with the charts listed here.**	
	Column Line. Displays leftmost measures as bars, last measure as a line.	• Requires two or more measures. • Measures must be placed in the Columns location. • Fields and dimensions can be placed only in the Rows location.
	Stacked Column Line. Displays leftmost measures as stacked bars, last measure as a line.	• Requires three or more measures. • Measures must be placed in the Columns location. • Fields and dimensions can be placed only in the Rows location.

Icon	Description	Rules
	Column Spline. Displays leftmost measures as bars, last measure as a spline.	• Requires two or more measures. • Measures must be placed in the Columns location. • Fields and dimensions can be placed only in the Rows location.
	Stacked Column Spline. Displays leftmost measures as stacked bars, last measure as a line.	• Requires three or more measures. • Measures must be placed in the Columns location. • Fields and dimensions can be placed only in the Rows location.
	Multi-Axis Line. Displays each measure as a separate axis line.	• Requires two or more measures. • Measures must be placed in the Columns location. • Fields and dimensions can be placed only in the Rows location.
	Multi-Axis Spline. Displays each measure as a separate axis spline.	• Requires two or more measures. • Measures must be placed in the Columns location. • Fields and dimensions can be placed only in the Rows location.
	Multi-Axis Column. Displays each measure as a separate axis column.	• Requires two or more measures. • Measures must be placed in the Columns location. • Fields and dimensions can be placed only in the Rows location.

Icon	Description	Rules
Time Series charts	**Time Series charts illustrate data points at successive time intervals.**	
	Line. Displays date and time data points connected with straight lines.	• Requires a single date/time field in the Rows location. • Field must be set to the "day" group function.
	Spline. Displays date and time data points connected with a fitted curve.	• Requires a single date/time field in the Rows location. • Field must be set to the "day" group function.
	Area. Displays date and time data points connected with a straight line and a color below the line; groups are displayed as transparent overlays.	• Requires a single date/time field in the Rows location. • Field must be set to the "day" group function.
	Area Spline. Displays date and time data points connected with a fitted curve and a color below the line; groups are displayed as transparent overlays.	• Requires a single date/time field in the Rows location. • Field must be set to the "day" group function.
Scatter charts	**Scatter charts show the extent of correlation between the values of observed quantities.**	
	Scatter. Displays first measure as the x-axis, the second measure as the y-axis. Other fields and dimensions in the column group become data points.	• Requires exactly two measures. • Measures must be placed in the Columns location.
Bubble charts	**Bubble charts show the correlation between three measures, displayed as disks.**	
	Bubble. Displays first measure as the x-axis, the second measure as the y-axis, and the third measure determines the size of the disk.	
Pie charts	**Pie charts display values as slices of a circular graph.**	
	Pie. Multiple measures of a group are displayed as sectors of a circle.	
	Dual Pie. Multiple measures of a group are displayed as sectors of concentric circles.	

Icon	Description	Rules
	Semi-Pie. Multiple measures of a group are displayed as sectors of a half-circle.	
Range charts	**Range charts display values heat maps.**	
	Heat Map. Individual values represented as colors.	Non-OLAP: • One field, followed by one Measure, required in the Columns location. • One field required in the rows location. OLAP: • One Dimension level, followed by one Measure, required in the Columns location. • One Dimension level required in the Rows location.
	Time Series Heat Map. Individual values across dates/times represented as colors.	• One Measure required in the Columns location. • One Date/Time field required in the Rows location. • Only available for non-OLAP data sources.
	Dual Measure Tree Map. Displays data as color-coded rectangles; the size of each rectangle is proportional to the first measure and the color represents the second measure.	• Two Measures required in the Columns location. • One field required in the Rows location. • Only available for non-OLAP data sources.
	Tree Map. Displays data as rectangles; the size of each rectangle is proportional to the measure of the data it represents. The tree map displays nested rectangles when you have more than one field; the parent rectangle represents the leftmost measure while the nested rectangles represent the current level of aggregation. Click on a parent rectangle to drill down to the nest rectangles.	• One Measure required in the Columns location. • One or more fields required in the Rows location. • Only available for non-OLAP data sources.

Icon	Description	Rules
▦	**Parent Tree Map**. Displays data as nested rectangles; the size of each rectangle is proportional to the measure of the data it represents. The nested rectangles represent the current level of aggregation while the larger rectangle represents the parent level in the hierarchy. Click on a parent rectangle to drill down to the nest rectangles.	• One Measure required in the Columns location. • Two or more Fields required in the Rows location. • Only available for non-OLAP data sources.

4.3.3 Formatting Charts

You can control some aspects of how data points, field names and labels are displayed on your chart, including:

- Whether data points are displayed.
- Showing a measure name on charts including only a single measure.
- Restricting the number of labels displayed.
- Rotating the direction of label text.

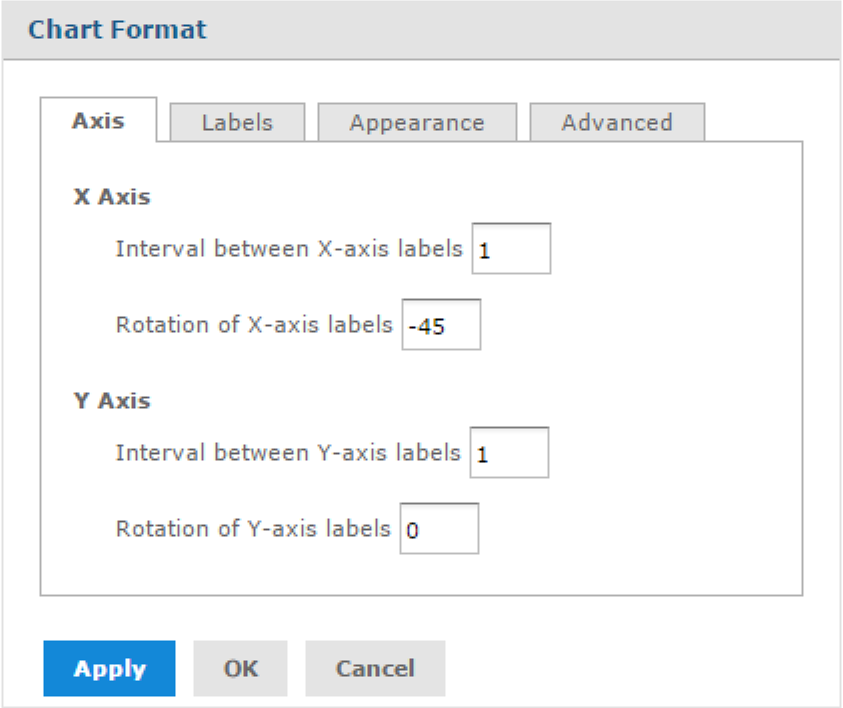
JasperReports Server also allows you to edit many of the chart's properties for cases when you want more control over the appearance of a chart.

4.3.3.1 Displaying Data Points

You can choose whether to display data points in your line, time series, or area chart. Data points can help users visualize chart data more accurately.

To show data points on a chart:

1. In the Ad Hoc View panel, click the ⚙ icon to show the Canvas Options menu.
2. Select **Chart Format...** from the menu. The **Chart Format** window is displayed.
3. Click the **Appearance** tab and select **Show data points on line charts**.
4. Click **Apply**, then **OK**. The name appears along the value axis.
5. To remove data points from the chart, open the **Appearance** tab and deselect **Show data points on line charts**.



X Axis

Interval between X-axis labels 1

Rotation of X-axis labels -45

Y Axis

Interval between Y-axis labels 1

Rotation of Y-axis labels 0

Figure 4-5 Chart Format Window

4.3.3.2 Displaying the Measure Name on the Value Axis

By default, in charts that include only a single measure, that measure's name is not displayed. For instance, if your measure displays the number of employees for each store in the region, that measure's label appears along the Y- (or value) axis, but the name of the measure ("Number of Employees" or similar) does not. You can, however, choose to display that measure name to clarify the information on your chart.

To show a measure name on the value axis:

1. In the Ad Hoc View panel, click the ⚙ icon to show the Canvas Options menu.
2. Select **Chart Format...** from the menu. The **Chart Format** window is displayed.
3. Click the **Labels** tab.
4. Select **Show measure name on value axis**.
5. Click **Apply**, then **OK**. The name appears along the value axis.
6. To remove a measure name, open the **Labels** tab and deselect **Show measure name on value axis**.

4.3.3.3 Restricting Label Display

By default, every field included in your chart has a label displayed along either axis. Measures will show up as numeric values, often along the Y axis, and fields being measured will show up as text along the X axis. On some charts - especially those with many included fields - these labels may overlap, crowd together, or become difficult to read. You can alleviate this problem by "stepping," or reducing, the number of labels displayed on your chart.

To reduce the number of labels on your chart:

1. In the Ad Hoc View panel, click the ⚙ icon to show the Canvas Options menu.

2. Select **Chart Format...** from the menu. The **Chart Format** window is displayed.
3. Click the **Axis** tab.
4. In the **Interval between X-axis labels** numeric entry box, specify how often you want the axis label to appear. For instance:
   - To display every second label, enter 2.
   - To display every third label, enter 3, and so on.
5. Repeat this process in the **Interval between Y-axis labels** numeric entry box.
6. Click **Apply**, then **OK**. The labels appear as entered.
7. To display every label, open the **Axis** tab and enter **1** in the numeric entry boxes.

4.3.3.4 Rotating Label Text

By default, labels on your chart are displayed horizontally. Multiple labels, or very long ones, can be difficult to read. You can change the direction of these labels, on both the X and Y axes, to improve chart readability.

To rotate label text:

1. In the Ad Hoc View panel, click the ⚙▾ icon to show the Canvas Options menu.
2. Select **Chart Format...** from the menu. The **Chart Format** window is displayed.
3. Click the **Axis** tab.
4. In the **Rotation of X-axis labels** specify the degree of rotation to apply to labels. For instance:
   - To rotate labels clockwise 90 degrees, enter 90.
   - To rotate labels counter-clockwise 90 degrees, enter -90.
   - To rotate labels clockwise 45 degrees, enter 45, and so on.
5. Repeat this process in the **Rotation of Y-axis labels** entry box for measures along the Y axis.
6. Click **Apply**, then **Close**. The labels are rotated.
7. To return the labels to their original, horizontal position, open the **Axis** tab and enter **0** in the numeric entry boxes.

4.3.3.5 Changing the Chart's Appearance Using Advanced Formatting

The **Advanced** tab on the **Chart Format** window gives you more control over the appearance of an Ad Hoc chart by allowing you to edit certain chart properties. For example, you can specify a list of colors for the chart, change the position of the legend, choose whether to display data values in the chart, and much more. You can find a full list of supported commands by clicking the **More Information** link on the **Advanced** tab or browsing to the Advanced Chart Formatting page on the Jaspersoft Community Site.

To edit the chart's properties:

1. In the Ad Hoc View panel, click the ⚙▾ icon to show the Canvas Options menu.
2. Select **Chart Format...** from the menu. The **Chart Format** window is displayed.
3. Click the **Advanced** tab.
4. Click **Add New Property**.
5. In the Property field, enter the chart property you want to format and then enter the value(s) for the property. For instance:
   - To format the chart's colors, enter colors for the property and a comma-separated list of colors in brackets for the values, such as ["red", "blue", "green", "magenta", "purple", "black", "yellow"].
   - To change the vertical alignment of the legend, enter legend.verticalAlign for the property and "top" or "bottom" or "center" for the value.

- To display data values on the chart, enter plotOptions.series.dataLabels.enabled for the property and true for the value.

Click the **More Information** link to see a list of the properties you can edit.

The Ad Hoc Designer does not validate the property name or values. The chart will ignore any invalid property you enter.

6. Click ✓ to save the property formatting.
7. Click **Apply**, then **Close**. The chart is updated with the new formatting.
8. To remove the formatting, open the **Advanced** tab and click 🗑 next to the property you want to remove.

4.3.4 Interacting with Charts

Once you have chosen rows and columns and selected the type of chart you want, you can further explore your data using interactive features such as brushing an area of the chart to zoom in or clicking on a legend to hide the members of a group.

4.3.4.1 Zooming

Zooming lets you view a specific area of a chart more closely. Zooming can also be helpful when the labels at the bottom of a chart are difficult to read, because only the labels corresponding to the selected area are displayed.

Zoom is a viewing feature of the Ad Hoc Editor. When you save an Ad Hoc view, or create a report from an Ad Hoc view, the zoom is automatically reset to show the whole chart.

To zoom in on an area of a chart:

- Click and drag or brush the area you want to zoom in on. As you are dragging or brushing a pale blue area indicates your selection. When you release the mouse button, the view zooms in on the area you selected.

To view the whole chart again:

- Click **Reset zoom** at the upper right of the canvas.

The following images show a bar chart before and after zooming:

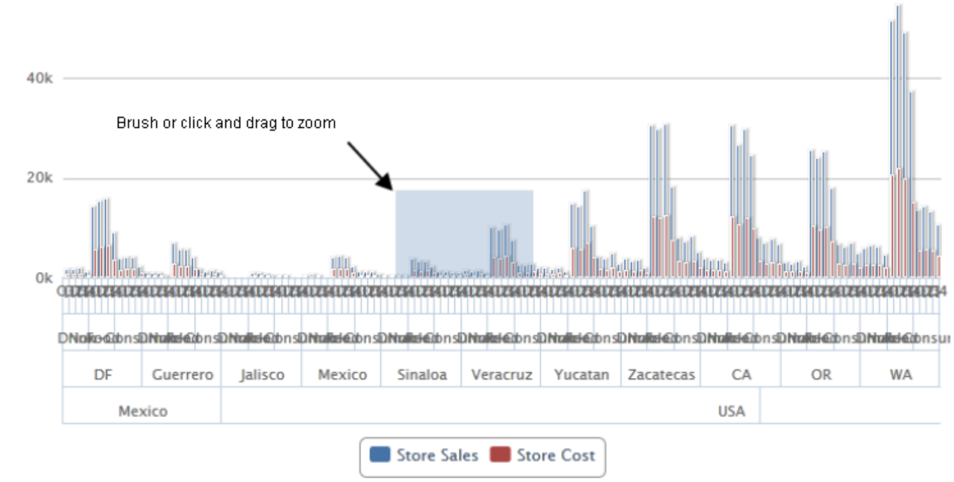


Figure 4-6 Selecting a Zoom Area

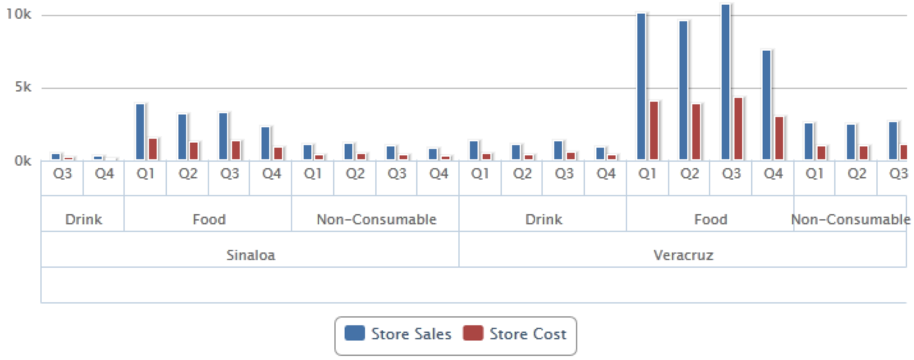


Figure 4-7 Area After Zooming

4.3.4.2 Hiding Group Members

Use the legends below the chart to hide or show group members.

- To hide a group member, click the member name in the legend below the chart. The member is removed from the chart and the legend is grayed out.
- To unhide a group member that has been hidden, click the grayed-out legend for the member.

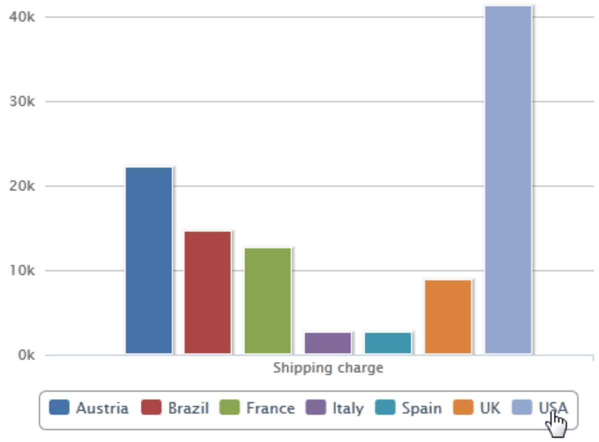


Figure 4-8 Hiding a Group Member

Hidden members are a view feature of the Ad Hoc Editor. When you save an Ad Hoc view, or create a report from an Ad Hoc view, the chart is automatically reset to show all members.

4.4 Working with Standard Crosstabs

This section describes crosstabs based on Topics and Domains. For information about crosstabs based on OLAP connections, refer to **"Working with OLAP Connection-based Crosstabs" on page 121** and **"Working with Topics" on page 157**.

Crosstabs have different data, layout, and format options than tables or charts.

If you selected Crosstab when you created a view, as described in **"Ad Hoc View Types" on page 92**, the following sections explain tasks specific to your crosstab development.

4.4.1 Using Fields in Crosstabs

Fields can be added to crosstabs as row groups or column groups. Measures can be added to crosstab rows or columns as well, but all measures must be included as either a row or a column – that is, you can add one or more measures as columns, or add one or more measures as rows, but you cannot have one measure as a column and another as a row in the same crosstab.

4.4.1.1 Crosstab Rows and Columns

When creating a crosstab view, keep in mind that row and column groups are arranged in hierarchies. Drag the group headings to rearrange the hierarchy; you can also right-click a heading and select a **Move** option from the context menu or press the cursor keys. Rearranging the groups may change the preview data in the editor.

To add a field or measure to a crosstab group:

1. In the **Data Source Selection** panel, click to select the field(s) you want to add to the crosstab as a group. Use Ctrl-click to select multiple items.
2. Drag your selection into the **Columns** or **Rows** box in the Layout Band.

4.4.1.2 Crosstab Measures

Measure labels are displayed in the crosstab based on their status as a row or column:

- Measures included as rows appear in the crosstab below the **Measures** heading.
- Measures included as columns appear in the crosstab to the right of the **Measures** heading.

You can right-click a measure in the crosstab to open a context menu that provides these options:

- **Change Summary Function**
- **Change Data Format**
- **Remove From Crosstab**
- **Create Filter**
- **Move Up** or **Move Down**

Measures are arranged in cells. You can add any number of measures. All the measures appear together in every cell. To rearrange the measures, drag them in the measure label area.

4.4.1.3 Pivoting

You can pivot a crosstab in two ways:

- Pivot the entire crosstab by clicking . The row and column groups switch places. For more information, see **"Creating a View from a Domain" on page 152**.
- Pivot a single group:
  - To pivot a single row group, right-click it and select **Switch To Column Group**.
  - To pivot a single column group, right-click it and select **Switch To Row Group**.

Pivoting removes any custom sorting applied to headings in your crosstab. It does not affect column or row sorts.

4.4.1.4 Slicing

The slice feature lets you keep or exclude group members in a crosstab. To slice, right-click a group member and select:

- **Keep Only** to remove all groups except the selected one from the crosstab.
- **Exclude** to remove this group from the crosstab.

Use Ctrl-click and Shift-click to select multiple groups to keep or exclude.

You can select multiple row groups or multiple column groups; you can't slice by both row groups and column groups at once. Compare slice to drill-through, drill to details, and filtering.

For more information about filtering, see **"Using Filters and Input Controls" on page 145**.

4.4.1.5 Summarizing

All row and column groups are summarized automatically:

- To turn off a group summary, right-click any heading in the group and select **Delete Row Summary** or **Delete Column Summary** from the context menu. To reapply the summary, right-click the heading and select **Add Row Summary** or **Add Column Summary.**

The Delete Summary option is available only for the outermost group on either axis (that is, either the outermost row group or the outermost column group)

- To select the summary function and data format for a measure, right-click the measure label and select from the context menu. Note that you can’t change the summary function on custom fields that calculate percents (Percent of Total, Percent of Column Group Parent, and Percent of Row Group Parent).
- The summary functions for numeric fields are Sum, Average, Maximum, Minimum, Distinct Count, and Count All. Distinct Count is the number of different items in the row or column; Count All is the total number of items. For instance, if there are 3 widgets of type A and 3 widgets of type B, Distinct Count is 2 and Count All is 6.

4.4.1.6 Collapsing and Expanding Members

By default, the editor displays each row and column group of a crosstab in a collapsed state. This means you can see the totals for the group, but not the measures for its individual members. To see measures for a group’s members, right-click the group label and select **Expand Members**.

When a group’s members are expanded, select **Collapse Members** from the same menu to hide the measures. Collapsing an outer group also collapses its inner groups. The **Expand Members** and **Collapse Members** options are available only for outermost groups, or for inner groups nested in an expanded outer group.

When you collapse a group, its summary is automatically displayed; this prevents invalid crosstab layouts in which there is nothing to display for some totals if the summary has been deleted previously.

4.4.1.7 Merging and Unmerging Crosstab Cells

By default, the editor merges cells containing the same data into a larger, single cell to make the crosstab data easier to read. The following image shows a crosstab with merged cells.

To display all of the individual cells in the crosstab instead of merged cells, hover over ☑ and select **Unmerge crosstab cells**. The following image shows a crosstab with unmerged cells.

To merge the crosstab cells, hover over ☑ and select **Merge crosstab cells**.

4.4.1.8 Sorting

By default, the rows and columns of crosstabs are sorted in alphabetical order of the group names.

To sort your crosstab:

- Right-click the heading you want to use for sorting and select one of these options:
  - **Sort Ascending**
  - **Sort Descending**
  - **Don't Sort**

  The crosstab is updated to reflect your sorting option. A blue dot appears in the context menu next to the currently applied sort option.

When the crosstab includes more than one row group or more than one column group, the inner groups are also sorted according to your selection. Only one measure can be used for sorting at any one time; changing the sort order for another measure resets all others to the default.

To filter top or bottom N values:

You can filter the numeric data shown in a crosstab to show only the rows with the top or bottom N values, where N is a number that you specify. For example, you can filter a crosstab to display only the top 10 values in a column.

1. Right-click the heading you want to use for filtering and select one of these options:

  - **Filter Top N Values**
  - **Filter Bottom N Values**
  - **Don't Filter Values**

2. Enter the number of values you want to show in the crosstab.
3. Select whether to show an aggregate of the unranked values in the crosstab.
4. Select whether to apply the filter across all row groups.
5. Click **OK**.

   The crosstab is updated to reflect your filter option. The ▼≡ icon appears in the heading when filtering a column for the top N values, while the ▼≡ icon appears when filtering for the bottom N values. A blue dot appears in the context menu next to the currently applied filter option. Only one measure can be used for filtering at any one time; changing the filtering or sort order for another measure resets the filtered column.

4.4.1.9 Resizing and Layout

Many of the layout and formatting options that are set manually in tables are set automatically in crosstabs. In particular, row and column sizes are fixed and no spacer is available.

4.4.1.10 Drilling Through Data

Drill-through functionality is available for crosstab data. A drill-through table displays the supporting details for the selected roll-up value.

To view the drill-through table for a value in your crosstab:

- Click hyperlinked data to display additional columns from that specific fact data.

By default, the drill-through table opens in its own window or tab, depending on your browser settings.

4.5 Working with OLAP Connection-based Crosstabs

An OLAP connection is a definition for retrieving OLAP data for use in Ad Hoc views and reports. An OLAP connection is either a direct Java connection (Mondrian connection) or an XML-based API connection (XML/A connection). You must create OLAP client connections in order to use this functionality, which may be restricted by your license. For more information on creating OLAP client connections, as well as other administrative tasks regarding OLAP, refer to the *Jaspersoft OLAP User Guide*.

The Ad Hoc Editor displays a more focused tool bar when your Ad Hoc view is based on an OLAP connection. It contains a subset of the options available in the standard tool bar, including Display Mode, Redo, Undo All, Switch Group, Display Options, and View MDX Query. For a description of these options, see **Table 4-1**.

4.5.1 Dimensions and Measures

When you create a view based on an OLAP connection, the cube metadata defined in the connection's OLAP schema is automatically applied to your data: the dimensions and measures in the cube are shown in the Data Source Selection panel.

Dimensions and measures describe your data in terms of these organizing principles and facts:

- **Dimension**: A categorization of the data in a cube. For example, a cube that stores data about sales figures might include dimensions such as time, product, region, and customer's industry. A dimension is a hierarchical series of relationships in which each level has a parent and may also have children. Note that a level is a particular location within the dimension, whereas a member is a specific data element at a particular level. For example, if the dimension is Geography, then one of its levels might be Country; at the Country level, USA might be a member.

Some dimensions include a special level at the top of the hierarchy that includes every level in the dimension. This All member is used to display the summarized data across the dimension. The All member of dimensions is a common feature in many OLAP solutions.

- **Measure**: A field that displays the facts that constitute the quantitative data in a cube. For example, a cube that stores data about sales figures might include measures such as unit sales, store sales revenue, and store sales cost.

The items in the **Dimensions** section of the Data Source Selection panel may appear in a multi-level tree structure. You can add the entire dimension tree, or any sub-trees, to your crosstab by dragging the top-level name into the Layout Band. Individual items can be added to the crosstab in the same manner.

Items in the Measures section are organized into a single-level list. Any calculated measures are indicated by a calculation icon .

When working with OLAP connections, the Layout Band at the top of the Ad Hoc View panel allows you to drag and drop dimensions and measures into the crosstab. As you add columns and rows, the crosstab automatically builds on the panel. In the **Columns** and **Rows** fields, each item in your crosstab is represented as a token; drag tokens left and right to change their position on an axis or drag them between the fields to pivot them.

The cube metadata applied to your view provides structure to your crosstab: you can add a dimension with any number of levels, but it must follow the hierarchy defined in the cube. For example, if a dimension's levels are Country, State, and City, you can place any one of them in the crosstab, but if you add two or more, their order (defined in the OLAP schema) is enforced. For example:

Possible	Impossible
Country, State, City	State, Country, City
Country, City	City, Country
State, City	City, State
Country, State	State, Country

In addition, you can't mix members from different dimensions. For example, consider a crosstab that includes both a Geography dimension (Country, State, and City) and a Time dimension (Year, Quarter, and Month): you can't insert the Year level between the Country and State levels. The levels of each dimension are always kept together.

All measures in the crosstab must be either rows or columns. You can't have measures on both axes at the same time. To move measures between dimensions, right-click and select **Switch To Column Group** or **Switch To Row Group**. All the measures move to the other axis.

4.5.2 Drilling Through Data

Drill-through functionality is available for crosstab data. A drill-through table displays the supporting details for the selected roll-up value.

To view the drill-through table for a value in your crosstab:

1. Click hyperlinked data to display additional columns related to that information.
2. Use the page controls to navigate the table.

The drill-through table opens in its own window or tab, depending on your browser settings.

4.5.3 Sorting

Like standard crosstabs, the rows and columns of an OLAP crosstab are sorted in alphabetical order of the group names.

To change the sorting of your OLAP crosstab:

- Right-click the heading you want to use for sorting and select one of these options:
  - **Sort Ascending**
  - **Sort Descending**
  - **Don't Sort**

  The crosstab is updated to reflect your sorting option. A blue dot appears in the context menu next to the currently applied sort option.

When the crosstab includes more than one row group or more than one column group, the inner groups are also sorted according to your selection. Only one measure can be used for sorting at any one time; changing the sort order for another measure resets all others to the default.

4.5.4 Viewing the MDX Query

You may want to view the MDX query created for the OLAP connection used with your crosstab, to verify what data users are hitting. If this option is enabled, you can do this in the Ad Hoc Editor with the **View Query** button.

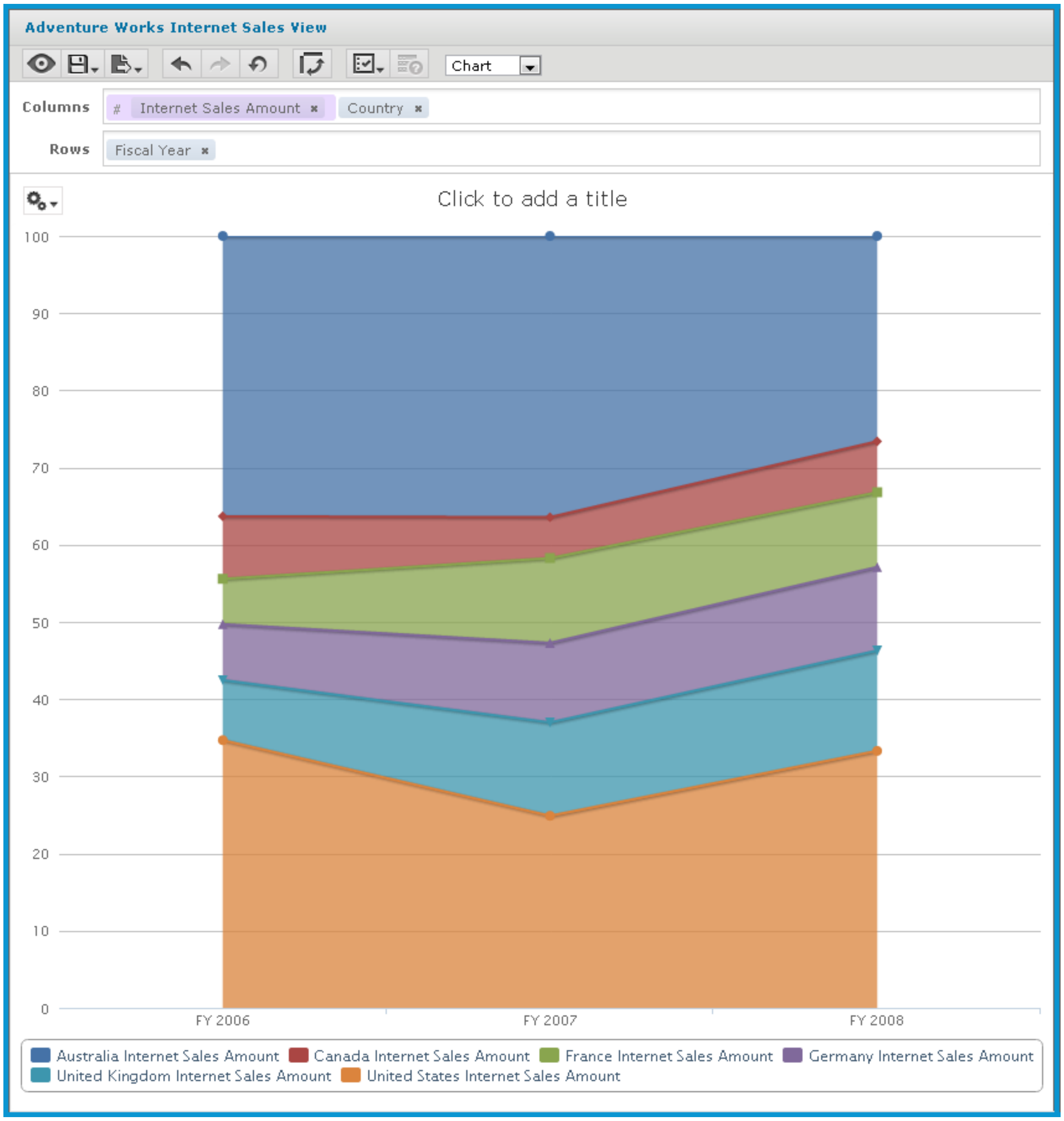
The query is read-only, but can be copied to a document for review.

To view the MDX query:

- In the tool bar, click . The View Query window opens, displaying the query.

4.5.5 Working with Microsoft SSAS

The Ad Hoc Editor allows you to view data from Microsoft SQL Server Analytical Services (SSAS) to populate views and reports with multi-dimensional OLAP data. XML/A connections that point to your SSAS data are listed with the other OLAP connections when you create an Ad Hoc view, as in the following chart example.


Adventure Works Internet Sales View
Chart
Click to add a title
100
90
80
70
60
50
40
30
20
10
0
FY 2006
FY 2007
FY 2008
Australia Internet Sales Amount
Canada Internet Sales Amount
France Internet Sales Amount
Germany Internet Sales Amount
United Kingdom Internet Sales Amount
United States Internet Sales Amount

Columns: # Internet Sales Amount, Country

Rows: Fiscal Year

Figure 4-9 Search Results Listing

For details, refer to the *Jaspersoft OLAP User Guide*.

4.6 Calculated Fields and Measures

You can create new fields or measures in an Ad Hoc view by applying formulas to a view's existing fields and measures. For example, consider a view that includes both a Cost and a Revenue field. You could calculate the profit for each record by creating a custom field that subtracts the Cost field from the Revenue field. For example, you can divide the Profit custom field in the previous example by the Revenue field to express each record's margin as a percent.

If you have installed the samples, the Ad Hoc view **10. Calculated Fields and Measures** includes examples of calculated fields (shown by the icon) or calculated measures (shown by the icon). You can explore their formulas by right-clicking the field or measure name and selecting **Edit**. You can also create tables, charts, or crosstabs to see how these calculations work in views.

4.6.1 Overview of the Calculated Fields Dialog Box

The **Calculated Field** or **Calculated Measures** dialog box allows you to create a calculated field or measure and set its summary function. This section describes the functionality available in this dialog box.

To open the calculated fields dialog box for Ad Hoc views:

1. Create or open an Ad Hoc view.
2. Open the calculated fields dialog using one of these methods:
   - Click at the top right of either the **Fields** section or the **Measures** section of the Data Source Selection panel and select **Create Calculated Field...** from the context menu.
   - Right-click on an existing calculated field (shown by the icon) or calculated measure (shown by the icon) and select **Edit**.

   The dialog displays a text field for the name and two tabs, **Formula Builder** and **Summary**.

The following are reserved words and cannot be used as field names: AND, And, and, IN, In, in, NOT, Not, not, OR, Or, or. Names containing these strings, such as "Not Available", can be used.

4.6.1.1 The Formula Builder Tab

The Formula Builder tab is where you create the formula for your calculated field or measure. This tab includes the following:

- Formula entry box — Shows the current formula for calculating your field or measure. You can edit the formula by typing directly in the panel. You can also add Fields, Measures, and Functions by double-clicking them. Click the buttons below the Formula field to add operators. Formulas must use the following syntax:
  a. Labels for fields and measures must be in double quotes ("): "Customer ID", "Date ordered".
  b. Text must be in single quotes ('): '--'.
  c. Levels must be in single quotes ('): 'ColumnGroup', 'Total'. See **Levels in Aggregate Functions** for more information about levels.

  For more information on syntax, see **Calculated Field Syntax**.
- Operator buttons — Click these buttons to insert the operator in the Formula entry box. For more information on operators, see **Operators in Ad Hoc Views**
- Fields and Measures — Lists all the fields and measures currently in your Ad Hoc view, including any calculated fields or measures you have already created.
- Functions — Lists all the available functions you can use in your formula. For more information, see **Calculated Field Reference**.
- Function Description panel — Gives a brief description of the function selected in the Functions list, if any. The sample inputs are intended to be as descriptive as possible. See **Calculated Field Reference** for the precise syntax each function requires.
- Show arguments in formula checkbox — When this checkbox is selected, double-clicking a function name in the Functions list adds the full description to the Formula entry box; when the checkbox is not selected,

double-clicking a function name adds only the function. For example, double-clicking `Round` adds `Round ("NumberFieldName", Integer)` when the checkbox is selected, and adds `Round()` when the checkbox is not selected. If you select this checkbox, you can double-click on a string, such as `NumberFieldName`, and then replace it by double-clicking a name in the Fields and Measures list.

- Validate button — Checks the formula for syntax errors, such as missing parentheses or quotes. Your calculated field or measure must be valid before you can create it. Syntax validation does not guarantee that your formula will give the results you want.

4.6.1.2 The Summary Tab

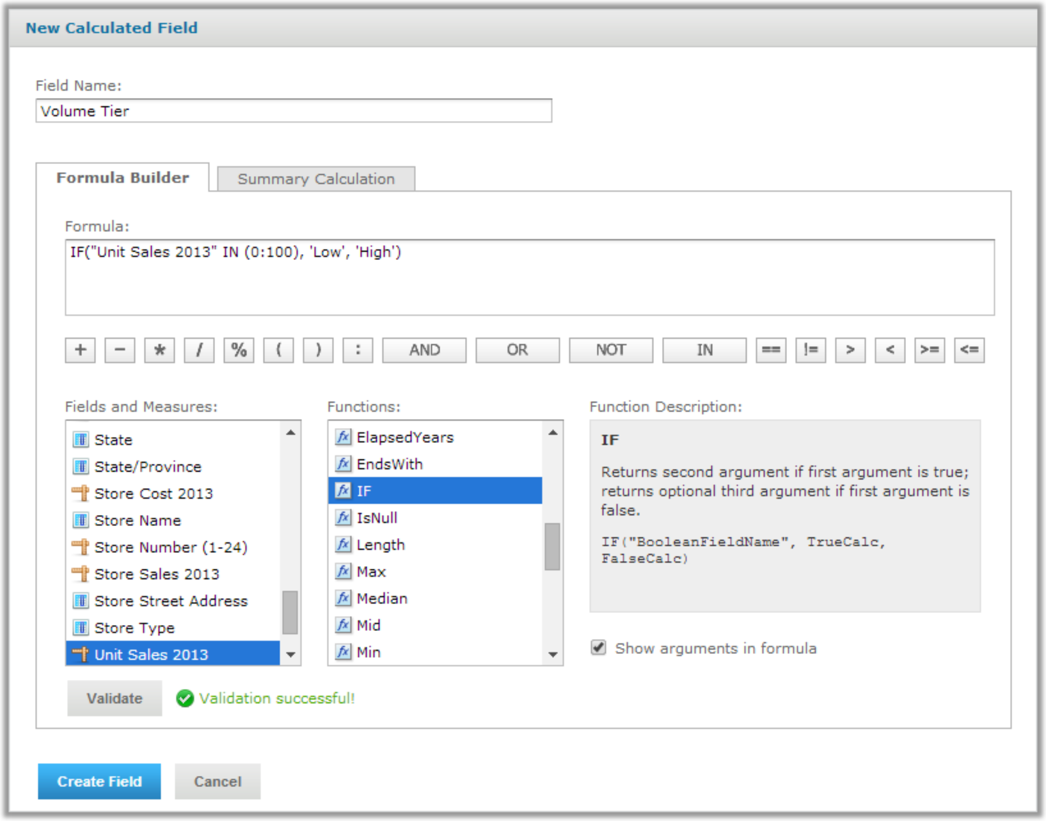
Summaries show a result applied to all data values. For example, for a numeric fields such as Cost, the summary value might be the sum of all the costs; for a text field such as Customer Name, the summary value might be the count of all customers. The Summary tab lets you set the default summary function for your calculated field or measure.

- Calculation list — Displays allowed summary functions for your calculated field or measure. The available options depend on the data type of the calculation. See **Summary Calculations** for more information. Depending on your selection, you may see additional options:
  - Custom selection — Displays the same options available in the Formula Builder tab, including the Formula entry box, operator buttons, Fields and Measures list, Functions list, and Validate button. You use these options to build a formula for your custom summary. However, for summaries, you are limited to aggregate functions, that is, functions that operate on all the values in your field. For example, `Sum` and `Mode` are valid summary functions, because they use all available field values to get a result. `Round` is not a valid summary function, because it operates on a single value at a time. See **Summary Calculations** for more information.
  - Weighted Average — Displays a Weighted On drop-down list, which allows you to choose another field or measure to use as the weight for the average.

4.6.2 Creating a Calculated Field

To create a calculated field:

1. First, create the Ad Hoc view to use. To do this, select **Create** > **Ad Hoc View** from the menu. The **Select Data** wizard appears.
2. Click and navigate to **Domains**.
3. Select **Supermart Domain**. Click **Choose Data**. The Data Chooser window appears.
4. In the Data Chooser window, double-click **Sales** to select it and click **OK**. A new Ad Hoc view opens.
5. In the Ad Hoc view, click at the top right of the **Fields** section and select **Create Calculated Field...** from the context menu. The New Calculated Field dialog box appears, displaying the Formula Builder.



Field Name: Volume Tier

Formula:

```
IF("Unit Sales 2013" IN (0:100), 'Low', 'High')
```

IF

Returns second argument if first argument is true; returns optional third argument if first argument is false.

`IF("BooleanFieldName", TrueCalc, FalseCalc)`

☑ Show arguments in formula

Validation successful!

Figure 4-10 Formula Builder Tab in New Calculated Measure Dialog Box

6. Enter Volume Tier for the **Field Name**.

Creating the formula:

This section shows how to create a simple text formula that says Low when the Unit Sales amount is under 100 and High otherwise.

Formulas must use the following syntax:

a. Labels for fields and measures must be in double quotes ("): "Customer ID", "Date ordered".

b. Text must be in single quotes ('): '--'.

c. Levels must be in single quotes ('): 'ColumnGroup', 'Total'.

7. Make sure **Show arguments in formula** is selected.

8. Now create the formula. Double-click IF() in the **Functions** list.

   Because **Show arguments in formula** is selected, `IF("BooleanFieldName", TrueCalc, FalseCalc)` is entered in the Formula Builder.

9. Double-click `BooleanFieldName` to select it, then double-click Unit Sales 2013 in the **Fields and Measures** list.

   The Formula Builder displays `IF("Unit Sales 2013", TrueCalc, FalseCalc)`.

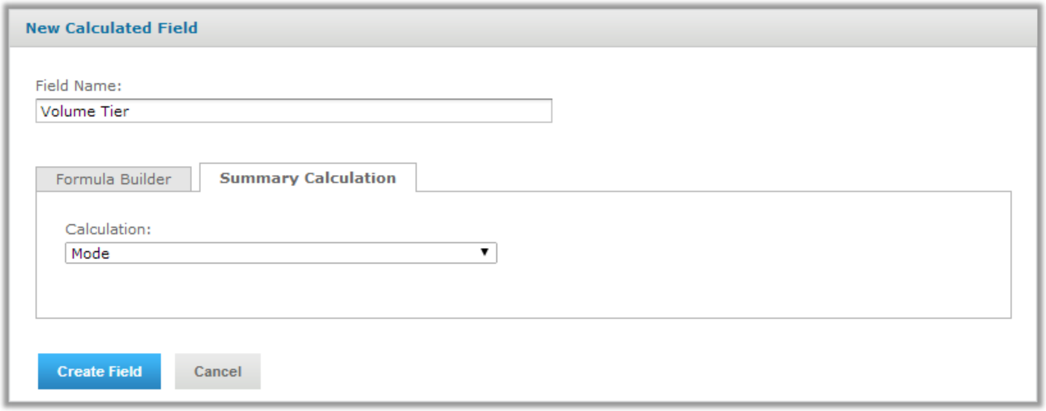
10. Edit the expression in the Formula Builder to read as follows: `IF("Unit Sales 2013" IN (0:100), 'Low', 'High')`.

11. Click **Validate** to verify that the formula does not have any syntax errors.

Creating a summary calculation:

The Ad Hoc Editor creates a default summary calculation based on the type of formula you have entered. This section shows how to select a different summary function.

12. Click the **Summary Calculation** tab.



New Calculated Field

Field Name: Volume Tier

Calculation: Mode

Figure 4-11 Summary Tab in New Calculated Measure Dialog Box

13. Select Mode from the **Calculation** menu.
14. Click **Create Field**.

    The calculated field appears in bold text at the bottom of the list of available fields. A special icon indicates it is a calculated field fx.

If you have installed the samples, the Ad Hoc view **10. Calculated Fields and Measures** includes examples of calculated fields and measures (indicated by fx). You can explore their formulas by right-clicking the field or measure name and selecting **Edit**. You can also create tables, charts, or crosstabs to see how these calculations work in views.

4.6.3 Planning and Testing Calculated Fields and Measures

When you are creating calculated fields and measures, you may need to follow an iterative process: first create your new fields and measures, then view the results, and finally adjust as needed. The following practices can help:

- Reduce the size of your data set — To speed field creation and testing, reduce the size of your working data set in the following ways:
  - Select **Sample Data** from the drop-down menu in the Ad Hoc tool bar.
  - Create one or more filters. This is especially useful for tables, which display all data by default.
  - Limit the number of fields and measures you add to your test reports to restrict the number of summary levels to one or two.
- Create one or more formulas, as described in **Creating a Calculated Field**.
- After you have created your fields and measures, add them to a table or crosstab in your view and verify that they behave as you expect. You may need to edit your fields to get the desired behavior. See **Creating a Calculated Field** for more information.

In addition, keep the following in mind when creating calculated fields:

- When you create a calculated field, it appears at the bottom of the list.
- Calculated fields that use aggregate functions cannot be added to groups and should not be used as filters.
- By default, the Ad Hoc Editor supports only two decimal places. If the calculated fields return data that are significant to the third decimal place, you can add new masking options by editing configuration files. For more information, refer to the *JasperReports Server Administrator Guide*.
- You can't delete a calculated field that is in use; these fields have their name in *italics*.
  - If the calculated field is used in the Ad Hoc view panel, remove the calculated field from the Ad Hoc View panel and then delete it from the **Data Source Selection** panel.
  - If the calculated field is the basis of another field, you can't delete it until you delete the one that builds on it.

4.6.4 Calculated Field Reference

This section lists all functions that can be used to create calculated fields and measures in Ad Hoc views. For details of the supported syntax, see **Calculated Field Syntax**.

These functions are available for calculated fields and measures in Ad Hoc views only. See **Operators and Functions** for supported operators and functions in Domains.

The examples in this section indicate correct syntax, but are not necessarily associated with an Ad Hoc view. If you have installed the Jaspersoft samples, the Ad Hoc view **10. Calculated Fields and Measures** includes examples of calculated fields (shown by the icon) or calculated measures (shown by the icon). You can explore their formulas by right-clicking on the field or measure name and selecting **Edit**. You can also create tables, charts, or crosstabs to see how these calculations work in views.

Absolute(*NumericExpression*)

Returns the absolute value of a number or field, that is, the non-negative value of the number.

Example:

`Absolute("Transaction Amount")` — Shows the magnitude of each transaction, regardless of whether the transaction is positive or negative.

`"Commission Rate" * Absolute("Transaction Amount")` — Computes a positive commission on all transactions, regardless of whether the transaction is positive or negative.

Attribute('*StringExpression1*', ['*StringExpression2*'])

Given the name of an attribute as the first argument and an optional category as the second argument, returns the attribute's current value as a String. If a category is specified (USER, TENANT, or SERVER), returns a categorical attribute; returns a hierarchical attribute otherwise. This value can be cast to a different datatype using one of the casting functions available in Ad Hoc: `Boolean()`, `Date()`, `Decimal()`, `Integer()`, `Time()`, or `Timestamp()`. It is your responsibility to ensure the attribute's format and values work correctly with your Ad Hoc view. See the *JasperReports Server Administrator Guide* for more information about creating and using attributes.

Average(*NumericExpression*[,'*Level*'])

Returns the average value of a measure or numeric field, based on an optional level. Null values are not included. See **Levels in Aggregate Functions** for more information.

Example:

```
Average("Salary", 'RowGroup')
```

Boolean ('*StringExpression*')

Casting function that takes a String expression and converts it to a Boolean data type. The String can be any expression that returns a supported string, including a field value or an attribute retrieved with the `Attribute()` function. The `Boolean()` function requires one of the following Strings: true, false, True, False, TRUE, or FALSE. Other Strings will return an error.

Example:

```
Boolean('true')
```

Case(*Expression*, *ValueExpression1*, *ReturnExpression1*, *ValueExpression2*, *ReturnExpression2*[,..., *ValueExpressionN*, *ReturnExpressionN*][,*DefaultReturnExpression*])

Takes 2N+1 or 2N+2 arguments: an expression followed by one or more value expression/return expression pairs, with an optional final return expression. Compares the expression in the first argument to each value expression in order of appearance. Returns the value of the expression immediately following the first value expression that matches. If no expression matches, returns the final `DefaultReturnExpression` if present, null otherwise.

The types of all the return expressions must be compatible. For example, you can't mix numeric and text return value types.

Example:

```
Case("Shipped by", 1, 'FedEx', 2, 'UPS', 3, 'USPS', 'Unknown')
```

CaseRange (*NumericExpressionInput*, *NumericExpression1*, *ReturnExpression1*, *NumericExpression2*, *ReturnExpression2*[,..., *NumericExpressionN*, *ReturnExpressionN*][,*DefaultReturnExpression*])

Takes 2N+1 or 2N+2 arguments: an expression followed by one or more numeric expression/return expression pairs, with an optional final return expression. Finds the first numeric expression that is greater than the input expression and returns the value of the corresponding return expression. If no expression is greater, returns the final `DefaultReturnExpression` if present, null otherwise.

The types of all the return expressions must be compatible. For example, you can't mix numeric and text return value types.

Examples:

```
Case("Temperature", 60, 'too cold', 80, 'just right', 'too hot')
```

```
CaseRange(CountAll("Shipping charge") % CountAll("Shipping charge", 'Total'), 2.0, 'Less
than 2%', 5.0, '2% - 5%', 'More than 5%')
```

CaseWhen(*BooleanExpression1*, *ReturnExpression1*, *BooleanExpression2*, *ReturnExpression2*[,..., *BooleanExpressionN*, *ReturnExpressionN*][, *DefaultReturnExpression*])

Takes 2N or 2N+1 arguments: one or more pairs of Boolean expressions followed by return expressions, with an optional final return expression. Returns the expression immediately following the first true Boolean expression.

If no expression is true, returns the final `DefaultReturnExpression` if present, null otherwise. This is the most flexible construct.

The types of all the return expressions must be compatible. For example, you can't mix numeric and text return value types.

Example:

```
CaseWhen("Shipped by" == 1, 'FedEx', "Shipped by" == 2, 'UPS', "Shipped by"== 3, 'USPS',
'Unknown')
```

```
Case("Temperature" <= 60, 'too cold', "Temperature" > 80, 'too hot', 'just right')
```

Concatenate(*TextExpression1*[,*TextExpression2*,...,*TextExpressionN*])

Combines multiple text strings and/or fields into a single text field. Text strings are enclosed in single quotes; labels for fields or measures in Ad Hoc are enclosed in straight quotes.

Examples:

```
Concatenate("Last Name", ' , ', "First Name")
```

```
Concatenate("Product Category", ' -- ', "Product Name")
```

Contains(*TextExpression1* ,*TextExpression2*])

Boolean that returns true if the first string contains the second, false otherwise.

Examples:

```
Contains("Product Name", 'Soda')
```

CountAll(*Expression*[,'*Level*')]

Returns the count of non-null items in a field or measure; note that `CountAll` always returns a non-negative integer. `Level` can be one of the following: `Current` (default), `ColumnGroup`, `ColumnTotal`, `RowGroup`, `RowTotal`, `Total`. See **Levels in Aggregate Functions** for more information.

Example:

`CountAll("Transaction Amount", 'RowGroup')` — Counts the total number of non-null transactions in the specified group.

CountDistinct(*Expression*[,'*Level*'])

Returns the distinct count of non-null items in the input. Always returns a non-negative integer. `Level` can be one of the following: `Current` (default), `ColumnGroup`, `ColumnTotal`, `RowGroup`, `RowTotal`, `Total`. See **Levels in Aggregate Functions** for more information.

Example:

`CountDistinct("Customer Name", 'Total')`— Counts the number of distinct customers.

Date('*StringExpression*')

Casting function that takes a String expression and converts it to a Date data type. The String can be any expression that returns a supported string, including a field value or an attribute retrieved with the `Attribute()` function. The `Date()` function requires a String value formatted as 'yyyy-MM-dd'. Other Strings will return an error.

Example:

```
Date('2015-07-17')
```

Decimal('*StringExpression*')

Casting function that takes a String expression and converts it to a Decimal data type. The String can be any expression that returns a supported string, including a field value or an attribute retrieved with the `Attribute()` function. The `Decimal()` function requires a String value in decimal format, for example, '123.45'. Other Strings will return an error.

Example:

```
Decimal('1234.567')
```

DayName (*DateExpression*)

Given a date field, returns a text field with the name of the day of the week.

Example:

`DayName("Open Date")`— Displays the day of the week on which a store was opened.

`Mode(DayName("Open Date"), 'Total')`— The day of the week on which the most stores were opened.

DayNumber (*DateExpression*)

Numeric field that returns the day of the month from a date field.

`DayNumber("Open Date")`— Displays the day of the month on which a store was opened.

ElapsedDays (*DateExpression1*, *DateExpression2*)

Calculates the number of days elapsed between two date fields that contain time values.

Example:

```
ElapsedDays ("Date shipped","Date required")
```

ElapsedHours (*DateTimeExpression1*, *DateTimeExpression2*)

Calculates the number of hours elapsed between two date fields that contain time values.

Example:

```
ElapsedHours ("Date shipped","Date required")
```

This replaces the two-date custom field operation **Date Difference > Hours** available in Ad Hoc views created in Server 5.5 or earlier.

ElapsedMinutes (*DateTimeExpression1*,*DateTimeExpression2*)

Calculates the number of minutes elapsed between two date fields that contain time values.

Example:

```
ElapsedMinutes ("Date shipped","Date required")
```

This replaces the two-date custom field operation **Date Difference > Minutes** available in Ad Hoc views created in Server 5.5 or earlier.

ElapsedMonths (*DateTimeExpression1*,*DateTimeExpression2*)

Calculates the number of months elapsed between two date fields that contain time values.

Example:

```
ElapsedMonths ("Date shipped","Date required")
```

ElapsedQuarters (*DateExpression1*, *DateExpression2*)

Calculates the number of quarters elapsed between two date fields.

Example:

```
ElapsedQuarters ("Date shipped","Date required")
```

ElapsedSeconds (*DateTimeExpression1*,*DateTimeExpression2*)

Calculates the number of seconds elapsed between two date fields that contain time values.

Example:

```
ElapsedSeconds ("Date shipped","Date required")
```

This replaces the two-date custom field operation **Date Difference > Seconds** available in Ad Hoc views created in Server 5.5 or earlier.

ElapsedSemis (*DateExpression1*,*DateExpression2*)

Calculates the number of semi-years elapsed between two date fields.

Example:

```
ElapsedSemis ("Date shipped","Date required")
```

ElapsedWeeks (*DateExpression1*,*DateExpression2*)

Calculates the number of weeks elapsed between two date fields that contain time values.

Example:

```
ElapsedWeeks ("Date shipped","Date required")
```

This replaces the two-date custom field operation **Date Difference > Weeks** available in Ad Hoc views created in Server 5.5 or earlier.

ElapsedYears (*DateExpression1*,*DateExpression2*)

Calculates the number of years between two date fields.

Example:

```
ElapsedYears ("Date shipped","Date required")
```

This replaces the two-date custom field operation **Date Difference > Years** available in Ad Hoc views created in Server 5.5 or earlier.

EndsWith(*TextExpression1*, *TextExpression2*]

Boolean that returns true if the first text input ends with the string specified in the second input; false otherwise.

Examples:

```
EndsWith("Product Name", 's')
```

IF (*BooleanExpression*, *ExpressionWhenTrue*[, *ExpressionWhenFalse*])

Given a Boolean field or calculation as the first argument, returns the second argument if true; optionally returns the third argument if false. Returns null if the first argument is null.

`ExpressionWhenFalse` must be of the same type as `ExpressionWhenTrue`. For example, if `ExpressionWhenTrue`is a date, `ExpressionWhenFalse` must be a date in the same format. If `ExpressionWhenFalse` is not set, then a false result returns a null value.

You can create a `BooleanExpression` using the comparison operators(“==”, “!=”, “>”, “>=”, “<”, “<=”); any functions that return Boolean values (`StartsWith`, `EndsWith`, `IsNull`, `Contains`) and logical operators (`and`, `or`, `not`).

When dates are used in comparisons or the IF function, they must be the same type, (date only, date/time, or time only). Make sure to use the correct modifier (d, ts, t) when using date constants in comparisons.

Example:

`IF(Contains("Product Name", 'Soda'), 'Yes', 'No')`— Uses the Contains function to see whether the product name contains the string "Soda"; if it does, sets the field value to Yes.

Integer('*StringExpression*')

Casting function that takes a String expression and converts it to a Integer data type. The String can be any expression that returns a supported string, including a field value or an attribute retrieved with the `Attribute()` function. The `Integer()` function requires a String value that can be read as an integer, such as '123'. Other Strings will return an error.

Example:

```
Integer('123')
```

IsNull(*Expression*)

Boolean that returns true if the field value is null; false otherwise.

Examples:

```
IsNull("First Name")
```

Length(*TextExpression*)

Given a text string, returns its length. Null values return null.

Examples:

```
Length("First Name")
```

Max (*NumericExpression*|*DateExpression*[,'*Level*'])

Returns the maximum value reached by the specified field or calculation. `Level` can be one of the following: `Current` (default), `ColumnGroup`, `ColumnTotal`, `RowGroup`, `RowTotal`, `Total`. See **Levels in Aggregate Functions** for more information.

Example:

```
Max("Salary")
```

Median (*NumericExpression|DateExpression*[,'*Level*'])

For an odd number of values, returns the middle value after all values are listed in order. For an even number of values, returns the average of the middle two values. For example, if a field has only five instances, with values {1,1,3,10,20}, the median is 3. `Level` can be one of the following: `Current` (default), `ColumnGroup`, `ColumnTotal`, `RowGroup`, `RowTotal`, `Total`. See **Levels in Aggregate Functions** for more information.

Example:

```
Median("Salary")
```

Mid (*TextExpression,Integer1,Integer2*)

Given a text string, returns the substring starting at Integer1 with length Integer2.

Example:

`Mid("Phone", 1, 3)` — Given an American phone number starting with a 3-digit area code, extracts the area code.

Min (*NumericExpression|DateExpression*[,'*Level*'])

Returns the minimum value reached by the specified field or calculation based on an optional level. `Level` can be one of the following: `Current` (default), `ColumnGroup`, `ColumnTotal`, `RowGroup`, `RowTotal`, `Total`. See **Levels in Aggregate Functions** for more information.

Example:

```
Min("Salary")
```

Mode (*Expression*[,'*Level*'])

Returns the most frequent value reached by the specified input, based on an optional level. For example, if a field has only five instances with values {1,2,2,4,5}, the mode is 2. `Level` can be one of the following: `Current` (default), `ColumnGroup`, `ColumnTotal`, `RowGroup`, `RowTotal`, `Total`. See **Levels in Aggregate Functions** for more information.

Example:

`Mode (DayName ("Order Date",RowGroup))` — For each row group, returns the day of the week on which the most orders were placed.

MonthName (*DateExpression*)

Returns a text field with the name of the month.

Example:

```
MonthName ("Order Date" )
```

MonthNumber (*DateExpression*)

Returns the number of the month, with January = 1 and December = 12. Null values return null.

Example:

```
MonthNumber ("Order Date")
```

PercentOf (*NumericExpression*[,'*Level*'])

Returns the value as a percent of total for the specified level. Null values are ignored. Note that possible values for Level are `ColumnGroup`, `ColumnTotal`, `RowGroup`, `RowTotal`, `Total` (default). See **Levels in Aggregate Functions** for more information.

`PercentOf (NumericExpression, Total)` replaces the custom field calculation **Percent of Total Group** available in Ad Hoc views created in Server 5.5 or earlier.

Calculated fields using the `PercentOf` function should not be used as filters; if `PercentOf` is used as a filter, then the total percent may not be 100.

Example:

```
PercentOf("")
```

Range (*NumericExpression*[,'*Level*'])

The difference between the largest and smallest values of the given input.

Example:

```
Range("Salary",'ColumnGroup')
```

Rank (*NumericExpression*)

Returns the position of each value relative to the other values after all the values are listed in order. For example, the top ten in sales are the top ten in rank. Null values are ignored.

Example:

```
Rank("Store Sales")
```

Round (*NumericExpression*[,*Integer*])

Rounds a number to a specified number of digits; default is zero (0) digits. Decimal values greater than 0.5 are rounded to the next largest whole number, and values less than 0.5 are rounded down.

Example:

```
Round("Sales")
```

StartsWith(*TextExpression1*, *TextExpression2*]

Boolean that returns true if the first text input starts with the string specified in the second input; false otherwise.

Examples:

```
StartsWith("Product Name", 'Q')
```

StdevP (*NumericExpression*[,'*Level*'])

Standard deviation based on the entire population, taken over the values at the specified (optional) level. Null values are excluded. `Level` can be one of the following: `Current` (default), `ColumnGroup`, `ColumnTotal`, `RowGroup`, `RowTotal`, `Total`. See **Levels in Aggregate Functions** for more information.

Examples:

```
StdevP("Sales",'RowTotal')
```

StdevS (*NumericExpression*[,'*Level*'])

Standard deviation based on a sample, taken over the values at the specified level. Null values are excluded. `Level` can be one of the following: `Current` (default), `ColumnGroup`, `ColumnTotal`, `RowGroup`, `RowTotal`, `Total`. See **Levels in Aggregate Functions** for more information.

Example:

```
StdevS("Sales",'RowTotal')
```

Sum (*NumericExpression*[,'*Level*'])

The sum of all values in the range. Null values are excluded. `Level` can be one of the following: `Current` (default), `ColumnGroup`, `ColumnTotal`, `RowGroup`, `RowTotal`, `Total`. See **Levels in Aggregate Functions** for more information.

Examples:

```
Sum("Sales",'RowGroup')
```

Time('*StringExpression*')

Casting function that takes a String expression in the format 'HH:mm:ss.SSS' and converts it to a Time data type. The String can be any expression that returns a valid String, including a field value or an attribute retrieved with the `Attribute()` function.

Example:

```
Time('17:12:33:147')
```

Timestamp('*StringExpression*')

Casting function that takes a String expression in the format 'yyyy-MM-dd HH:mm:ss.SSS' and converts it to a Timestamp data type. The String can be any expression that returns a valid String, including a field value or an attribute retrieved with the `Attribute()` function.

Example:

```
Timestamp('2015-07-17 17:12:33:147')
```

Today (*Integer*)

Calculates the date that is the specified number of days from the current system date.

Examples:

`Today (0)` — The current system date.

`Today(1)` — The day after the current system date.

`Today(-1)` — The day before the current system date.

WeightedAverage (*NumericExpression1*,*NumericExpression2*,'*Level*')

Returns the weighted average for the first input weighted with respect to the second input, calculated at an optional level. Null values are excluded. `Level` can be one of the following: `Current` (default), `ColumnGroup`, `ColumnTotal`, `RowGroup`, `RowTotal`, `Total`. See **Levels in Aggregate Functions** for more information.

Examples:

`WeightedAverage ("Price","Units", 'Current')` — The extended price based on the number of units.

`WeightedAverage ("Price","Units", 'RowGroup')` — The sum of the extended price for all units in the row group.

Year (*DateExpression*)

Given a date field, returns the year.

Examples:

`Year("Order Date" )`

4.6.5 Calculated Field Syntax

This section describes the syntax required when creating calculated fields in Ad Hoc views. See **Calculated Field Reference** for more information.

Use the following syntax for inputs:

- To reference a text string, use single quotes (') — `'`*`Text String`*`'`.
- To reference a field label, use double quotes (") — `"`*`Ad Hoc Label`*`"`.
- To reference date constants, indicate the date type as part of the syntax, as listed below:
  - To reference a date without time data (for example: yyyy-dd-mm), use `d` followed by single quotes (') — `d'`*`2014-06-10`*`'`
  - To reference a date with day and time data (for example: yyyy-dd-mm hh:mm:ss), use `ts` followed by single quotes (') — `ts'`*`2014-06-10 01:30:00`*`'`. If you use `ts` and enter the date information only, the time is automatically set to `00:00:00`.
  - To reference a date with time data only (hh:mm:ss), use `t` followed by single quotes (') — `t'`*`01:30:00`*`'`
- To reference a date field label, use double quotes (") — `"`*`Ad Hoc Date Field Label`*`"`.

The following are reserved words and cannot be used as field names: AND, And, and, IN, In, in, NOT, Not, not, OR, Or, or. Names containing these strings, such as "Not Available", can be used.

When dates are used in comparisons or the IF function, they must be the same type, (date only, date/time, or time only). Make sure to use the correct modifier (d, ts, t) when using date constants in comparisons.

In the function descriptions for calculated fields in **Calculated Field Reference**, the argument name describes the type of input the function accepts. For more information about input types, see **Datatypes**:

- `BooleanExpression` — Any expression that takes on Boolean values, including the label of a Boolean field or measure, a Boolean calculation, or a Boolean value.

You can create a BooleanExpression using the following: comparison operators (==, !=, >, >=, <, <=, in); functions that return Boolean values (StartsWith, EndsWith, IsNull, Contains) and logical functions (AND, OR, NOT).

- `DateExpression` — Any type of date or timestamp values, including the label of a date field or measure, or a calculation that returns dates.
- `DateTimeExpression` — Date expressions that contain time values, including the label of a date field or measure, or a calculation that returns dates. These values are also known as timestamp values.
- `Expression` — Any valid date, date-time, numeric, or string expression.

- `NumericExpression` — Numeric values, including the label of a numeric field or measure, or a calculation that returns numbers.
- `TextExpression` — Text values, including the label of a text field or measure, or a text string.
- `Level` — For aggregate functions, specifies the set of values used to compute the calculation. Possible values include Current (not available for PercentOf), ColumnGroup, ColumnTotal, RowGroup, RowTotal, Total. See **Levels in Aggregate Functions** for more information.

4.6.6 Operators in Ad Hoc Views

Ad Hoc views support the following operators in calculated fields. Operators are evaluated in the order they are shown in the table:

Operator	Syntax	Description
multiply, divide	`i * j / k`	Arithmetic operators for numeric types only.
percent	`i % j`	Calculates i as a percent of j; numeric types only.
add, subtract	`i + j - k`	Arithmetic operators for numeric types only.
equal	`i == j`	Comparison operators for string, numeric, and date types.
not equal	`i != j`	
less than	`i < j`	Comparison operators for numeric and date types only.
less than or equal	`i <= j`	
greater than	`i > j`	
greater than or equal	`i >= j`	
IN *set*	`i IN ('apples','oranges')`	Sets can be of any type.
IN *range*	`i IN (j:k)`	Ranges must be numeric or date types.
NOT	`NOT( i )`	Boolean operators. Parentheses are required for `NOT`.
AND	`i AND j AND k`	
OR	`i OR j OR k`	
parentheses	`()`	Used for grouping.

When dates are used in comparisons or IF functions, they must be the same type, (date only, date/time, or time only). Make sure to use the correct modifier (d, ts, t) when using date constants in comparisons.

The following reserved words cannot be used as field names: AND, And, and, IN, In, in, NOT, Not, not, OR, Or, or. Names containing these strings, such as "Not Available", can be used.

4.6.7 Aggregate Functions

Aggregate functions in calculated fields perform calculations based on groups of rows, rather than on single rows. For example, it doesn't make sense to use Sum or Average on a single value; instead, you want to take the sum or average over a row or column group or over the total set. In many cases, aggregate functions in the Ad Hoc Editor are analogous to SQL functions that can be used with the GROUP BY clause in a SELECT statement.

The aggregate functions are as follows:

Average
CountAll
CountDistinct
Max
Median
Min
Mode
PercentOf
Range
StdDevP
StdDevS
Sum
WeightedAverage

Because aggregate functions already operate on groups, their use is restricted in the following ways:

- You can use aggregate functions only in calculated measures; aggregates should not be used to create non-measure fields.
- You cannot add an aggregate function to a group.
- You should not use an aggregate function as a filter.
- Only AggregateFormula, Custom, or None are supported as summary calculations for aggregate functions. Custom appears in the **Change Summary** right-click menu only if you have defined a custom function in the **Create Calculated Field** dialog box.

4.6.7.1 Levels in Aggregate Functions

Many aggregate functions accept an optional level to specify the grouping of the aggregate. A level used in an aggregate, must be enclosed in straight quotes ('), for example, `'RowGroup'`.

The available levels are as follows:

- `Current` (default) — use the current value when at a looking at detail rows in a table view.
- `RowGroup` — use the parent values from a row location.
- `RowTotal` — use the grand total value from a row location.
- `ColumnGroup` — use the parent values from a column location.
- `ColumnTotal` — use the grand total value from a column location.
- `Total` — use the grand total value from a cross tab and the RowTotal from a Table.

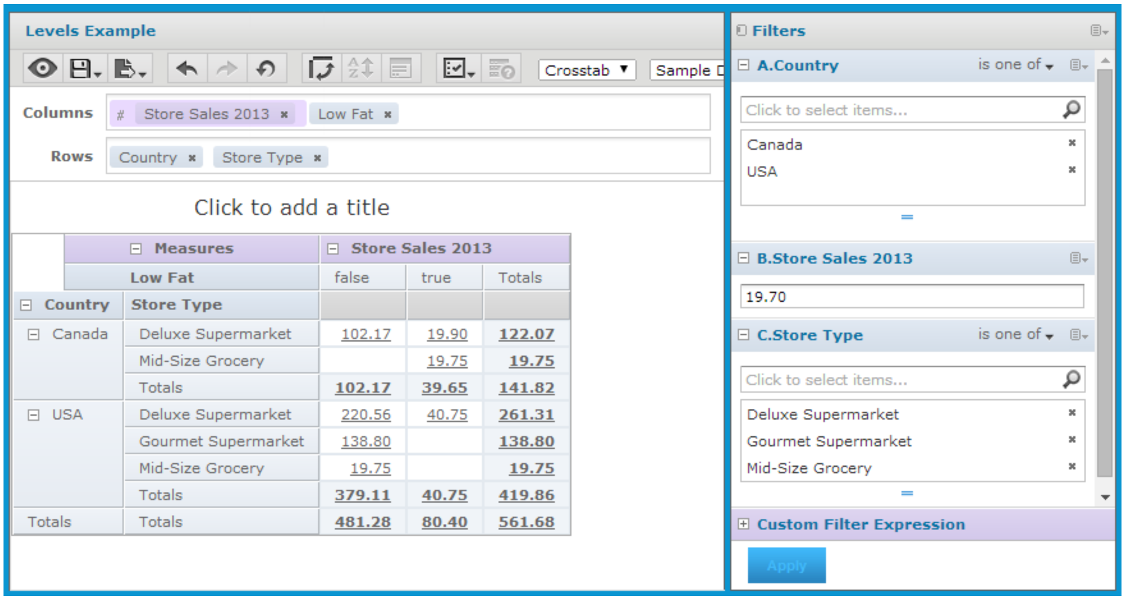
The following example shows how RowGroup works with the PercentOf() function.

Setting up the Ad Hoc view:

The examples in this section uses the Ad Hoc view created in **Creating a Calculated Field**. The initial view for these examples can be set up as follows:

1. Select Crosstab from the View menu.
2. Add Store Sales 2013 and Low Fat to the Columns entry bar.
3. Add Country and Store Type to the Rows Entry Bar.
4. To make the example clearer, the data has been restricted. To do this, create three filters:
   a. Expand Regions in the Fields Picker, right-click on Country, and select **Create Filter**. In the Filters pane, set the filter to **is one of** then select Canada and USA.
   b. Create a filter for Store Sales 2013. In the Filters pane, set the filter to **is greater than**, and enter 19.70.

c. Right-click on Store Type and select **Create Filter**. In the Filters pane, set the filter to **is one of** then select Deluxe Supermarket, Gourmet Supermarket, and Mid-Size Grocery.



	⊟ Measures	⊟ Store Sales 2013		
	Low Fat	false	true	Totals
⊟ Country	Store Type			
⊟ Canada	Deluxe Supermarket	102.17	19.90	122.07
	Mid-Size Grocery		19.75	19.75
	Totals	102.17	39.65	141.82
⊟ USA	Deluxe Supermarket	220.56	40.75	261.31
	Gourmet Supermarket	138.80		138.80
	Mid-Size Grocery	19.75		19.75
	Totals	379.11	40.75	419.86
Totals	Totals	481.28	80.40	561.68

⊟ A.Country is one of

Click to select items...

Canada

USA

⊟ B.Store Sales 2013

19.70

⊟ C.Store Type is one of

Click to select items...

Deluxe Supermarket

Gourmet Supermarket

Mid-Size Grocery

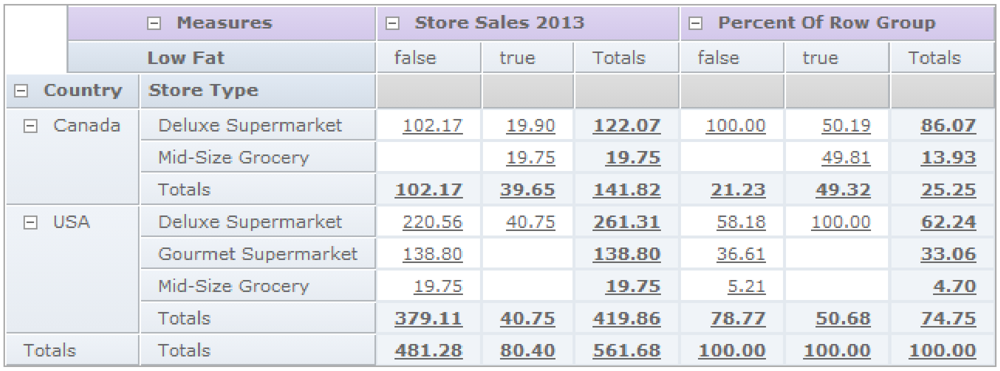
Figure 4-12 Base Example for Levels

RowGroup example:

To create the layout for this example:

1. Click ▤▾ at the top right of the **Measures** section of the Data Source Selection panel and select **Create Calculated Measure...** from the context menu.
2. In the Create Calculated Measure dialog box, enter Percent of Row Group for the name, and enter `PercentOf("Store Sales 2013",'RowGroup')` in the Formula entry box. Then click **Create Measure**.
3. Drag the measure you just created to the Columns bar, between Store Sales 2013 and Low Fat.

The crosstab appears as shown in the following figure.



	⊟ Measures	⊟ Store Sales 2013			⊟ Percent Of Row Group		
	Low Fat	false	true	Totals	false	true	Totals
⊟ Country	Store Type						
⊟ Canada	Deluxe Supermarket	102.17	19.90	122.07	100.00	50.19	86.07
	Mid-Size Grocery		19.75	19.75		49.81	13.93
	Totals	102.17	39.65	141.82	21.23	49.32	25.25
⊟ USA	Deluxe Supermarket	220.56	40.75	261.31	58.18	100.00	62.24
	Gourmet Supermarket	138.80		138.80	36.61		33.06
	Mid-Size Grocery	19.75		19.75	5.21		4.70
	Totals	379.11	40.75	419.86	78.77	50.68	74.75
Totals	Totals	481.28	80.40	561.68	100.00	100.00	100.00

Figure 4-13 Example for RowGroup

Look at the Low Fat values for "false" in the Canada group. The only non-null value under Store Sales 2013 is in the first row, for Deluxe Supermarket. This value, 102.17, is 100% of the row group total of 102.17 on the third line of the crosstab. This percentage is shown in the "false" subcolumn of the Percent of Row Group column.

Compare this to the "true" entry under Store Sales for the same (first) row. There, the value is 19.90, and the row group total is 39.65. The corresponding percentage shown in the "true" subcolumn of Percent of Row Group is 50.19%.

4.6.8 Summary Calculations

Summary calculations are aggregate functions used for sub-totals and totals. Summary calculations can be set in the Domain Designer or in the Ad Hoc view.

- In Ad Hoc table views, each field can display a single summary calculation. The summary calculation is automatically applied to all groups in the table. Summaries appear at the bottom of each group, as well as at the bottom of the view. When a new group is added, it includes a summary for each column.
- In crosstabs, each measure displays a summarized value. Summaries determine the values of the Totals at the intersection of each row and column.
- In charts, the type of chart determines whether measures are summarized. If summaries are used, they determine the size or location of the graphical elements that represent your data.

For dual pie charts, the summary function for the field needs to be Sum or CountAll. Dual pie charts with other summary functions may give unexpected results.

In general, you can change the summary calculation of any measure. By default, JasperReports Server summarizes fields of each datatype as shown in the following table.

Table 4-3 Default Summary Functions in Calculated Fields

Datatype	Default Summary Calculation	Description
Numeric	Sum	Displays the sum of all values in the set.
Date	CountAll	Displays the total number of values in the set.
String	CountAll	Displays the number of values in the set.
Boolean	CountAll	Displays the number of values in the set.
Aggregate	AggregateFormula	For a calculated field that uses an aggregate function, uses the same aggregate formula as the summary.
Combined	None	For a calculated field that combines an aggregate function with a non-aggregate function, the summary calculation is null.

Select from the following options to set a measure’s summary function in any type of view.

Function	Meaning	Available for....
AggregateFormula	For a calculation that uses an aggregate function, uses the same aggregate formula as the summary.	Aggregate
Average	Displays the average of all values in the set.	Numeric
CountAll	Displays the number of rows in the set.	Boolean, Date, Numeric, and String
CountDistinct	Displays the number of unique values in the set.	Boolean, Date, Numeric, and String
Custom	Allows you to enter an aggregate calculation for the summary. Only available for calculated fields and measures in the Ad Hoc Editor; not available for Domains.	Aggregate, Date, Numeric
Max	Displays the highest value in the set.	Date, Numeric
Median	Displays the median value of the set.	Date, Numeric
Min	Displays the lowest value in the set.	Date, Numeric
Mode	Displays the value that occurs most frequently in the set.	Boolean, Date, Numeric, and String
None	The aggregate function is null; no summary function is displayed.	Aggregate, Boolean, Date, Numeric, and String
Range	Displays the difference between the minimum and maximum values of the set.	Numeric
RangeDays	Displays the difference in days between the minimum and maximum values of the set.	Date
RangeHours	Displays the difference in hours between the minimum and maximum values of the set.	DateTime
RangeMinutes	Displays the difference in minutes between the minimum and maximum values of the set.	DateTime
RangeMonths	Displays the difference in months between the minimum and maximum values of the set.	Date

Function	Meaning	Available for….
RangeQuarters	Displays the difference in quarters between the minimum and maximum values of the set.	Date
RangeSemis	Displays the difference in semi-annual periods between the minimum and maximum values of the set.	Date
RangeWeeks	Displays the difference in weeks between the minimum and maximum values of the set	Date
RangeYears	Displays the difference in years between the minimum and maximum values of the set.	Date
Aggregate Formula	Uses the aggregate formula used to define the calculated field as the summary function, and sets the level appropriately for the context.	Aggregate
StdDevP	Displays the standard deviation for the population of the set.	Numeric
StdDevS	Displays the standard deviation on a sample for the set.	Numeric
Sum	Displays the grand total for the set.	Numeric
WeightedAverage	Displays the weighted average for the set, based on a second numeric field or expression. Only available for calculated fields and measures in the Ad Hoc Editor; not available for Domains.	Numeric

Note the following about summaries:

- If you select a summary calculation other than the default, that calculation is shown in parentheses after the field name in the fields picker.
- In Ad Hoc views, you see special behavior when you create a calculated field or measure with the following type of summary calculation:
  - If you create a Custom summary calculation for a field or measure, **Custom** is available on the **Change Summary Function** menu for that field. It is not available otherwise.
  - If you create a WeightedAverage summary calculation for a field or measure, **WeightedAverage** is available on the **Change Summary Function** menu for that field. It is not available otherwise.
- You can remove summaries by setting the summary function to **None**.
- Only AggregateFormula, Custom, or None are supported as summary calculations for aggregate functions. Custom only appears in the **Change Summary** right-click menu if you have defined a custom function in the **Create Calculated Field** dialog box.

4.7 Using Filters and Input Controls

JRXML Topics, Domains, and OLAP connections use different mechanisms for screening the data they return:

- JRXML Topics can contain Parametrized queries. The parameters can be mapped to input controls that allow users to select the data they want to include.
- Domains (and Domain Topics) can be filtered by selecting fields in the Domain and specifying comparison values. The filters can be configured for users to select the data to include.
- Within the Domain design, filters based on conditions can also be defined; these filters are not displayed in the report viewer when the report runs.
- OLAP connections rely on XML schemas to filter the data in an underlying transactional database. Input controls are never generated directly from the OLAP schema.

You can define filters in the Ad Hoc Editor regardless of whether you are working with data from a Domain, Topic, or OLAP connection. Such filters can be helpful in improving the view's initial performance by reducing the amount of data the view returns by default. For more information, see **“Input Controls and Filters Availability” on page 151**. To prevent users from seeing the full dataset, you can also use input controls in a JRXML Topic or filters defined in the Domain design, which can be hidden from end users.

If you want to return different data to different users of the same view, define data-level security based on a user roles and profile attributes. This is available for reports based on a Domain, Domain Topic, or OLAP client connection. For more information, refer to the *Jaspersoft OLAP User Guide*.

Input controls and filters interact seamlessly. For example, you can create filters in an Ad Hoc view that gets data from a JRXML Topic that includes input controls.

The server refreshes the editor against both the filters and the input controls. Because some combinations

of input controls and filters don’t return data, this can result in an empty view.

If the result set is empty, check for an incompatible combination of filters and input controls, such as a standard filter (set to Mexico against a Country field) and a **Keep Only** filter (set to Canada against a Country field), or an incorrectly-defined advanced filter expression (data must meet all criteria in multiple filters, rather than meeting criteria in a subset of those filters). See **“Custom Filtering” on page 147** for information on advanced filter expressions.

In rare cases, filters can conflict with view parameters, and you’ll need to rename the field causing the conflict by editing the JRXML file. Refer to the *Jaspersoft Studio User Guide* for more information about editing JRXML files.

For more information about:

- JRXML Topics and input controls, see **“Adding Input Controls” on page 176**.
- Domain Topic filters, see **“Creating Topics from Domains” on page 159**.
- Localizing input control prompts and lists of values, see **“Localizing Reports” on page 190**.

4.7.1 Using Filters

Filters can be defined at three levels:

- In the Domain Designer.
- When creating a view from a Domain (with the Data Chooser).
- In the Ad Hoc Editor (even when the view is based on a JRXML Topic or OLAP connection).

In this section, we discuss how to define filters in the Ad Hoc Editor. For information on defining filters in the Domain Designer, see **"The Pre-filters Tab" on page 228**. For information on defining filters in the Data Chooser, see **"The Pre-filters Page" on page 155**.

In addition, you can control how and what filters are applied to a field or fields by using custom expressions. For more information, see **"Custom Filtering" on page 147**.

To create a filter in the Ad Hoc Editor:

1. Right-click a field in the **Data Source Selection** panel and select **Create Filter**. A new filter appears in the Filters panel. If the Filters panel was hidden, it appears when you create a new filter.

> If the results are empty and you don't understand why, check for an incompatible combination of filters and input controls. Click ▤ to compare input controls with the filters in the Filters panel.

2. Use the fields in the filter to change its value. Depending on the datatype of the field you selected, the filter maybe multi-select, single-select, or text input.
3. Click ▤▾ and select **Minimize All Filters** or **Maximize All Filters** to toggle expansion of the items in the filter.
4. Click ▤▾ and select **Remove All Filters** to remove the filters.
5. Click ▼ to hide the filter's details. Click ▶ to display them again.
6. Click the **Select All** check box (if it appears in the Filters panel) to select all values currently available in the dataset. The **Select All** check box does not appear in the Filters panel for numbers and dates.

   Note that the **Select All** check box doesn't guarantee that all values are selected every time the report runs. Instead, the check box is a shortcut to help you quickly select all the values currently available in the dataset. To ensure that all values appear in the view whenever it is edited or a report is run, remove the filter entirely. On the panel, you can also create a filter from the right-click context menu of a column in a table. On the **Chart** tab, you must right-click the field in the **Data Source Selection** panel.

When you change a filter, the server uses the filter's new value to determine what data to display. If you change only the operator in a filter, you must deselect the value in that filter, then reselect it to apply the updated filter.

For filters with multiple values, you do not need to reselect all values. After changing the operator, use Ctrl-click to deselect just one of the values, then Ctrl-click to reselect that value.

4.7.1.1 Relative Dates

You can filter information in your view based on a date range relative to the current system date. You can accomplish this using date-based filters, and entering a text expression describing the relative date or date span you want to display, using the format `<Keyword>+/-<Number>` where:

- **Keyword** indicates the time span you want to use. Options include: DAY, WEEK, MONTH, QUARTER, SEMI, and YEAR.
- **+** or **-** indicates whether the time span occurs before or after the chosen date.
- **Number** indicates the number of the above-mentioned time spans you want to include in the filter.

For example, if you want to look at all Sales for the prior week, your expression would be: WEEK-1.

To create a relative date filter:

1. Following the instructions in **"Using Filters" on page 145**, create a filter based on a date field. The filter appears in the Filters panel.

2. In the filter’s first text entry box, enter an expression describing the relative date or date span you want to display.
3. In the filter’s second text entry box, enter the date you want to base your filter on.

   For instance, if you want to display all the sales numbers for one month before the current date, enter MONTH-1 in the first text entry box, and enter today's date in the second box.

When you right-click a group member in a crosstab and select **Keep Only** or **Exclude**, you create complex filters. When you create a filter against an inner group, the filter that appears may be created as a complex filter. You can't edit a complex filter, but you can remove it. Complex filters also appear in the Ad Hoc Editor if a Data Chooser filter was created and locked.

4.7.1.2 Custom Filtering

When you create multiple filters they are, by default, connected with an implicit AND operator. That is, the data displayed in your table, chart, or crosstab is what remains after all your filters are applied.

However, with the custom filter functionality, you can exercise greater control over the displayed data by applying a custom expression that includes more complex, nested AND, OR, and NOT operators, as well as by applying multiple filters to a single field.

Custom filters are not available for Ad Hoc views created from OLAP connections.

Custom filters are useful in a number of situations, including:

- **When using the AND operator isn’t sufficient**. Consider an international company that wants to view data for stores located on the Pacific Rim; they may create a custom expression with the following criteria:
  - Country is USA
    AND
  - State is California OR Washington OR Oregon OR Hawaii OR Alaska.
    OR
  - Country is Japan OR Indonesia

  Using the AND operator for all of these criteria returns an empty view, as no store is located in all of those areas.
- **When you need to eliminate some results in a field.** For example, if your food and beverage distribution company wants to view sales for all drinks except for high-price items, you might include the following criteria in a custom expression:
  - Product Group is Beverages
    NOT
  - Price is greater than 39.99

  This filter displays all items in the Beverage Product Group, but filters out those with prices over $39.99

These are only two scenarios where custom filters can hone your results and make your view more precise. There are, of course, many other situations where they can be applied.

In this section, we take you through these tasks:

- Creating a custom expression
- Editing a custom expression
- Removing a custom expression
- Applying multiple filters to a single field

Custom filters are applied to views, but filter details don’t appear on previews or on the report generated from that view.

To create and apply a custom filter:

1. Create two or more filters for your data, as described in **“Using Filters” on page 145**. These can be standard field-based filters, or **Keep Only** and **Exclude** filters.

   Note that as you create the filters for use in a custom expression, you may find that the data in your view disappears, since most (if not all) of the data won’t meet all of the filter criteria. When you create your custom expression and change some of the ANDs to ORs and NOTs, data reappears in the panel.
2. At the bottom of the Filters panel, expand the **Custom Filter Expression** section.
3. In the text entry box, enter a filter expression using the letter designations, and including the following operators:
   - **AND** narrows your results and includes only fields that meet the criteria of both filters before and after the operator.
   - **OR** broadens your results and includes fields that meet the criteria of either filter before or after the operator.
   - **NOT** excludes results that match the criteria.
   - **Parentheses** combines multiple filters into a single item in the expression.

   Filter letter designations are case sensitive, and must be UPPERCASE.
4. Click **Apply**. Your view is updated to reflect the newly-applied filter criteria.

After creating a custom filter, you may want to add another filter to the expression or remove one already included in the expression.

If the simple filter you want to delete is part of a custom filter, you must first remove it from the custom filter expression; otherwise deleting the filter deletes the custom filter expression.

To add a new filter to an existing custom expression:

1. If necessary, create the new filter in the **Filter** panel.
2. In the **Custom Filter Expression** section, click inside the text entry box to edit the expression.
3. Add the new filter to the expression.
4. Click **Apply** to apply the new filter criteria.

To remove a filter from a custom expression:

1. Expand the **Custom Filter Expression** section.
2. In the text entry box, remove the unwanted filter from the expression and adjust the expression as needed.
3. Click **Apply** to apply the new filter criteria.

When working with custom expressions, you may decide to delete an expression and create a new one.

To remove a custom expression from a view:

1. Expand the **Custom Filter Expression** section.
2. Clear the expression from the text entry box.
3. Click **Apply**. The expression is removed, leaving the remaining filters intact.

When you refine your custom expression, you may also want to delete unused filters from the Filters panel.

- If the filter you want to remove isn't part of the custom filter, hover your mouse over ▤▾ in the filter's title bar and select **Remove Filter**.
- If you want to remove all existing filters, including the custom expression, hover your mouse over ▤▾ in the upper right corner of the Filters panhandle and select **Remove All Filters**.

You can apply multiple simple filters to a single field, if needed, to further refine your custom filter results. For example, a user may want to view the data in the Shipping Cost field, but only when it meets certain criteria combinations:

- When shipping costs to French cities with postal codes that begin with the number 5 are under five Euros
- When shipping costs to German cities with postal codes that begin with the number 1 are under five Euros

You can recreate the scenario below using the **demo for adhoc** topic.

In the following example a user has a table including the following columns:

- Country
- Postal Code
- Shipping Charge

To analyze the specific shipping costs described above, the user creates the following (simple) filters - including two filters each for the Country and Postal code fields:

- A. Country *equals* France
- B. Postal code *starts with* 5
- C. Country *equals* Germany
- D. Postal code *starts with* 1
- E. Shipping Charge *is less than* 5

Then, to display only the information she needs, she creates the following custom expression:

- ((A and B) or (C and D)) and E

This translates to:

- ((FRANCE and POSTAL CODES THAT START WITH 5) or (GERMANY and POSTAL CODES THAT START WITH 1)) and SHIPPING CHARGES LESS THAN 5 EUROS.

4.7.2 Using Input Controls

In the Ad Hoc Editor, you can display the input controls defined in the Topic as visible to users. You can accept the controls' default values or enter other values. The Ad Hoc Editor indicates that the view has input controls by displaying ▤ as active on the tool bar. Click this icon to select new values or to save values as the new defaults for this view.

There are two types of input controls: Single select and multi-select. The input control type is determined by the operator you use. In turn, the available operators are determined by the field type (date, text, numeric, or boolean) you use as a filter.

Single select controls present a calendar or drop-down list of values, from which you can choose a single value. To create this type of input control select one of the following operators:

- equals
- is not equal to

- is greater than
- is less than
- is greater or equal to
- is less or equal to
- contains
- does not contain
- starts with
- does not start with
- ends with
- does not end with
- is before
- is after
- is on or before
- is on or after

Multi-select controls display a calendar or drop-down list of values, from which you can choose multiple values. You can click to select individual values or shift-click to select multiple sequential values. You can also search for values, ✓ select all available values, ✕ deselect all available values, or ⇄ invert the selection. A **Selected** tab shows only items that are selected and allows you to delete them. To create this type of input control select one of the following operators:

- is one of
- is not one of
- is between
- is not between

To add an input control to the view using a filter:

1. Create a new filter or use an existing one in the Filters panel.
2. In the Filters panel, click the operator drop-down menu in the filter's title bar.
3. Select an operator from the drop-down. The operator you select determines whether the input control is single-select or multi-select.
4. Click **Apply**. The filter appears as an input control when the view is used to run the report.
5. Place your cursor over 💾▾ , select **Save Ad Hoc View as....**
6. Name the view, select a location, and click **Save**.
7. On the tool bar, click ▤ .

   Only the input controls defined in the topic appear here. Again, if no input controls were defined in the topic, the button appears inactive. You can create a report and open it in the report viewer to see a filter listed as an input control.

To edit the values for a view’s input controls:

1. On the tool bar, click ▤. A window listing the input controls defined in the Topic appears.

   The Parametrized Report Topic already includes three input controls created when the report was uploaded: Country, RequestDate, and OrderId.

2. Select new values. For example, select USA from the **Country** drop-down.

3. To change default values of input controls, select the check box, **Set these values as defaults when saving your view**. The selected values become the default values when you save the view.
4. Click **OK**. The Ad Hoc view shows USA data.

4.7.3 Input Controls and Filters Availability

Input controls and filters can appear in the Editor and when a report runs:

- Input controls can be set to be visible or invisible when you edit a view:
  - Input controls set to **Always prompt** are displayed in the editor and always appear before the report is run.
  - Input controls that aren't set to **Always prompt** are always hidden in the editor and hidden when the report is run.
- Filters defined in the Domain design are always hidden in the editor and when the report is run.
- Filters created in the Data Chooser can be locked or unlocked:
  - Filters that are unlocked display filter information in the editor and are available from the **Options** button when the report is run.
  - Filters that are locked display input controls in the editor when you click 👁 to see the view in display mode but are not available from the **Options** button when the report is run. Users can remove the filter while in the editor, allowing them to see all the data unfiltered when the report is run.
  - You can't change whether the filter is displayed after the report is created.
- Filters defined in the editor are always available in the Filters panel of the editor and from the **Options** button when the report is run.

When setting up input controls for a huge view that takes a long time to run, consider setting the view to **Always prompt**. Before a report is run, the report viewer prompts you to provide the input options**,** preventing the report from running with the default input options.

Filters that are unlocked are available. When input controls or filters don't appear in the report viewer, click the **Options** button to view them. You can learn more about how filters and input controls interact in the editor by walking through the data exploration tutorial with the Filters panel open.

To set an input control to always prompt:

1. Locate a Topic, such as the Parametrized Report Topic, in the repository and click **Edit**.
2. On the Controls & Resources page of the JasperReport wizard, under Input Control Options, select **Always prompt**:

To determine whether an input control is visible:

1. Locate a Topic, such as the Parametrized Report Topic, in the repository and click **Edit**.
2. On the Controls & Resources page, click the name of an input control, such as Country.
3. On the Locate Input Control page, click **Next**.

   At the bottom of the Create Input Control page, if the **Visible** check box is selected, the input control appears on the report when it runs. For more information, see **"Adding Input Controls" on page 176**.

If you don't provide a default value for the input control, users are prompted to select a value when they create a view based on the Topic.

To lock a filter:

1. Click **Create** > **Ad Hoc View**.
2. In the Select Data wizard, click and browse to **Domains** to create a new view based on a Domain.
3. Click **Choose Data**.
4. In **Fields**, move tables and fields from **Source** to **Selected Fields**.
5. Click **Pre-filters**.
6. Double-click a field in the **Fields** panel.
7. In the Filters panel, define a filter as described in **"The Pre-filters Page" on page 155**.
8. Check the **Locked** check box, and click **OK**.
9. Click **Table** to open the Ad Hoc Editor.

   In the Filters panel, the name of the filter and a note about the lock appears under the heading **Locked**.

4.8 Creating a View from a Domain

Like Topics and OLAP connections, administrators and data analysts create Domains for reuse to simplify access to data during view design. Domains give view makers more flexibility than Topics in choosing fields from the database and allow filtering of the data before it is included in a view and the subsequent report.

A view based on a Domain can prompt the user for input that determines what data is presented. For example, if a Domain includes all sales data for a company, the view can present detailed information grouped by postal code and prompt the user to select the geographic area, such as a US state. The view displays only the pertinent information.

For a more complete description of how to create Domains, see **"Example of Creating a Domain" on page 204**.

To begin create a basic view from a Domain:

1. On the Home page, click **Create > Ad Hoc View**. The Select Data wizard opens.
2. Click and navigate to **Domains**. A description of the selected Domain appears at the bottom of the Domains tab.

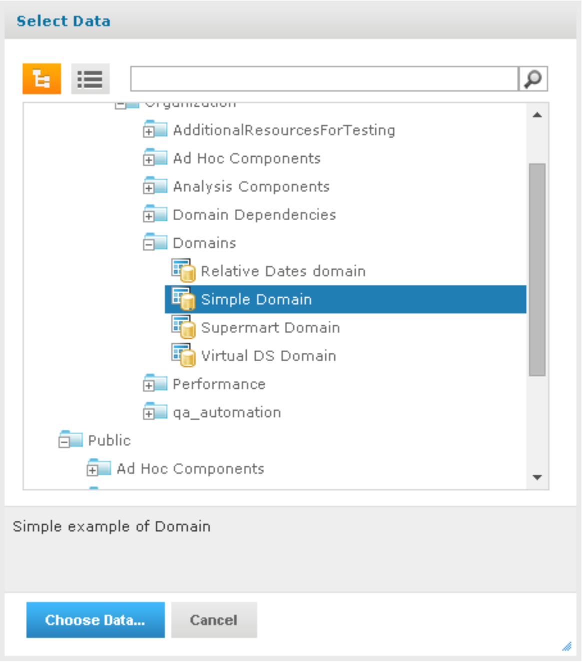


Figure 4-14 Simple Domain Selected in the Select Data Dialog

3. Select the domain you want to use.
4. Click **Choose Data**. The Data Chooser opens to the Fields page.

   You are now ready to configure your data using the Data Chooser Wizard.

4.8.1 Referential Integrity

When you create an Ad Hoc view from a Domain, the two elements are connected - the data in the view is dependent on the Domain. This relationship affects users working with the Domain and the dependent view in a number of ways:

- When an item or items from the Domain are used in the dependent Ad Hoc view, the Domain's administrator is notified, and those items cannot be removed from the Domain.
- Items not used in dependent views can be removed from the Domain by the Domain's administrator; those items no longer appear in the view's Available Fields list.

For more on this topic, see **“Maintaining Referential Integrity” on page 237**

4.8.2 Using the Data Chooser Wizard

To design a Domain Topic or a view based on a domain, use the Data Chooser wizard. To open the Data Chooser wizard:

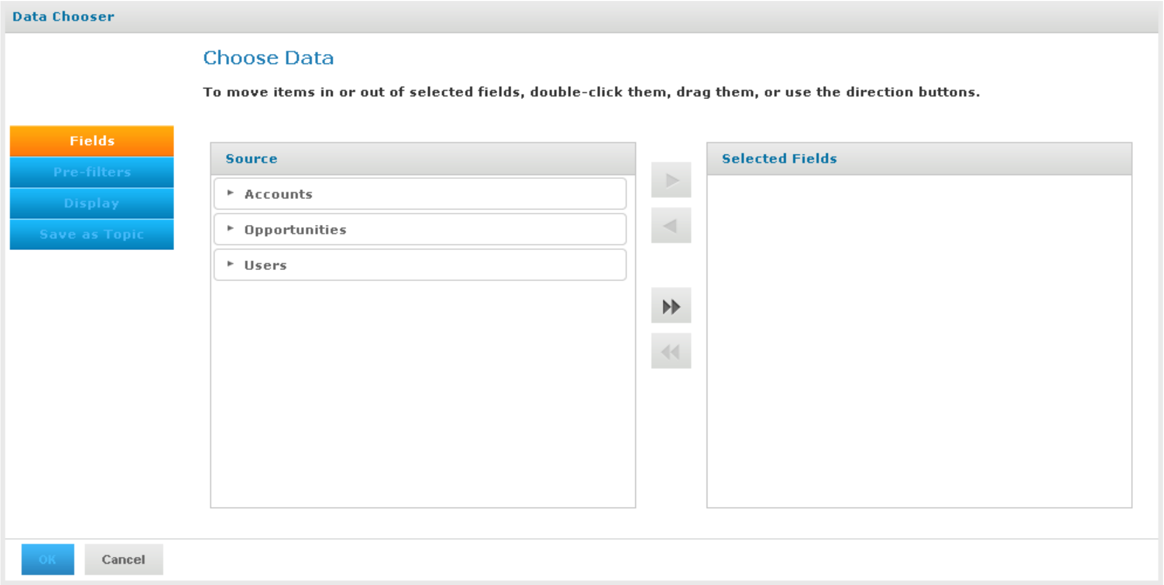
1. Click **Create** > **Ad Hoc View**.
2. Click and browse to a Domain, then click **Choose Data** to access the following pages of the Data Chooser:
   - **The Select Fields Page** – Choose the fields to make available in the Ad Hoc Editor.

- **The Pre-filters Page** – Define a filter on any field, with the option of prompting for user input, or to compare fields.
- **The Display Page** – Change the order and names of fields that appear in the Ad Hoc Editor.
- **The Save as Topic Page** – Save the settings as a Domain Topic.

You must start by selecting some fields on the Select Fields page, but the other three pages are optional and can be completed in any order. Click **Table, Chart,** or **Crosstab** at any time to begin designing a view based on the chosen data.

4.8.2.1 The Select Fields Page

Use this page to choose fields and sets of fields to use in the view or make available in the Domain Topic:



Choose Data

To move items in or out of selected fields, double-click them, drag them, or use the direction buttons.

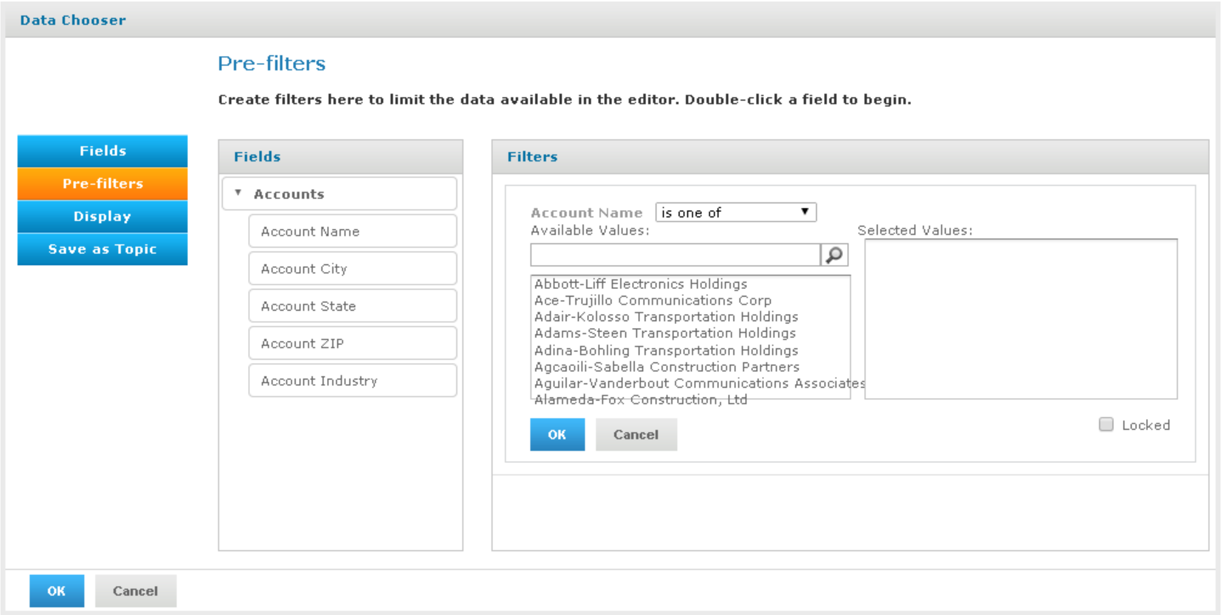
Figure 4-15 The Fields Page of the Data Chooser

- The Source panel displays the sets of fields in the Domain. Use ▼ and ▶ to collapse or expand each set.
- The Selected Fields panel shows the items you selected. You can move a field or set back and forth between the panels by dragging, double-clicking, or selecting the item and clicking an arrow button, such as ▶.
- When you select any field from a set in the Source panel, the set name appears with the field in the Selected Fields panel, If you do not want sets, use the settings on the Display page.
- Some Domains define sets that are not joined, also called data islands. When you select a field from such a set, the behavior on the Select Fields page depends on how the joins were created in the Domain:
  - If the Domain uses basic joins, the unjoined sets aren't available. The Domain Designer only creates basic joins.
  - If the Domain uses advanced joins, all joins are available regardless of the join set of the fields you add. In this case, you must manually make sure that you do not add fields that are in different data islands to a single Ad Hoc view. Otherwise you will receive errors when attempting to work with the view.

4.8.2.2 The Pre-filters Page

You can pre-filter data in the Data Chooser before launching the Ad Hoc Editor or creating a Domain Topic. Pre-filtering data limits the data choices available in a Domain Topic or the fields that ultimately appear in the Ad Hoc view. You can also define a filter on a field that does not appear in the final report. The filter is still applied and only data that satisfies all defined filters appear in the final report. For example, you can filter data to select a single country, in which case it doesn't make sense for the Country field to appear as a row, column, or group. You can also design reports that prompt users to input data to use as a filter.

The Pre-filters page provides powerful functionality for designing views within the server.



Data Chooser

Pre-filters

Create filters here to limit the data available in the editor. Double-click a field to begin.

Account Name is one of

Available Values:

Abbott-Liff Electronics Holdings
Ace-Trujillo Communications Corp
Adair-Kolosso Transportation Holdings
Adams-Steen Transportation Holdings
Adina-Bohling Transportation Holdings
Agcaoili-Sabella Construction Partners
Aguilar-Vanderbout Communications Associates
Alameda-Fox Construction, Ltd

Selected Values:

□ Locked

Figure 4-16 Condition Editor in the Filters Panel on the Pre-filters Page

To define a filter:

1. In the Data Chooser, click **Pre-filters**.
2. Expand the options in the **Fields** panel.
3. Double-click to select a field in the **Fields** panel. Choices appear for filtering the selected field:
4. Choose a comparison operator.

   Text fields have both substring comparison operators such as "starts with" or "contains" and whole string matching such as "equals" or "is one of." When you select a whole string matching operator, a list appears showing all existing values for the chosen field retrieved in real-time from the database.

   In the Filters panel, a drop-down appears containing the account names from which you can select multiple values.
5. Click each value for comparison in Available Values to move it to Selected Values. The account names appear in Selected Values.

If there are more than 50 Available Values, click 🔍 to search for the value. The maximum number of items that can be displayed in Available Values is configurable. For details, see the *JasperReports Server Administrator Guide*.

6. To limit the view design to the four account names in Selected Values, check the **Locked** check box.

   By default, the **Locked** check box is unchecked, making the filter available to end-users running the report.

   In the Report Viewer, users can click the **Options** button to enter a comparison value for this condition; when the user clicks **Apply** or **OK**, the report preview refreshes with data that match the condition. The condition is available as a prompt even if the filtered field does not appear in the report. For example, the final report might present data for a single country, but the country is chosen by the user. Once defined, filter prompts can be modified in the Ad Hoc Editor, as explained in **"Using Input Controls" on page 149**.

   Note that when the **Locked** check box is checked, the filter is not available to end-users running the report. The condition can be removed from the view, if needed, but not edited.

7. Click **OK** to create the filter.

   The **Filters** panel shows the filters you have defined.

8. In the **Filters** panel, click **Change** to modify the condition. Click **OK** to save the changes. After selecting a row, you can also click **Remove** to delete it from the list.

Data rows must match all conditions. In other words, the overall filter applied to the data is the logical AND of all conditions you have defined.

4.8.2.3 The Display Page

Use the Display page to change the default label and order of the fields as they should appear in the list of fields in the Ad Hoc Editor. You can always change the field labels and ordering in the Ad Hoc Editor, but setting them here makes them available in a Domain Topic. The page includes these options:

- To change the order of fields, click once anywhere in a field's row and use the **Move to top**, **Move up**, **Move down**, or **Move to bottom** buttons:

  ⤒, ▲, ▼, and ⤓

  Fields may be moved only within their set, but sets as a whole may be also be moved.
- By default, the field name becomes the display label for the row, column, or measure that you create from it. To change the default display label of a field or set, double-click anywhere in the row and type the new label in the text box.
- Sets and the fields they contain appear in the list of fields in the Ad Hoc Editor. Sets are not used in views, but can be used to add all their fields at once, expediting view creation.
- If you don't want to use sets in the Ad Hoc Editor, select **Flat List** at the top of the Data Source Selection panel. You can now relabel the fields and reorder them.

4.8.2.4 The Save as Topic Page

Here you can enter a name and a description to save the Data Chooser settings as a Domain Topic. Thereafter, you can create different views from the Domain Topic, using its fields, filters, and display label settings. You can also edit the Domain Topic to change the settings.

If an administrator has enabled the Data Staging feature, you can turn data staging on for your domain topic. Data staging was added in version 6.0 of Server. With data staging, the entire dataset for a Domain Topic is indefinitely cached in the server's Ad Hoc cache. Ad Hoc views and reports from that Domain Topic run faster because they fetch data from the local cache, not by querying the production database. To keep data fresh, you can specify a refresh interval, and the server will periodically access the database in the background to reload the entire dataset.

- By default, Domain Topics are saved in the standard Topics folder. This corresponds to the **Ad Hoc Components > Topics** location in the repository; JRXML Topics and Domain Topics in this folder appear on the **Topics** tab when you start a view. Do not modify this folder name.
- The description text appears with the Domain Topic in the repository and at the bottom of the **Topics** tab on the Ad Hoc Source dialog. Enter an informative description that helps users understand the nature and purpose of this Domain Topic.

4.9 Working with Topics

When a user clicks **Create > Ad Hoc View** on the Home page, the Select Data wizard offers a path to a list of Topics populated from the Ad Hoc Components/Topics folder in the repository. There are two types of Topics:

- JRXML-based Topics – Created by administrators using Jaspersoft Studio and uploaded as JRXML files to the proper location in the repository. Topics are typically of this type.
- Domain Topics – Created from a Domain by administrators using JasperReports Server.

Either type of Topic is an empty view associated with a data source in the server, and is then built on in the Ad Hoc Editor.

4.9.1 Uploading a Topic Through the Web UI

JRXML-based topics are the most common type of topic. You can upload previously created JRXML topics via the repository.

To upload a JRXML-based Topic:

1. Log into the server as administrator and select **View > Repository**.

   While any user with sufficient repository permissions can upload a Topic to the server, this example requires an administrator login to access the JServer Jdbc data source.

2. Locate the folder where Topics are stored. The location of the Topics folder depends on your system configuration; by default, Topics are in the **Ad Hoc Components > Topics** folder.
3. Right-click the Topics folder name and select **Add Resource > JasperReport** from the context menu. The Set Up the Report page of the JasperReport wizard appears.

   Add Resource appears on the context menu only if your login account has write privilege to the folder.

4. In the Set Up the Report page, give the Topic a name, a Resource ID, and an optional description, then click **Next**.
   - The **Name** field is the visible name of the file in the repository, such as `Example Topic`.
   - The **Resource ID** field is the internal ID of the object, such as `Example_Topic`. The server does not accept spaces in an internal ID.
   - The **Description** field, such as `Topic uploaded for User Guide example`, helps users understand the purpose of the file.

5. In the **Locate the JRXML File** section, select **Upload a Local File**, and click **Browse** to locate the file and upload the Topic from the file system. In this example, the file is <js-install>/samples/adhoc/topics/adhoc_sample.jrxml.

   To locate adhoc_sample.jrxml, you need access to the server host file system.

6. Click **Data Source**.

   Because this is a simple Topic without parameters, there are no controls and resources associated with it. If the Topic has a Parametrized query, you can create input controls for it. See **“Selecting a Data Source for Running the Complex Report” on page 186**. Such input controls can appear in the Ad Hoc Editor and when the report is run.

7. Click **Select data source from repository** and **Browse** to locate the data source named **Data Sources/JServer Jdbc data source**.
8. Select Data Source, then click **Select**.

   Topics must be associated with the data source that they were designed for.
9. Click **Submit** at the bottom of the screen.

   Topics usually do not need a query or customization, but you can define them.

When you select **Create > Ad Hoc View** and click in the Select Data wizard, you can browse to **Ad Hoc Components > Topics**. If you select the Example Topic, you can create a report using the columns available in the data source selected in **step 7**.

When you create a JRXML file for use as a Topic, you can specify the name to display for each field that the Topic returns. To do so, define a field property named `adhoc.display` for each field declared in the JRXML. The `adhoc.display` field property must have the following syntax:

```
<property name=”adhoc.display” value=”Any Name”/>
```

If the Topic includes fields with unusual datatypes, those fields don't appear in the Ad Hoc Editor because it's not equipped to manage them properly. For example, if the Topic is defined against a MongoDB instance that returns data of type array, this field isn't available in the Ad Hoc Editor. For more information on datatype support in the Ad Hoc editor, see the *JasperReports Server Administrator Guide*.

For example, this JRXML code declares a StoreState field that is displayed in reports as Store State:

```
<field name=”StoreState” class=”java.lang.String”>

 <property name=”adhoc.display” value=”Store State”/>
</field>
```

Topics also support the $R expressions for field names; for more information, see **“Localizing Reports” on page 190**.

For fields in a non-domain topic the following properties may be of interest:

- `dimensionOrMeasure`: marks a field as a field or a measure
- `defaultAgg`: which aggregation should be used for this measure (avg, etc)
- `semantic.item.desc`: A description for the field
- `defaultMask`: set a measure as a $, date etc

For more information on working with JRXML topics, see the *Jaspersoft Studio User Guide*.

4.9.2 Creating Topics from Domains

In some circumstances it is important to create a Topic based on the Domain and data settings you chose, but it's not always necessary. The main consideration is how reports based on the Domain-based view are used. The following table explains the choices.

If you want to ...	Then do this	Explanation
Create a single-use report from the Domain with your settings.	Click **Table**, **Chart**, or **Crosstab** after defining your settings in the Data Chooser, then format the view and create your report.	Your field selections appear in the Ad Hoc Editor and you can create views from them, but the settings themselves are not saved.
Run a report repeatedly as is or with prompting for new input, or make very similar reports in the Ad Hoc Editor.	Click **Table**, **Chart**, or **Crosstab** after defining your settings in the Data Chooser, then format and save the view in the Ad Hoc Editor.	After you save a view, users can create reports to be run from it repeatedly and be prompted for input each time if you defined unlocked filters.
Reuse your field, filter, and display settings on this Domain to create new views with different formatting.	After saving your settings as a Domain Topic in the Data Chooser, start a new view and choose your Domain Topic from the **Domains** tab.	Domain Topics are saved in JRXML format in the repository. They appear on the **Topics** tab when you start a view.
Be able to modify one or more of the settings you made in the Data Chooser wizard.	After saving your settings as a Domain Topic in the Data Chooser, find your Domain Topic in the repository and open it in the Domain Designer.	Domain Topics can be edited as described in **"Editing a Domain Topic " on page 160**. After you save your changes, you can create views based on the modified Domain Topic.
Create views from the same repository data source but with different Domain settings, such as derived fields and different joins to make new fields available in a report.	Edit the Domain and save it with a new name. Then start a new view and choose your data from the new Domain.	Domains provide advanced functionality such as table joins and derived fields based directly on a data source. Domains usually require administrator permissions to create and modify.

4.9.2.1 Access Permissions in Domain Topics

If other users create reports from your Domain Topic-based view, and the Domain is configured for security, it's important to consider everyone's access permissions. You might not have access to all of the fields in the Domain nor to all the data in those fields. There may be fields that can be seen only by other users, and in the fields that you can see, some data may be hidden from you. When you save the Domain as a Topic, only the fields that you selected appear in the Domain Topic. When you create a view from the Topic, only the data that you can access in those fields appears. When others create views from the Topic, they see only fields that they have permission to access. These rules also apply to the reports generated from these views.

For example, in a Domain, user Tomas can access fields B-C and data rows 1-3; Anita can access fields C-E and data rows 2-5. Tomas uses the Data Chooser and saves a Domain Topic based on the Domain. When Tomas and Anita create reports from the same view, they see different combinations of fields and the data in them.

	Fields	Data
Tomas's report from his Domain Topic	B C	1 2 3
Anita’s report from the same Domain Topic	C	2 3 4 5

Even though Anita has permission to see more fields, they're not available to her because Tomas did not have access to them when he created the Domain Topic. However, Anita does have permission to see more data than Tomas, so when she creates a view, or opens or runs the report based on that view, she can see more rows than Tomas can when he views the report. See **“Editing a Domain” on page 235** for a technical explanation of data security for Domains. See the note in **“Editing a Domain” on page 235** about the impact of editing a Domain or Domain Topic.

4.9.2.2 Saving Domain Settings as a Domain Topic

To save settings in the Data Chooser wizard as a Domain Topic:

1. While making selections in the Data Chooser, navigate to the **Save Topic** page.
2. Enter a Topic name and description.

   Do not change the location folder. Using the default /adhoc/topics folder makes the saved Domain Topic available in the **Ad Hoc Components > Topics** folder when you select **Create > Ad Hoc View**.
3. If your data selections, filter definitions, and display settings are complete, click **Table**, **Chart**, or **Crosstab**.

   If settings are incomplete, navigate to the other pages to finish, then click **Table**, **Chart**, or **Crosstab**

   The new Topic appears in the **Ad Hoc Components > Topics** folder.

   Because a Domain Topic is a type of report, it appears when the Search page is filtered to show reports:

4.9.2.3 Editing a Domain Topic

You can modify a Domain Topic you created using the Data Chooser .

To edit the settings in a Domain Topic:

1. Select **View > Repository** and search (or browse) for the Domain Topic you want to modify. Domain Topics are usually kept in the **Ad Hoc Components > Topics** folder.
2. Right-click the Domain Topic and select **Open in Designer** from the context menu. The Domain Topic opens in the Data Chooser wizard.
3. Follow the guidelines in **“Using the Data Chooser Wizard” on page 153** to edit the Domain Topic as needed.
4. To save changes to the selected Domain Topic, click **Table**, **Chart, or Crosstab** on any page.

Use caution when editing Domain Topics that may have been used to create other views. Users relying on the Domain Topic might receive unexpected data or errors. It's safer to save changes as a new Domain Topic.

To save changes as a new Domain Topic, navigate to the Save as Topic page and enter identifying information for the new Topic, then click **Table**, **Chart**, **or Crosstab**.

Chapter 5 Adding Reports Directly to the Repository

This section describes functionality that can be restricted by the software license for JasperReports Server. If you don’t see some of the options described in this section, your license may prohibit you from using them. To find out what you're licensed to use, or to upgrade your license, contact Jaspersoft.

Using the Ad Hoc Editor, you can create reports within JasperReports Server from pre-defined Topics and Domains. You can also create reports outside of JasperReports Server and add them to the repository. To add a report to the repository, you need a valid JRXML file. To create and validate this file, you can use Jaspersoft Studio. Jaspersoft recommends Jaspersoft Studio for most users because its graphical user interface simplifies the job. If you have a thorough understanding of the file structure, you can use a text editor to create the file containing JRXML code.

You can add a report to the server’s repository in two ways:

- From within the server
  Add the JRXML file and any other resources the report needs as a report unit. A wizard guides you through each step.
- From Jaspersoft Studio
  Design the report in Jaspersoft Studio, and use a connection to JasperReports Server to add the JRXML and resources to the JasperReports Server repository.

This chapter includes examples of adding a report to the repository using the server’s wizard and the plug-in. The chapter contains the following sections:

- **Overview of a Report Unit**
- **Adding a Simple Report Unit to the Server**
- **Adding a Complex Report Unit to the Server**
- **Adding Cascading Input Controls to a Report**
- **Editing JRXML Report Units**
- **Localizing Reports**

5.1 Overview of a Report Unit

In the server, a report unit is the collection of elements used for retrieving data and formatting output. **Figure 5-1 on page 164** shows these elements:

- The data source and the query that retrieves data for the report.
- The main JRXML that determines the layout and is the core of the report unit.

- The main JRXML defines other elements in one of the following ways:
  - Creating definitions internally
  - Referring to existing elements in the repository using the `repo:` syntax
- The input controls and other resources.

For more information about the report unit, refer to the *JasperReports Server Ultimate Guide*.

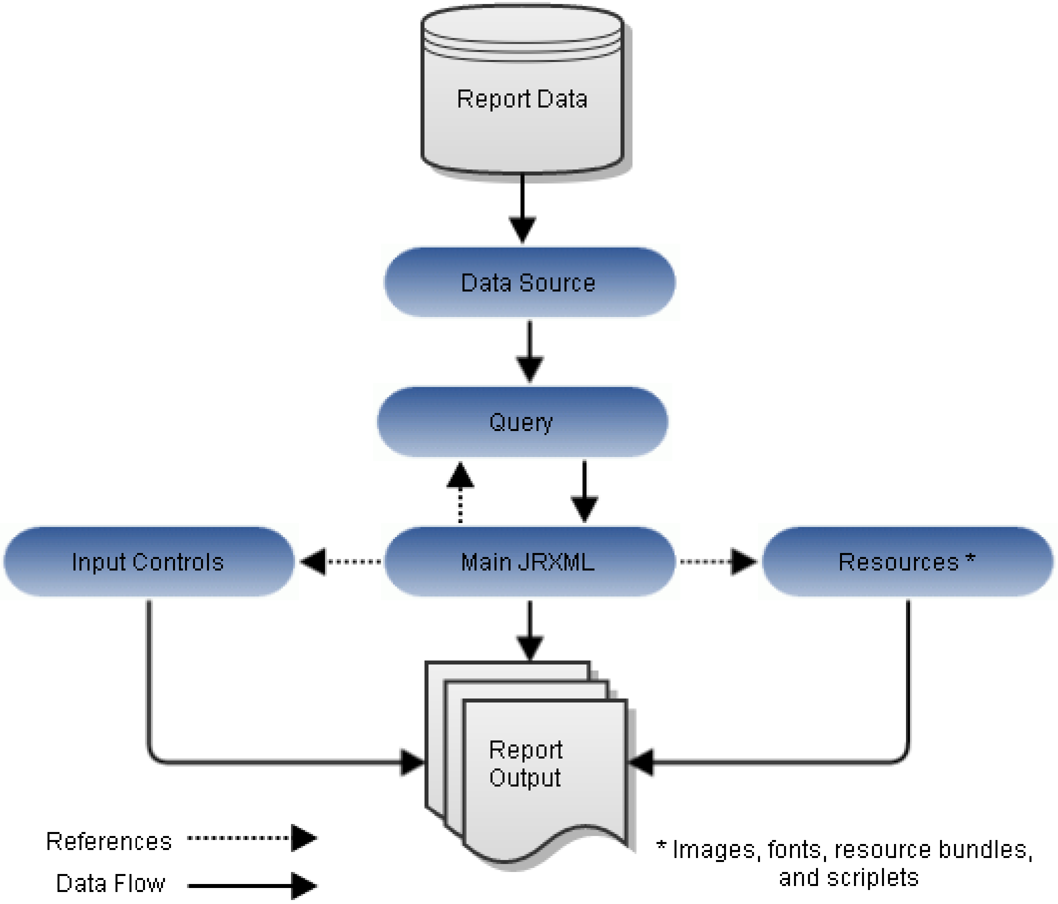


Figure 5-1 Anatomy of a Report Unit

5.2 Adding a Simple Report Unit to the Server

This section presents an example of uploading a JRXML to the server. The report only has two image resources. The example defines a custom query for the report unit.

To add the simple report unit to the server, you need access to the following sample data installed with the server:

- <js-install>/samples/reports/AllAccounts.jrxml

- <js-install>/samples/images/logo.jpg files

5.2.1 Uploading the Main JRXML

If you want to validate the JRXML before uploading it, use Jaspersoft Studio. The server doesn't validate the JRXML when you upload it.

This procedure shows you how to set up a name for the report in the repository and upload the main JRXML file that references all other elements.

To upload the main JRXML for the simple report example:

1. Log into the server as administrator and select **View > Repository**.

   If you log in as a user, you can upload a report unit to the server, but this example requires an administrator login to access the image resources.

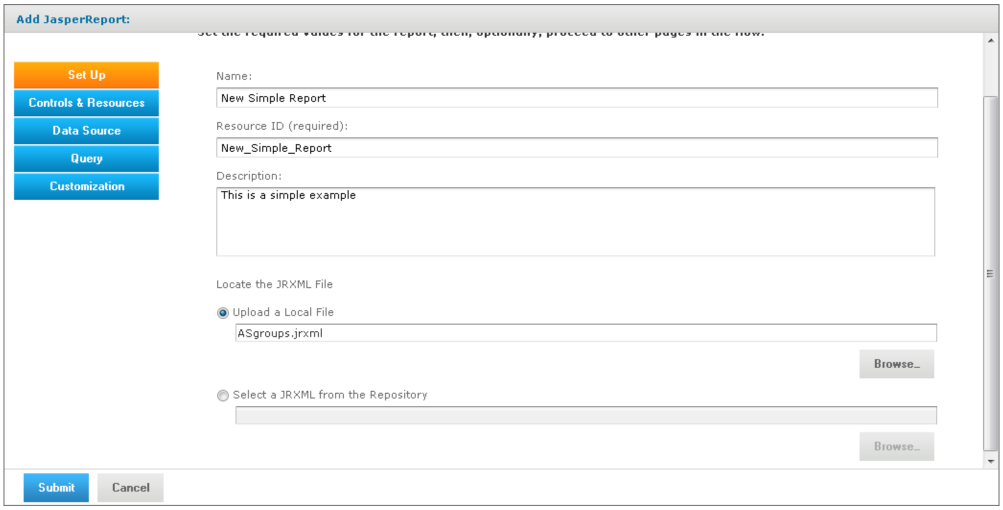
2. Locate the folder where you want to add the report. For example, go to **Organization > Reports**.
3. Right-click the Reports folder and select **Add Resource > JasperReport** from the context menu. The Set Up the Report page of the JasperReport wizard appears.

   Add Resource appears on the context menu only if you have write permission to the folder.

4. In **Naming**, enter the name and description of the new report and accept the generated Resource ID:
   - Name – Display name of the report: `New Simple Report`
   - Resource ID – Permanent designation of the report object in the repository: `New_Simple_Report`
   - Description – Optional description displayed in the repository: `This is a simple example`
5. Select **Upload a Local File** and **Browse** to <js-install>samples/reports/AllAccounts.jrxml.

   You can upload a new JRXML or select a JRXML from the repository.

   In **Figure 5-2** you can see the Set Up the Report page.



Set the required values for the report, then, optionally, proceed to other pages in the flow.

Name: New Simple Report

Resource ID (required): New_Simple_Report

Description: This is a simple example

Locate the JRXML File

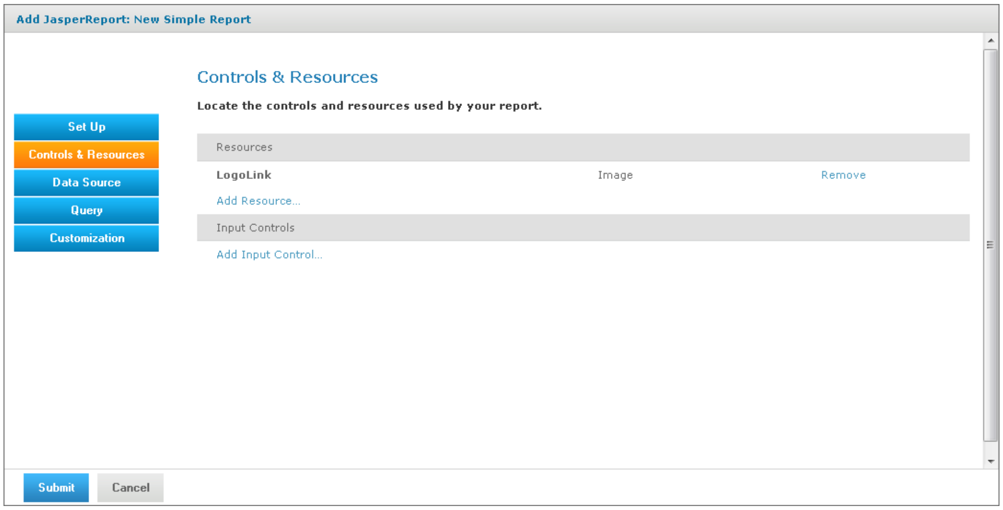
● Upload a Local File

ASgroups.jrxml

○ Select a JRXML from the Repository

Figure 5-2 Required Set Up Values

6. Click **Controls & Resources**. The server uploads the main JRXML file. The Controls & Resources page reappears. If the server needs additional resources to validate the report, you'll be prompted to locate those resources.



Controls & Resources

Locate the controls and resources used by your report.

Resources

LogoLink	Image	Remove

Input Controls

Figure 5-3 Suggested Resources in the Resources List

A JRXML file doesn't embed resources, such as images. When the server uploads the JRXML, it tries to detect any missing resources. For example, **Figure 5-3, "Suggested Resources in the Resources List," on page 166** shows that two image files are missing:

- Logolink
- AllAccounts_Res2

5.2.2 Uploading Suggested File Resources

If needed files are missing, as shown in **Figure 5-3**. you need to take one of the following actions:

- Upload resources that the report needs
- Select a resource from the repository

If the Controls & Resources page doesn't suggest resources, perhaps the report doesn't reference any. However, the server can't always detect all the referenced resources, as discussed in **"Uploading Undetected File Resources" on page 174**.

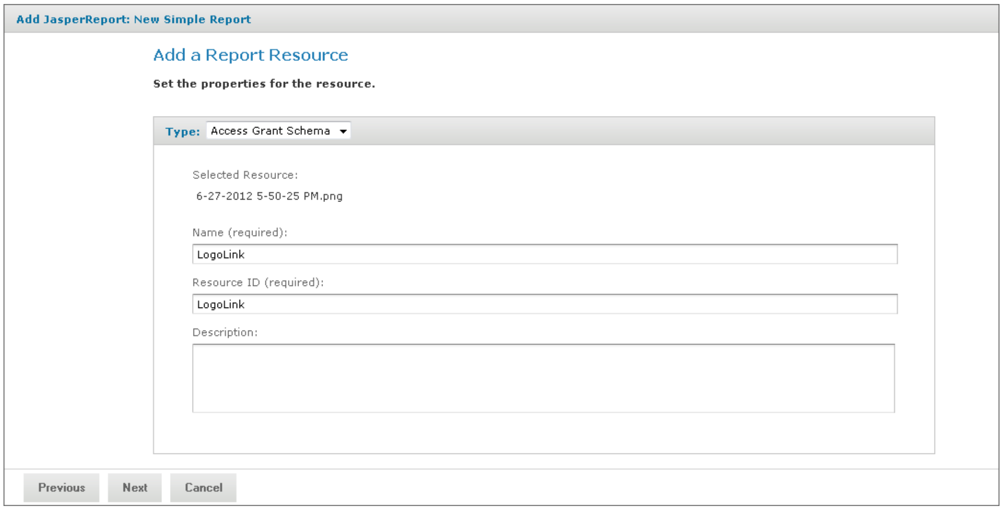
To upload suggested file resources for the simple report example:

1. Click **Add Now** in the row of the missing resource. For example, LogoLink. The Locate File Resource page appears.
2. Choose **Select a resource from the Repository** and **Browse** to the needed file (Images/JR Logo).

Any image file works.

3. Click **Select**.

   The path to the image appears in the wizard.
4. Click **Next**. The Add a Report Resource page appears.



Add JasperReport: New Simple Report

Add a Report Resource

Set the properties for the resource.

Type: Access Grant Schema

Selected Resource:
6-27-2012 5-50-25 PM.png

Name (required): LogoLink

Resource ID (required): LogoLink

Description:

Figure 5-4 Properties of a Resource

The properties include the LogoLink name, resource ID, and description. These properties don't redefine the properties of the JRLogo file in the repository.

5. Click **Next** to accept the default naming of the file resource.

   The Controls & Resources page appears again, showing that the LogoLink resource was added.
6. Click **Add Now** in the AllAccounts_Res2 row. The Locate File Resource page appears again.
7. Select **Upload a Local File** and **Browse** to <js-install>/samples/images/logo.jpg.
8. Open logo.jpg, and click **Next**. The server uploads the file.

   The Add a Report Resource page appears, showing the properties of the file resource.

The properties include the AllAccounts_Res2 name, resource ID, and description.

9. Click **Next** to accept the default naming of the file resource.

   The Controls & Resources page reappears, showing the addition of both resources referenced in the main JRXML.
10. Click **Data Source** and define a data source as described in the next section.

5.2.2.1 Referencing Styles

You can apply a style to your report by referencing a property sheet, created in Jaspersoft Studio, that contains information on a style.

To do this, you will need to edit the Styles.jrxml file and change the <templates> statement.

To reference styles from an external property sheet:

1. Locate and open the Styles.jrxml file.
2. In the JRXML file, declare `<template>"repo:styles"</template>` (do not include the .jrtx extension).
3. Save the JRXML file.
4. Log into the sever as an administrator, and select **View > Repository**.
5. Expand the **Organization** folder.
6. Right-click the Reports folder and select **Add Resource > JasperReport** to open the Add JasperReport wizard.
7. On the Set Up page, enter the required information.
8. Select **Upload a Local File** and click **Browse...**
9. Locate `Styles.jrxml`, and click **Open**.
10. Click **Submit**.
11. Copy the base_styles.jrtx file from the directory:

    `<jrs-install>\samples\templates\reports`

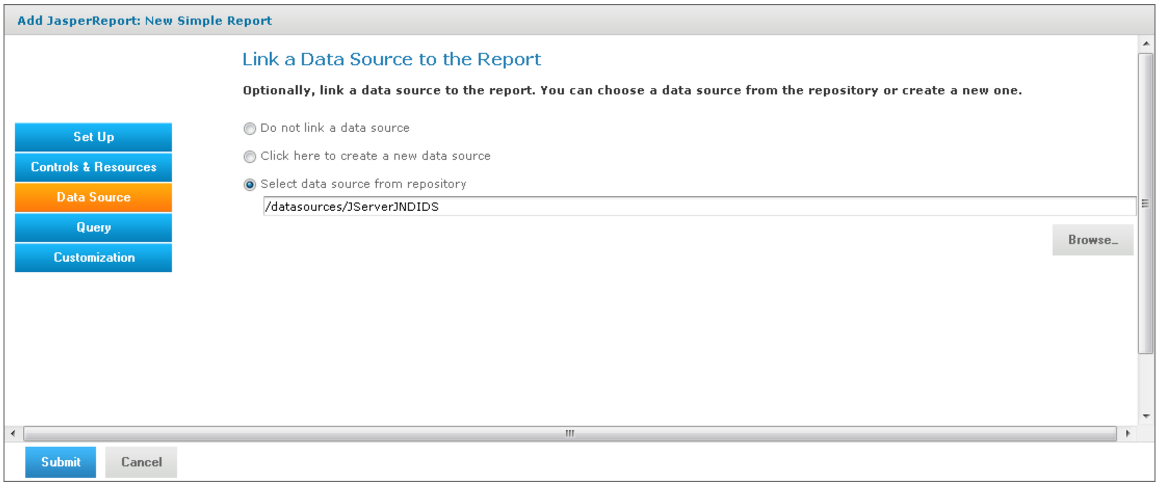
    and place into the `C:\` directory.
12. Edit the `styles.jrtx` file and add `C:\\ before base_styles.jrtx`.
13. Back on the server, select **View > Repository,** then expand the **Organization** folder**.**
14. Right-click **Images**, and select **Add Resource > File > Style Template.**
15. Under Path to File, click **Browse...** then locate the style.jrtx file.
16. Enter a Name (Styles) and click **Submit**.
17. In the repository, open the Reports > Styles folder, then right click on Styles report and select **Edit.**
18. Click **Control & Resources**, then click the **Add Resource...** link.
19. Select **Select a resource from the Repository** radio button, **Browse...**
20. Locate **Organization > Images > Styles**, and click **Select**.
21. Click **Next** button
22. On the Add a Report Resource screen, enter **Name: Styles**, then click **Next**.
23. Click **Submit**

5.2.3 Defining the Data Source

Data sources belong to the report engine, JasperReports Server, and are not defined in the main JRXML. The JRXML does not retain data sources defined in Jaspersoft Studio when you add the JRXML to the server. You need to define a data source for your report unit in the server. On the Data Source page, select a data source in the repository or create a new data source on-the-fly. The data source can be different from the database configured for JasperReports Server if the application server can find its driver. For example, in a default installation of JasperReports Server, Tomcat looks for data source drivers in <js-install>/apache-tomcat/lib. Put a copy of the driver in this location.

To define the data source for the simple report example:

1. In the JasperReport wizard, click **Data Source**. The Link a Data Source to the Report page presents these choices:
    - Do not link a data source – Select or define the data source at a later time. You see an error if you run the report in this state.
    - Click here to create a new data source – Define a new data source available only to your report.
    - Select data source from repository – Select an existing data source from the repository.
2. Choose **Select data source from the Repository** and **Browse** to /Data Sources/JServer JNDI Data Source. Click **Select**. The path to the data source appears.



Add JasperReport: New Simple Report

Link a Data Source to the Report

Optionally, link a data source to the report. You can choose a data source from the repository or create a new one.

○ Do not link a data source

○ Click here to create a new data source

● Select data source from repository

/datasources/JServerJNDIDS

Figure 5-5 Data Source Page

3. Click **Query** to define the Query as described in the next section.

5.2.4 Defining the Query

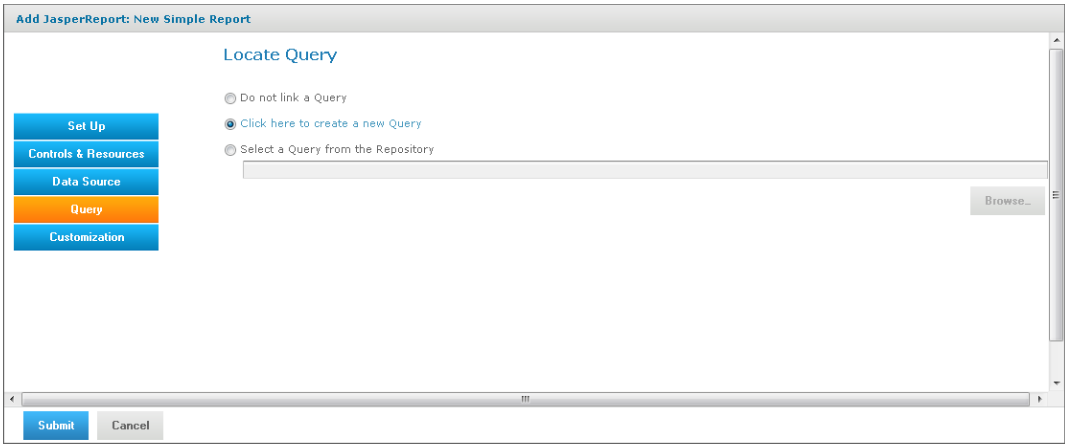
The query in the report unit determines the data that the server retrieves from a data source. You can use an existing query or define a new one. You can create multiple reports that look the same but contain different data by defining different queries for the same JRXML file. The simple report example uses a custom query that returns accounts from a single country.

To define a custom query for the simple report example:

1. In the JasperReport wizard, click Query. The Locate Query page presents these choices:
    - Do not link a Query – Select this option to use a query already defined within the main JRXML.

- Click here to create a new Query – Guides you through defining a new query for this report only.
- Select a Query from the Repository – Select a query from the repository. The AllAccounts.jrxml file uploaded in **“Uploading the Main JRXML” on page 165** already contains a query. This example overrides the existing query by defining a new one.

2. Select the **Click here to create a new Query** option.



Add JasperReport: New Simple Report

Locate Query

○ Do not link a Query

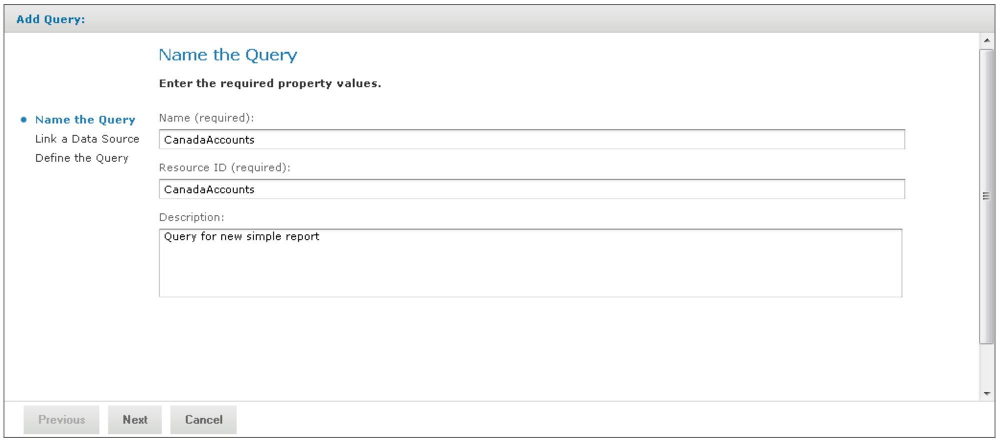
● Click here to create a new Query

○ Select a Query from the Repository

Figure 5-6 Query Page

3. Click the link, **Click here to create a new Query**. The Name the Query page appears.
4. Enter the name, resource ID, and description of the query. In this example, the query must retrieve only Canadian accounts. Enter the following values:
   - Name – `CanadaAccounts`
   - Resource ID – `CanadaAccounts`
   - Description – `Query for New Simple Report in User Guide`

   This query and its properties are visible only within the report unit.



Add Query:

Name the Query

Enter the required property values.

Name (required): CanadaAccounts

Resource ID (required): CanadaAccounts

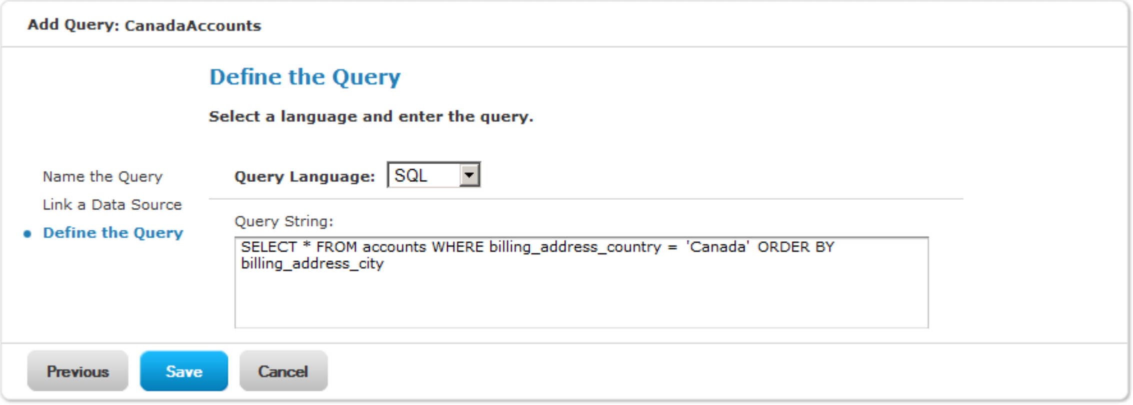
Description: Query for new simple report

Figure 5-7 Name the Query Page

5. Click **Next**. The Link a Data Source to the Query page appears. You can select a query from the repository, define a new one, or select not to link a data source.

6. Select **Do not link a data source** to use the data source you selected in **"Defining the Data Source" on page 169**.
7. Click **Next**. The Define the Query page appears.
8. Select **SQL** in the Query Language drop-down and enter the following Query String to retrieve only Canadian accounts:

```
SELECT * FROM accounts WHERE billing_address_country = 'Canada' ORDER BY billing_
address_city
```



Add Query: CanadaAccounts

Define the Query

Select a language and enter the query.

Query Language: SQL

Query String:

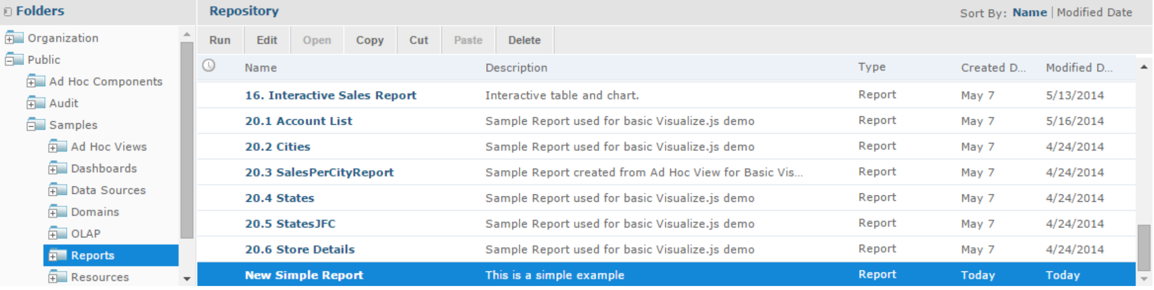
```
SELECT * FROM accounts WHERE billing_address_country = 'Canada' ORDER BY
billing_address_city
```

Figure 5-8 Definition of a Query

9. Click **Save** to save the query. The Customization page appears. No customization is required for the example.
10. Click **Submit** to submit the new report unit to the repository.

5.2.5 Saving the New Report Unit

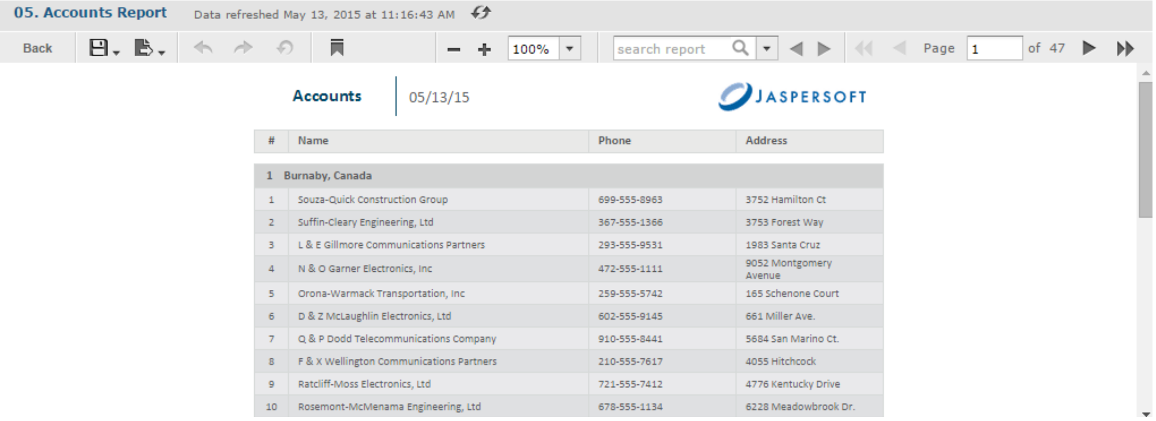
To submit a new report unit to the repository, click **Submit** on any page of the JasperReport wizard, from the Set Up page to the Customization page. You don't have to set options you don't need, such as Customization. When you click **Submit**, the server validates the report unit. If validation is successful, a message appears at the top of the repository page, indicating that the report was saved. The report appears in the repository with the description you entered on the Set Up page. To run the report and view the output, click its name, New Simple Report.



Name	Description	Type	Created D...	Modified D...
16. Interactive Sales Report	Interactive table and chart.	Report	May 7	5/13/2014
20.1 Account List	Sample Report used for basic Visualize.js demo	Report	May 7	5/16/2014
20.2 Cities	Sample Report used for basic Visualize.js demo	Report	May 7	4/24/2014
20.3 SalesPerCityReport	Sample Report created from Ad Hoc View for Basic Vis...	Report	May 7	4/24/2014
20.4 States	Sample Report used for basic Visualize.js demo	Report	May 7	4/24/2014
20.5 StatesJFC	Sample Report used for basic Visualize.js demo	Report	May 7	4/24/2014
20.6 Store Details	Sample Report used for basic Visualize.js demo	Report	May 7	4/24/2014
New Simple Report	This is a simple example	Report	Today	Today

Figure 5-9 New Simple Report Added to the Repository

The following figure shows the output, only Canadian accounts.



05. Accounts Report Data refreshed May 13, 2015 at 11:16:43 AM

Accounts 05/13/15

#	Name	Phone	Address
1	Burnaby, Canada		
1	Souza-Quick Construction Group	699-555-8963	3752 Hamilton Ct
2	Suffin-Cleary Engineering, Ltd	367-555-1366	3753 Forest Way
3	L & E Gillmore Communications Partners	293-555-9531	1983 Santa Cruz
4	N & O Garner Electronics, Inc	472-555-1111	9052 Montgomery Avenue
5	Orona-Warmack Transportation, Inc	259-555-5742	165 Schenone Court
6	D & Z McLaughlin Electronics, Ltd	602-555-9145	661 Miller Ave.
7	Q & P Dodd Telecommunications Company	910-555-8441	5684 San Marino Ct.
8	F & X Wellington Communications Partners	210-555-7617	4055 Hitchcock
9	Ratcliff-Moss Electronics, Ltd	721-555-7412	4776 Kentucky Drive
10	Rosemont-McMenama Engineering, Ltd	678-555-1134	6228 Meadowbrook Dr.

Figure 5-10 Output of the New Simple Report

In the report viewer, click to go to the end of the report. The logo images that you added as file resources appear.

5.3 Adding a Complex Report Unit to the Server

This section includes an example of how to add a report unit with all these resources:

- SalesByMonth.jrxml – the main JRXML file
- SalesByMonthDetail.jrxml – a subreport
- sales.properties – an English resource bundle file
- scriptlet.jar – a scriptlet class JAR file
- JR Logo – an image in the repository
- JServer JNDI data source – a data source file in the repository

These resources are part of the sample data installed with the server. To complete this example and run the report without server errors, you need access to these resources.

The example also guides you through defining every type of input control:

- Text
- Check box
- Drop-down
- Date
- Query

If you're not interested in creating all types of input controls, but want to work through part of the example, delete parameters for the input controls you don't create before you run the report.

The complex report you create in this example is almost exactly like the SalesByMonth report in the **Reports** folder of the repository.

To upload the main JRXML and suggested resource files for the complex report unit:

1. Log into JasperReports Server as administrator and select **View > Repository**.

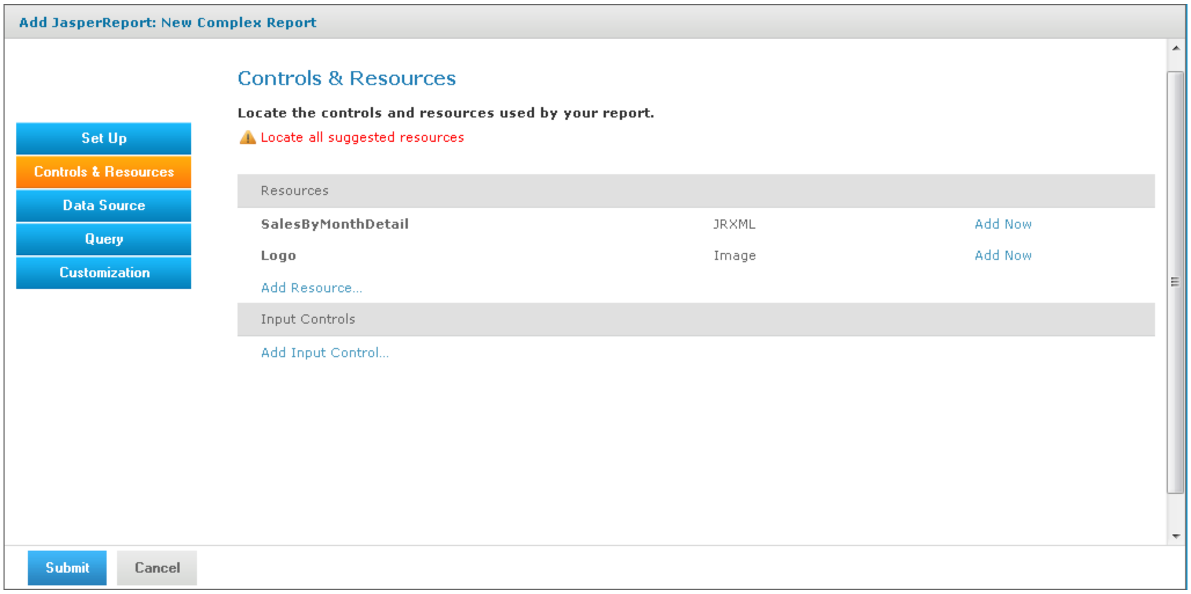
If you log in as a user, you can upload a report unit to the server, but this example requires an administrator login to access the image resources.

2. Navigate to the folder containing your report. For example, navigate to **Organization > Reports**.
3. Right-click the Reports folder and select **Add Resource > JasperReport** from the context menu. The Set Up the Report page appears.

Add Resource appears on the menu only if you have write privilege to the folder.

4. Enter these properties:
   - Name – `New Complex Report`
   - Resource ID – `New_Complex_Report`
   - Description – `This is a complex report`
5. Select **Upload a Local File**.
6. Click **Browse** to locate the file <js-install>/samples/reports/SalesByMonth.jrxml.
7. Click **Controls & Resources**.

   On the Controls & Resources page, **Figure 5-11**, suggests resources to be uploaded for the report:
   - A sub-report (the SalesByMonthDetail.jrxml file)
   - A logo image



Add JasperReport: New Complex Report

Controls & Resources

Locate the controls and resources used by your report.

⚠ Locate all suggested resources

Resources		
SalesByMonthDetail	JRXML	Add Now
Logo	Image	Add Now

Add Resource...

Input Controls

Add Input Control...

Figure 5-11 Suggested Resources for the Complex Report

8. On the Controls & Resources page, upload the sub-report:

   a. Click **Add Now** in the SalesByMonthDetail row. The Locate File Resource page appears.

   b. Select **Upload a Local File**.

   c. Click **Browse** and locate the file <js-install>/samples/reports/SalesByMonthDetail.jrxml. Select SalesByMonthDetail.jrxml.

   The path to SalesByMonthDetail.jrxml appears in the **Upload a Local File** field.

   d. On the Locate File Resource page, click **Next**.
   e. On the Add a Report Resource page, click **Next** to accept the default report resource name and resource ID.
9. On the Controls & Resources page, upload the logo image resource:
   a. Click **Add Now** in the Logo row. The Locate File Resource page appears.
   b. On the Locate File Resource page, click **Select a resource from the Repository**.
   c. Click **Browse** to locate the file /Images/JR Logo and select JR Logo.
   d. Click **Next**. The Add a Report Resource page appears.
   e. On the Add a Report Resource page, click **Next** to accept the default name, resource ID, and description.

5.3.1 Uploading Undetected File Resources

The JasperReport wizard can't detect every type of resource referenced in the main JRXML. You need to add the undetected resources; so the server can validate the report. For the following example we provide the names of these resources. To discover undetected resources open the JRXML in Jaspersoft Studio and examine its parameters and properties. For more information about the JasperReports Server Plug-in, see the *Jaspersoft Studio User Guide*.

These are the undetected resources in the SalesByMonth.jrxml:

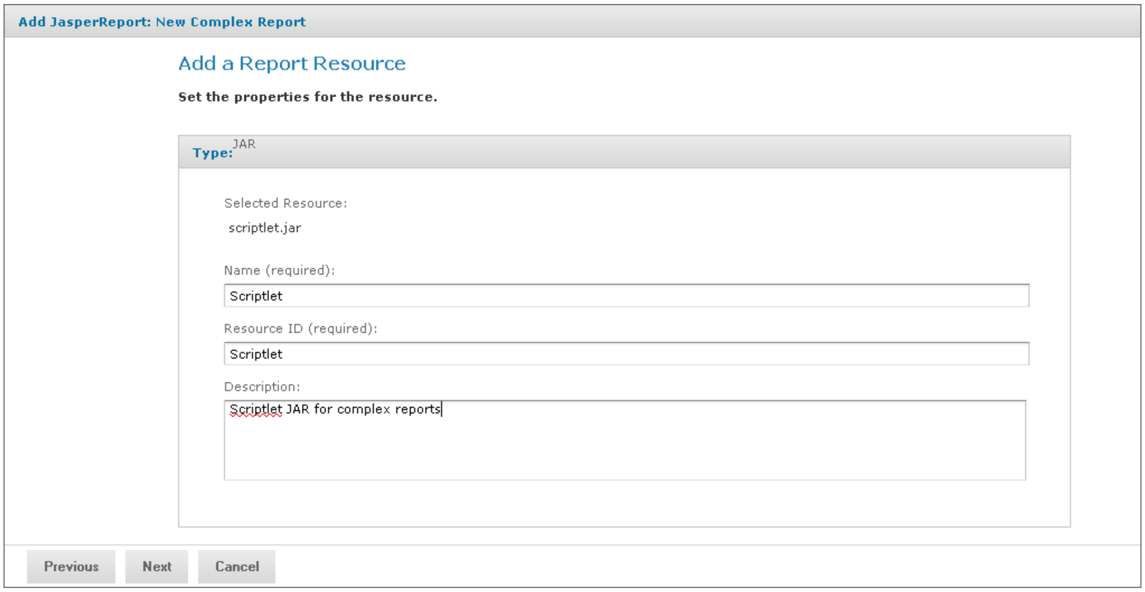
- A scriptlet JAR – The scriptlet writes the message, "I'm a scriptlet in a jar," to the last page of the report output.
- An English language resource bundle.
- The optional Romanian language resource bundle.
  If you're interested in working with a multi-lingual report, add the Romanian resource bundle. The Romanian resource bundle is part of the sample data installed with the server.

On the Controls & Resources page, upload the undetected resources to the server using exactly the same name Jaspersoft Studio uses for the resource ID.

To upload the undetected file resources for the complex report example:

1. Add and upload the scriptlet JAR file:
   a. On the Controls & Resources page, click **Add Resource**.
   b. On the Locate File Resource page, select **Upload a Local File** and click **Browse** to locate the <js-install>/samples/jars/scriptlet.jar file. Select scriptlet.jar.
      The path to the file appears in the **Upload a Local file** field.
   c. Click **Next**.
      The Add a Report Resource page appears. **Figure 5-12 on page 175** shows the file name scriptlet.jar, indicating that the server successfully loaded and automatically detected the JAR.
   d. Enter the following information:
      - Name – `Scriptlet`
      - Resource ID– `Scriptlet`. The Resource ID is referenced in the main JRXML file, so do not change it.
      - Description – `Scriptlet JAR for complex report`

      Figure 5-12 on page 175 shows these values entered on the Add a Report Resource page.



Set the properties for the resource.

Type: JAR

Selected Resource:
scriptlet.jar

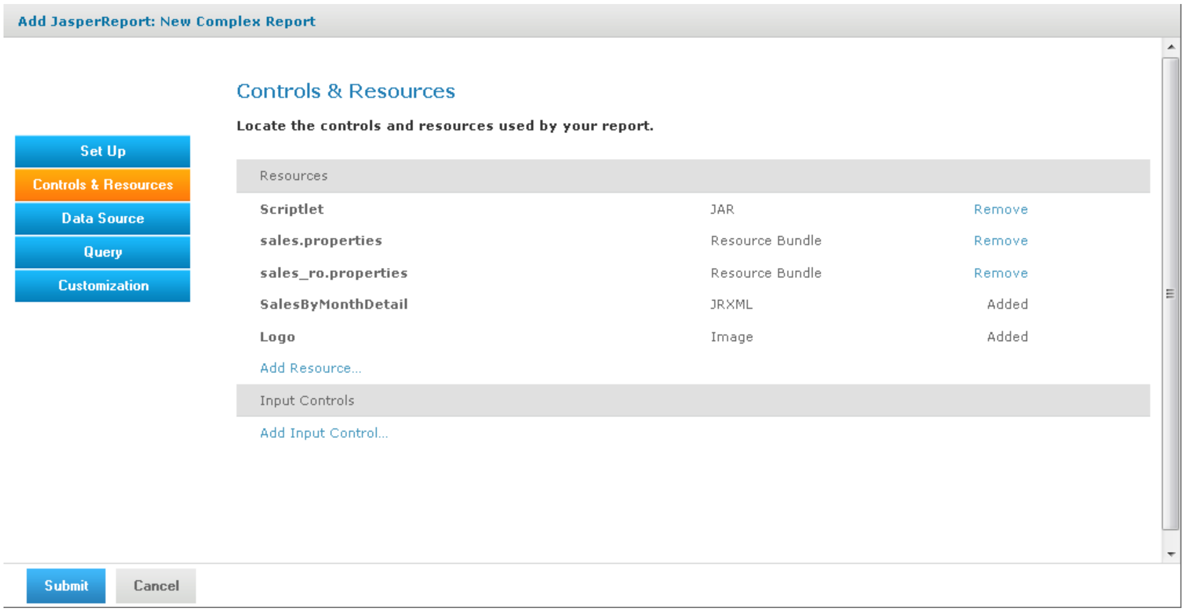
Name (required):
Scriptlet

Resource ID (required):
Scriptlet

Description:
Scriptlet JAR for complex reports

Figure 5-12 Scriptlet JAR Resource Properties

2. Click **Next**.
3. Add and upload the English resource bundle:
   a. On the **Controls & Resources** page, click **Add Resource**. The Locate File Resource page appears.
   b. Select **Upload a Local File**, click **Browse** to locate the file <js-install>/samples/resource_bundles/sales.properties, and select it. The path to the resource bundle appears in the **Upload a Local file** field.
   c. In Locate File Resource, click **Next**. The **Add a Report Resource** page indicates that the file was successfully loaded and automatically detected as a resource bundle.
   d. Enter the following information:
      - Name – `sales.properties`
      - Resource ID – `sales.properties`
      - Description – `Default English resource bundle`
4. Click **Next**.
5. Add and upload the Romanian Resource bundle:
   a. On the Controls & Resources page, click **Add Resource**.
   b. Select **Upload a Local File**, click **Browse** to locate the file <js-install>/samples/resource_bundles/sales_ro.properties, and select it.
   c. In Locate File Resource, click **Next**. The Add a Report Resource page shows that uploading the file was successful. The server recognized the type (resource bundle) and name (sales_ro.properties) of the selected resource.
   d. Enter the following information:
      - Name – `sales_ro.properties`
      - Resource ID – `sales_ro.properties`
      - Description – `Romanian resource bundle`
   e. Click **Next**. Controls & Resources lists all the files.



Add JasperReport: New Complex Report

Controls & Resources

Locate the controls and resources used by your report.

Resources		
Scriptlet	JAR	Remove
sales.properties	Resource Bundle	Remove
sales_ro.properties	Resource Bundle	Remove
SalesByMonthDetail	JRXML	Added
Logo	Image	Added

Add Resource...

Input Controls

Add Input Control...

Figure 5-13 List of Detected and Undetected File Resources

If you want to upload a different file for a named resource, click its resource ID in the **Resources** list and locate the new file or repository object. You can change the name and description of the resource, but not its resource ID. If there's a mistake in a resource ID:

- Locate the ID in the list of resources on the Controls & Resources page, and click **Remove**.
- Re-add the resource, entering the correct resource ID.

5.3.2 Adding Input Controls

Input controls are graphical widgets the server displays with the report. Input controls perform the following functions:

- Prompt the user for input
- Validate the format of the input
- Pass the input to the report

Based on the input, the server modifies the WHERE filter clauses in SQL parametrized queries.

Input controls correspond to the parameters defined in JRXML reports, such as `$P{name}`. The server maps the value the user enters for the input control to the parameter of the same name. If you define an input control in the JasperReport and the server can't find a parameter by the same name in the JRXML, the input control won't function when the report runs.

The JRXML can define a default value for the input control. To prevent users from changing the default, you can make the input control read-only or invisible.

When you create an input control, you provide a datatype. Datatypes define the expected input (numbers, text, date, or date/time) and can include range restrictions that the server enforces. The server uses the datatype to classify and validate the data.

To define a datatype, set properties on the Set the Datatype Kind and Properties page. Properties differ for other datatypes that appear on the page.

Type	The classification of the data, which can be `Text`, `Number`, `Date`, or `Date-Time`.
Name	The name of the datatype.
Resource ID	The unique ID of the datatype that you cannot edit.
Description	Any additional information you provide about the datatype.
Pattern	A regular expression that restricts the possible values of the field for the `Text` data type.
Minimum value	The lowest permitted value for the field.
Maximum value	The highest permitted value for the field.
Minimum Is Strict	If checked, the maximum value itself is not permitted; only values less than the maximum value are permitted.
Maximum Is Strict	If checked, the minimum value itself is not permitted; only values greater than the minimum value are permitted.

After determining the list of values to be presented to the user, choose one of these widget types for the input control:

- Boolean – A check box widget for entering a yes/no value.
- Single value – A text, number, date, or date/time widget. Input can be constrained to a minimum value, maximum value, or both. Text input can also be constrained by a matching pattern. A text box widget for entering a value, or a calendar for selecting date and date/time.
- Multiple values – To present a static or a dynamic list of values to the user, choose one of these:
  - Drop-down list to select a single value
  - Radio buttons to select a single value
  - Multi-select list to select multiple values
  - Check boxes to select multiple values

The query in the SalesByMonth.jrxml file has several input control parameters, one for each type of input control. These procedures show you how to add each type to the report unit.

5.3.2.1 Adding a Text Input Control

The simplest input control is a text box. In this example, the datatype for the input value is a number; the server verifies that the user enters a number into the text box.

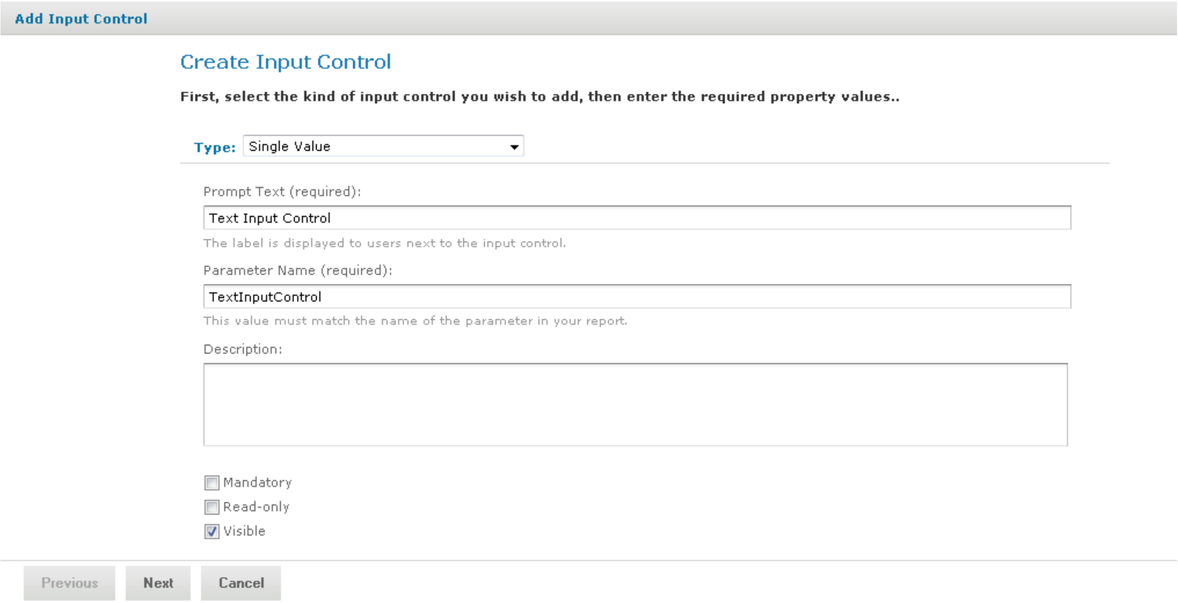
To add a text input control to the complex report example:

1. After completing steps in **“Uploading Undetected File Resources” on page 174**, click **Controls & Resources** in the JasperReport wizard.
2. On the Controls & Resources page, click **Add Input Control**. The Locate Input Control page appears.
3. Select **Define an Input Control in the next step**.
4. Click **Next**.
5. On the Create Input Control page, accept the default (**Single Value**) from the **Type** drop-down.

6. Enter the other properties for the input control:

   The name is referenced in the main JRXML file, so enter it exactly as shown.

   - Prompt Text – The label the user sees next to the widget for this input: `Text Input Control`
   - Parameter Name – The name of the report parameter that receives the user value: `TextInput`
   - Description – An optional description that appears only within the report wizard: leave blank in this example.
   - Mandatory, Read-only, Visible – A setting that determines how the input control is displayed: check only **Visible**.



Add Input Control

Create Input Control

First, select the kind of input control you wish to add, then enter the required property values..

Type: Single Value

Prompt Text (required): Text Input Control

The label is displayed to users next to the input control.

Parameter Name (required): TextInputControl

This value must match the name of the parameter in your report.

Description:

□ Mandatory
□ Read-only
☑ Visible

Figure 5-14 Properties of the Text Input Control

To reuse an input control, add it to the repository independent of any report using **Add Resource > Input Control**. Before using the input control in a report, check that the parameter name in the JRXML matches the name in the Create Input Control page. If it doesn't the server can’t run the report.

7. Click **Next**.
8. In Locate Datatypes, select **Define a DataType in the next step** and click **Next**:

Instead of defining a datatype, you can use one in the repository if its type and range are compatible with your input control.

9. In **Set the Datatype Kind and Properties**, enter the properties for the datatype:

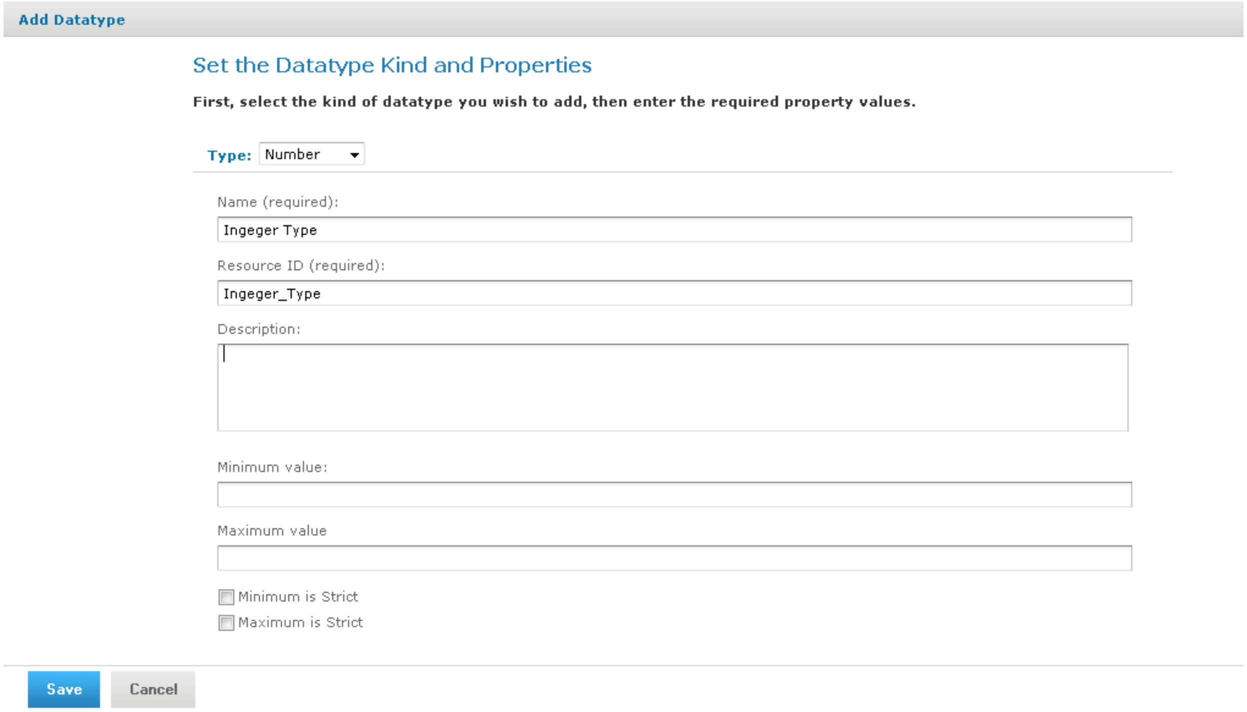
   a. In **Type**, select **Number** from the drop-down as the type of data the user can enter.

The number format allows users to enter integers and decimals.

b. Enter a name – `Integer Type`

c. Enter a resource ID – `Integer_Type`

The name and resource ID are required, but are visible only when defining the input control.



Set the Datatype Kind and Properties

First, select the kind of datatype you wish to add, then enter the required property values.

Type: Number

Name (required):

Ingeger Type

Resource ID (required):

Ingeger_Type

Description:

Minimum value:

Maximum value

□ Minimum is Strict

□ Maximum is Strict

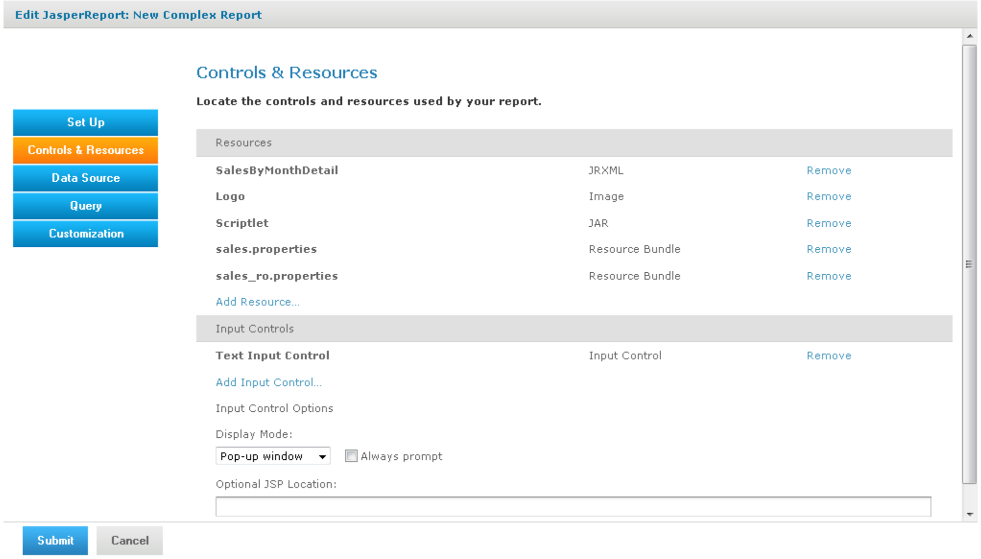
Figure 5-15 Integer Datatype Properties

d. Leave these properties blank in this example:

- Description – An optional description that appears only within the report wizard.
- Minimum value – The lower bound of the value the user may enter.
- Maximum value – The upper bound of the value the user may enter.
- Minimum is strict – Means the minimum value itself is not allowed.
- Maximum is strict – Means the maximum value itself is not allowed.

10. Click **Save**.

The Controls & Resources page now lists the Text Input Control.



Edit JasperReport: New Complex Report

Controls & Resources

Locate the controls and resources used by your report.

Resources		
SalesByMonthDetail	JRXML	Remove
Logo	Image	Remove
Scriptlet	JAR	Remove
sales.properties	Resource Bundle	Remove
sales_ro.properties	Resource Bundle	Remove

Input Controls		
Text Input Control	Input Control	Remove

Input Control Options

Display Mode: Pop-up window ▾ □ Always prompt

Optional JSP Location:

Figure 5-16 Text Input Control in Input Controls List

5.3.2.2 Adding a Simple Check Box Input Control

A check box input control accepts true/false (boolean) input from the user.

To add a simple check box input control to the complex report example:

1. Continuing with the previous example, on the Controls & Resources page, click **Add Input Control**.
2. On the Locate Input Control page, click **Define an Input Control in the next step**.
3. Click **Next**.
4. On the Create Input Control page, select **Boolean** from the **Type** drop-down.
5. Enter the other properties:
   - Prompt Text – `Check Box Input Control`
   - Parameter Name – `CheckboxInput`. Enter the parameter name exactly as shown because the main JRXML file references this name.
   - Description – Leave blank in this example.
   - Mandatory, Read-only, Visible – Check only visible.
6. Click **Submit**. The Controls & Resources page appears with the new check box input control.

5.3.2.3 Adding a Drop-Down Input Control

The drop-down input control, also called a list, gives the user a pre-determined list of choices. As a report designer, you make these decisions about a drop-down input control:

- To present a single-select or multi-select list to the user
- To present a single choice as a drop-down list or a set of radio buttons
- To present a multi-select control as a multi-select list or a set of check boxes

Radio buttons and check boxes usually work well for five or fewer choices. This example shows how to create an input control that presents three choices in a drop-down list. You can create a new list of values for this input control or use a list of values in the repository.

To add a drop-down input control to the complex report example:

1. Continuing with the previous example, on the Controls & Resources page, click **Add Input Control**.
2. On the Locate Input Control page, click **Define an Input Control in the next step**.
3. Click **Next**.
4. On the Create Input Control page, select **Single-select List of Values** from the **Type** drop-down.
5. Enter the other properties:
   - Prompt Text – `List Input Control`
   - Parameter Name – `ListInput` Enter the parameter name exactly as shown because the main JRXML file references this name.
   - Description – Leave blank in this example.
   - Mandatory, Read-only, Visible – Check only visible.
6. Click **Next**.
7. On the Locate List of Values page, select **Define a list of values in the next step**.

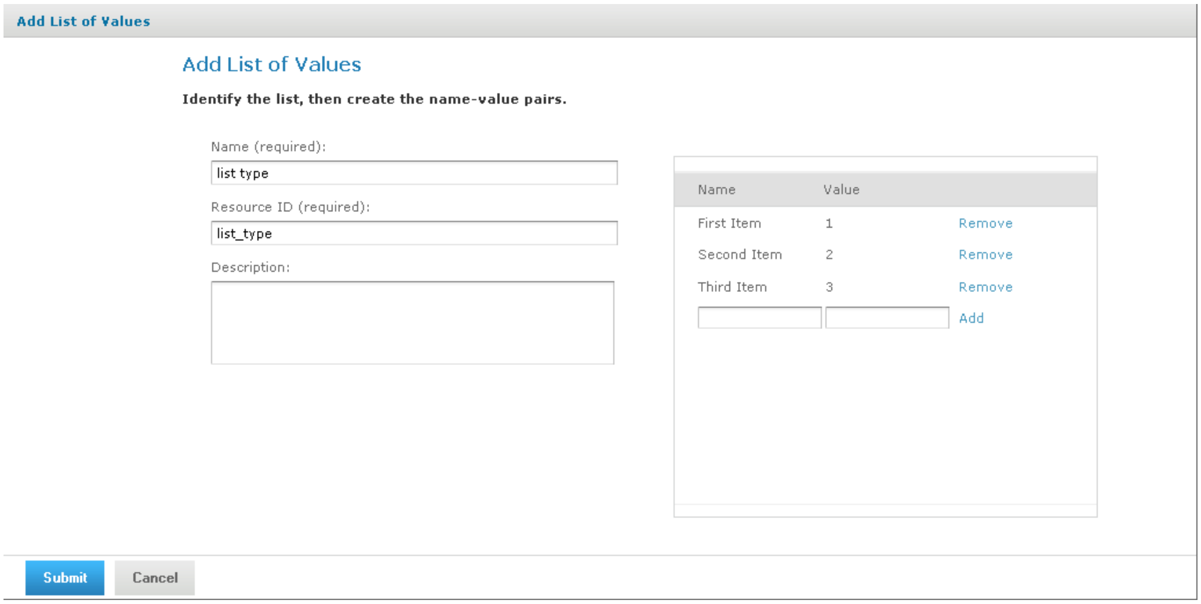
   Instead of defining a list of values, you can use one in the repository if its values are compatible with the parameter defined in the JRXML report.

8. Click **Next**.
9. On the Add List of Values page, enter a name, resource ID, and optional description for the list of values. These properties aren't visible outside of the input control. Enter these values:
   - Name – `list type`
   - Resource ID – `list_type`
   - Description – Leave blank in this example.
10. In the **Name Value** panel, enter names and values to present as choices to the user:
    - Enter unique names. The server requires unique names to distinguish which item the user chose.
    - Enter values of the type that match the parameter definition in the JRXML report.

    After entering a name and value, click **Add**. If you make a mistake click **Remove**.

    For this example enter:

    - Name `First Item` with value `1`.
    - Name `Second Item` with value `2`.
    - Name `Third Item` with value `3`.



Add List of Values

Add List of Values

Identify the list, then create the name-value pairs.

Name (required): list type

Resource ID (required): list_type

Description:

Name	Value	
First Item	1	Remove
Second Item	2	Remove
Third Item	3	Remove
		Add

Figure 5-17 Definition of the List of Values

11. Click **Submit**. The Controls & Resources page appears with the new List Input control.

5.3.2.4 Adding a Date Input Control

This example uses a datatype from the sample data in the repository.

To add a date input control to the complex report example:

1. On the Controls & Resources page click **Add Input Control**.
2. On the Locate Input Control page select **Define an Input Control in the next step**, then click **Next**.
3. On the Create Input Control page, select **Single-Value** from the **Type** drop-down.
4. Enter the other properties:
   - Prompt Text – `Date Input Control`
   - Parameter Name – `DateInput` Enter the parameter name exactly as shown because the main JRXML file references this name.
   - Description – Leave blank in this example.
   - Mandatory, Read-only, Visible – Check only Visible.
5. Click **Next**.
6. On the Locate Datatypes page select **Select a Datatype from the Repository**.
7. Click **Browse**.
8. In Select Resource from Repository, expand Input Data Types, and select the **Date Datatype**.
9. Click **Select**. The Locate DataTypes page shows the location of this datatype in the repository, /datatypes/DateDatatype.
10. Click **Next**. The Controls & Resources page appears with the new Date Input Control.

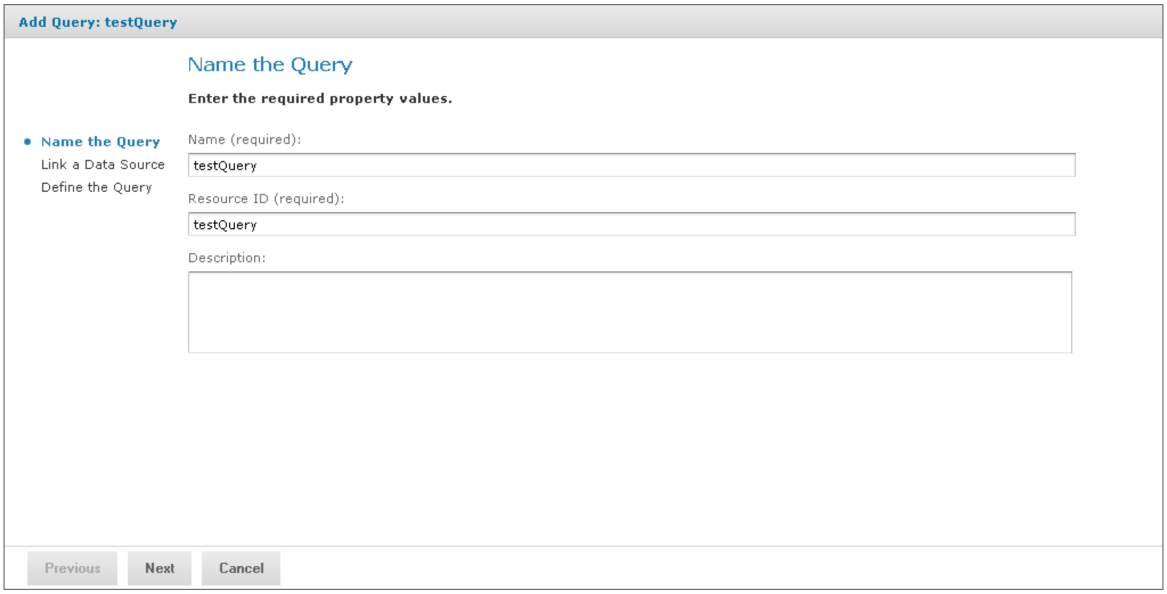
5.3.2.5 Adding a Query-Based Input Control

A query-based input control presents a dynamically-created list of choices to the user. The server performs a query whose results are used to create the list of choices. You must perform the following tasks:

- Configure the query.
- Designate how to display the results in the input control.
- Specify the value to pass as the corresponding parameter.

To add a query-based input control to the complex report example:

1. On the Controls & Resources page, click **Add Input Control**.
2. On the Locate Input Control page, select **Define an Input Control in the next step**.
3. Click **Next**.
4. On the Create Input Control page, select **Single-select Query** from the **Type** drop-down.
5. Enter the naming properties for the input control:
    - Prompt Text – `Query Input Control`
    - Parameter Name – `QueryInput` Enter the parameter name exactly as shown because the main JRXML file references this name.
    - Description – Leave blank in this example.
    - Mandatory, Read-only, Visible – Use the default settings in this example.
6. Click **Next**. The Locate Query page appears. Options are:
    - To locate a reusable query in the repository
    - To define a new query dedicated to this input control
7. For this example, select **Define a Query in the next step.**
8. Click **Next**.
9. On the Name the Query page, enter naming properties for the new query. For this example, enter `testQuery` in both the Name and Resource ID fields.



Add Query: testQuery

Name the Query

Enter the required property values.

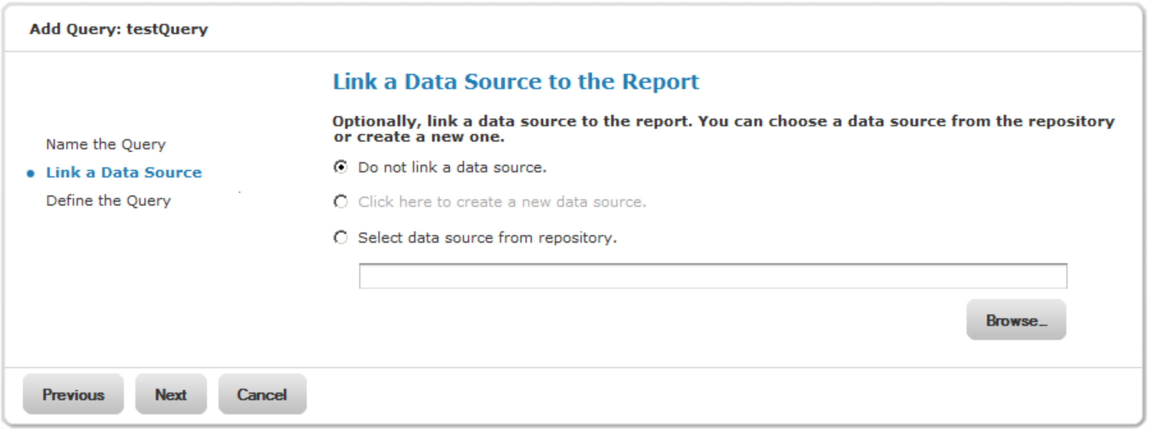
Name (required): testQuery

Resource ID (required): testQuery

Description:

Figure 5-18 Entering a Query Name

10. Click **Next**. The Link a Data Source to the Report page appears. Options are:
    - To use the same data source for the input control as you use for the report
    - To define a new data source, dedicated to this input control
    - To select a reusable data source from the repository
11. For this example, select **Do not link a data source** to use the same data source for the input control as you use for the report. You will select the data source for the report in **"Selecting a Data Source for Running the Complex Report" on page 186**.



Add Query: testQuery

Link a Data Source to the Report

Optionally, link a data source to the report. You can choose a data source from the repository or create a new one.

● Do not link a data source.

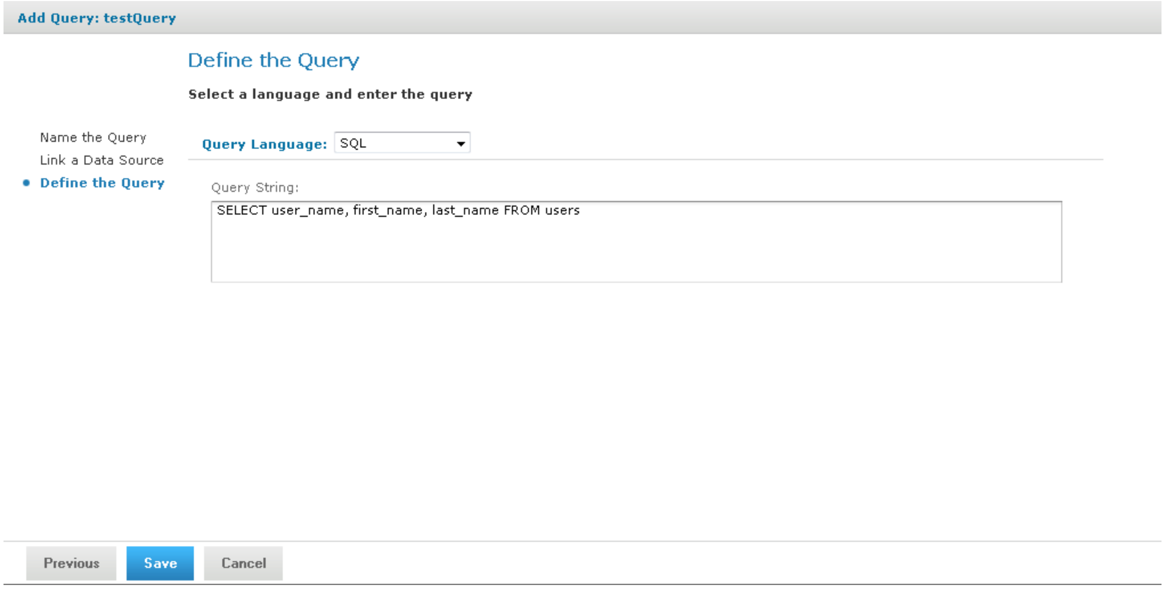
○ Click here to create a new data source.

○ Select data source from repository.

Figure 5-19 Data Source Link for the Query Input Control

12. Click **Next**.
13. On the Define the Query page, select **SQL** from the Query Language drop-down.
14. Enter this Query String to retrieve the labels and values to be displayed for this input control:

```
SELECT user_name, first_name, last_name FROM users
```



Add Query: testQuery

Define the Query

Select a language and enter the query

Query Language: SQL

Query String:

SELECT user_name, first_name, last_name FROM users

Figure 5-20 Query String Definition

15. Click **Save**.
16. For each row in the results of the query, the server presents a single value, such as Sarah Smith, in the input type widget (drop-down, radio buttons, multi-choice, check boxes). On the Query Information page, name the database columns to comprise the input value presented to the user. The column names must exactly match those in the `SELECT` clause of the query string:
    a. In the Value Column enter the `user_name`.
    b. In the Visible Column enter `first_name`.
    c. Click **Add**.
    d. In the Visible Column enter `last_name`.
    e. Click **Add**.

    For each column you want to display as a choice, enter the name then click **Add**. If you make a mistake click **Remove**.
17. Click **Submit**. The Controls & Resources page displays all the resources, including the new input controls. **Figure 5-21 on page 186** shows these resources.

5.3.2.6 Setting the Input Control Options

In this procedure, you set the display mode in **Input Control Options** at the bottom of the Controls & Resources page. These options appear only after you add an input control to the report. **Figure 5-21 on page 186** shows these options.

To configure the appearance of the input controls for the complex report example:

1. Select **Pop-up window**.

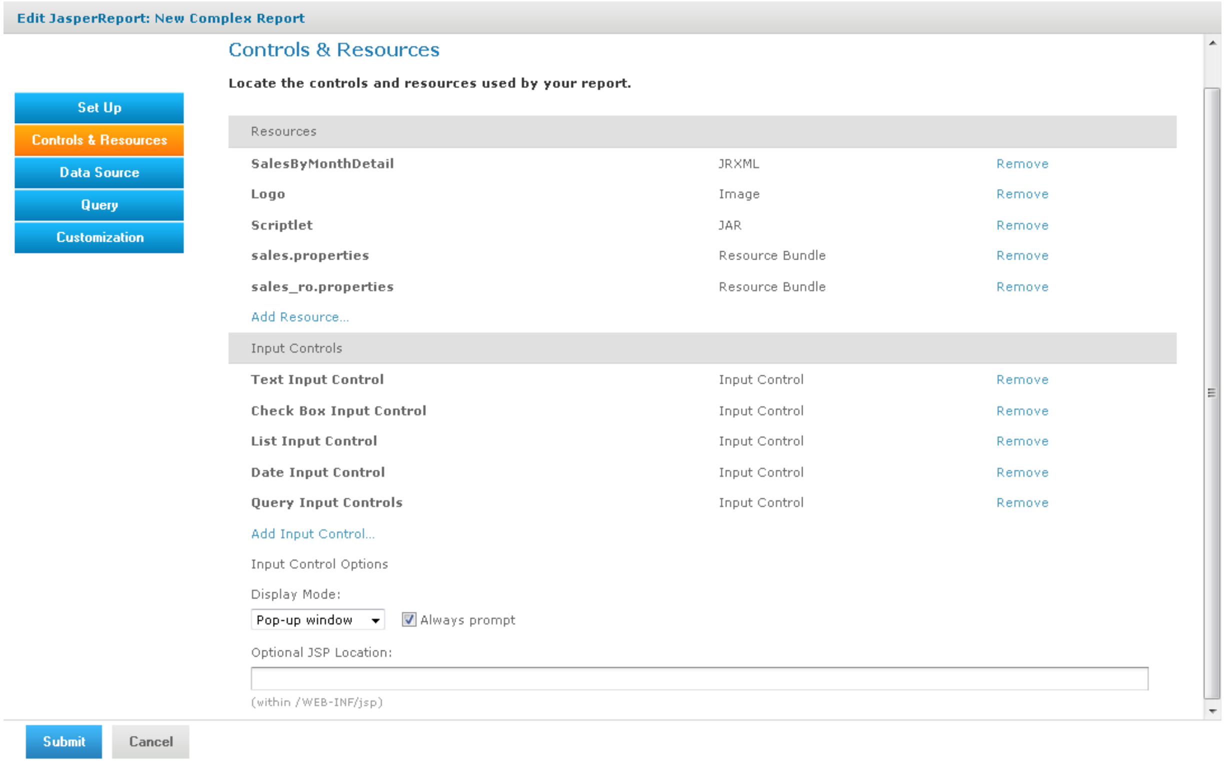
    You can also select **Separate page** to display the input controls in a separate browser window, **Top of page** to display them above the report, or **In page** to display them on the side of the report.

2. Check **Always prompt** when you want the server to display the Input Controls dialog to prompt the user when the report runs.

    The definition of input controls in this example specified Visible and not Mandatory. When input controls aren’t mandatory and Always prompt isn’t checked on the Controls & Resources page, the user must click the Options button in the report viewer to change input controls; otherwise the report runs with default input controls

3. Leave **Optional JSP Location** blank for this example.

    You can use the Optional JSP Location option to specify the path to a JSP file that affects the appearance of the input controls.



Edit JasperReport: New Complex Report

Controls & Resources

Locate the controls and resources used by your report.

Resources		
SalesByMonthDetail	JRXML	Remove
Logo	Image	Remove
Scriptlet	JAR	Remove
sales.properties	Resource Bundle	Remove
sales_ro.properties	Resource Bundle	Remove

Add Resource...

Input Controls		
Text Input Control	Input Control	Remove
Check Box Input Control	Input Control	Remove
List Input Control	Input Control	Remove
Date Input Control	Input Control	Remove
Query Input Controls	Input Control	Remove

Add Input Control...

Input Control Options

Display Mode:

Pop-up window ☑ Always prompt

Optional JSP Location:

(within /WEB-INF/jsp)

Figure 5-21 Input Controls and Resources

Select the data source to finish the complex report example.

5.3.3 Selecting a Data Source for Running the Complex Report

You select a data source to retrieve data for the report and the query input control; otherwise the report and the list of users in the query input control will be blank.

To select a data source and run the complex report:

1. On the Controls & Resources page of the JasperReport wizard, select **Data Source**.
2. On the Locate Data Source page, choose **Select data source from repository**.
3. Click **Browse**, choose **Organization > Data Sources > JServerJNDI Data Source**, and lick **Select**.
4. On Link a Data Source to the Report, click **Submit**.
5. On the Locate Query Page, click **Submit** again to save the complex report.

   Skip the Query and Customization pages of the JasperReport wizard to use the default settings on those pages.

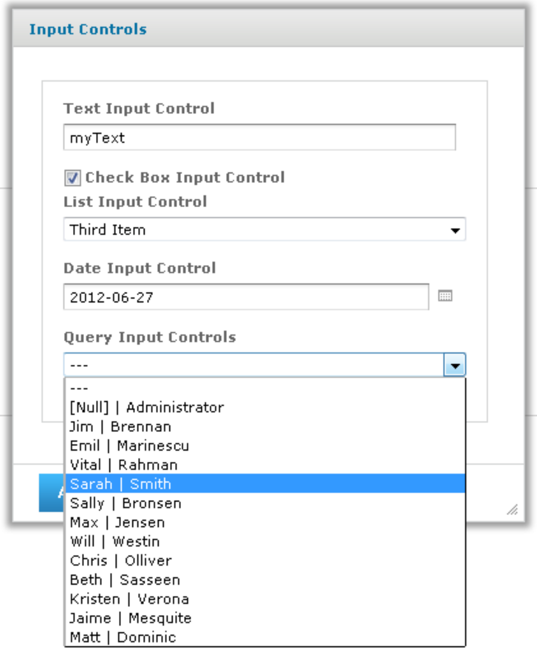
   The server validates the report and a message appears indicating that the report was added to the repository.
6. In the Repository, click the name New Complex Report to run and view the report.

   Input controls appear.
7. Enter these input values, as shown in **Figure 5-22**:
   a. Text Input Control: `myText`
   b. Check Box Input Control: Check the checkbox.

c. List Input Control: Select **Third Item**.

d. Date Input Control: Click 📅 and select December 31, 2010.

e. Query Input Control: Select **Sarah Smith** from the drop-down.



Input Controls

Text Input Control: myText

☑ Check Box Input Control

List Input Control: Third Item

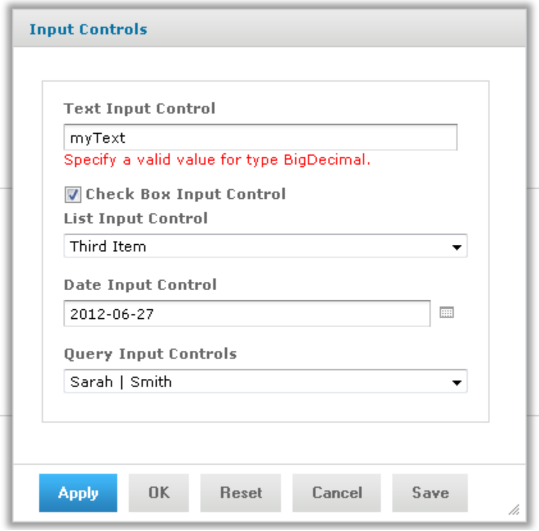
Date Input Control: 2012-06-27

Query Input Controls: ---

Figure 5-22 Input Controls Dialog for the New Complex Report

8. Click **OK** or **Apply** to run the report with the selected input, including the incorrect non-numerical input for the Text Input Control.

   The server enforces the proper format defined for each input control. You defined the Text Input Control as a numeric type, so it accepts only valid numbers, as indicated by the message to specify a valid float number, as shown in **Figure 5-23**.



Text Input Control

myText

Specify a valid value for type BigDecimal.

☑ Check Box Input Control

List Input Control

Third Item

Date Input Control

2012-06-27

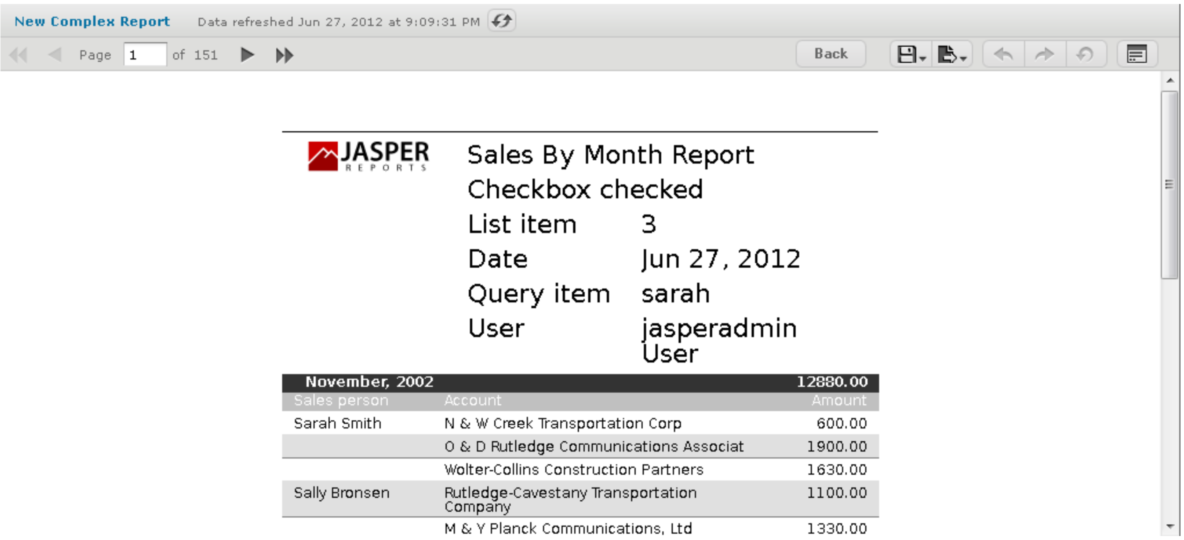
Query Input Controls

Sarah | Smith

Figure 5-23 Invalid Input Message

9. In Text Input Control, enter 3 and click **OK** or **Apply**.

   The sample report includes a header that displays the value of each parameter received from the input controls. Values and labels appear in the language specified by the active resource bundle, in this case English.



Sales By Month Report

Checkbox checked

List item	3
Date	Jun 27, 2012
Query item	sarah
User	jasperadmin User

November, 2002		12880.00
Sales person	Account	Amount
Sarah Smith	N & W Creek Transportation Corp	600.00
	O & D Rutledge Communications Associat	1900.00
	Wolter-Collins Construction Partners	1630.00
Sally Bronsen	Rutledge-Cavestany Transportation Company	1100.00
	M & Y Planck Communications, Ltd	1330.00

Figure 5-24 Output Controlled by Input

In the report viewer, you can open the Input Controls dialog at any time by clicking the **Options** button. Click **OK** to run the report using the chosen values and close the Input Controls dialog; click **Apply** to run the report using the chosen values, but keep the Input Controls open for choosing other values and re-running the report.

If you get an error when you run the report, open it for editing as described in **"Editing JRXML Report Units" on page 189**. Review your settings. If you can't find the problem, edit the SalesByMonth sample report (in the repository at /reports/samples) and compare its settings to your report.

To see the message written by the scriptlet JAR on the last page of the report, click ▶▶ in the report viewer.

5.4 Adding Cascading Input Controls to a Report

JRXML-based reports can include input controls that have dynamic values. The values depend on a user's selection in other input controls. For example, a report has input controls for country, state, and city. The options in the State input control depend on the value selected in the Country input control. When the user selects a state, the list of City values includes only those in the selected state. These cascading input controls use queries to determine the values to display in each input control field.

To use input controls as parameters for a query that populates another input control, you use a special syntax to reference a parameter name in the input control's query. The syntax is identical to the `$P{parameter_name}` and `$X{...} syntax` used in queries for .

For example, a report returns data identified by country and city. It includes input controls called COUNTRY and CITY. COUNTRY is a query-based input control that returns the list of countries in the data source. CITY is also a query-based input control, but its query uses COUNTRY as input:

```
select address_city from accounts where $X{EQUALS, address_country, COUNTRY}
```

When the user selects a country from the COUNTRY input control, the value selected is used by the query of the CITY input control. The CITY input control is refreshed to show the list of cities for the chosen country. Making two selections from smaller lists is much clearer and quicker for report users. For an example of viewing a report that has cascading input controls, see **"Cascading Input Controls" on page 73**.

Note that there are other ways to use a parameter in a query. For details about the $P and $X syntax and an example of creating a cascading input control, see the *JasperReports Server Administrator Guide*.

5.5 Editing JRXML Report Units

After you add a report unit to the repository, you can edit any of its elements, including file resources and input controls. This example modifies the display text of the ambiguous Text Input Control.

To edit the complex report example:

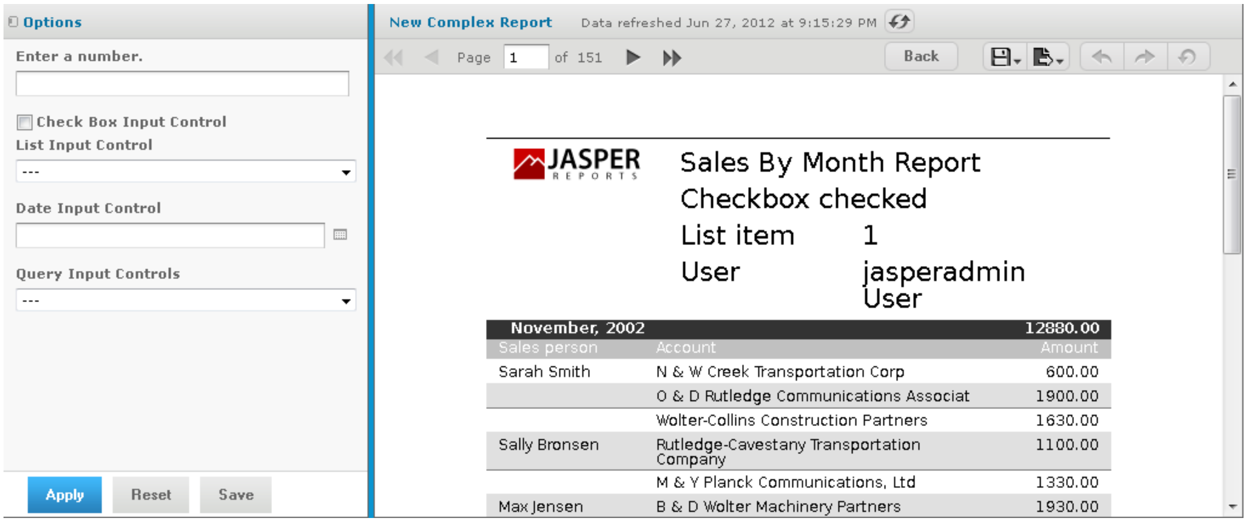
1. Log into the server as an administrator and select **View > Repository**

If you log in as a user, you can edit a report that you created. This example requires an administrator login because an administrator created the complex report.

2. Search or browse the repository to locate the report. In this example, go to **Organization > Reports**.
3. Right-click the New Complex Report and select **Edit** from the context menu. The JasperReport wizard opens the report unit.
4. Navigate to the page of the wizard for making the change; in this example, click **Controls & Resources**.

5. Make changes to an input control prompt and the display mode of the input controls, for example:
   a. Click the name of the **TextInput** control. The Locate Input Control page shows that this input control is locally defined.
   b. Click **Next**. The Create Input Control page appears.
   c. Change the contents of the Prompt Text field to `Enter a number`. Click **Next**. The Locate Datatypes page appears. You can select a different datatype from the repository. For this example, accept the existing datatype setting.
   d. In Locate Datatypes, click **Next**.
   e. In Set the Datatype Kind and Properties, click **Save** to accept the datatype property settings.
6. On the Controls & Resources page:
   a. Change the Display Mode to **In Page**.
   b. Clear the **Always prompt** check box.
   c. Click **Submit**.
7. Run the New Complex Report again.

   Instead of appearing in a pop-up before the report, the input controls appear in the Filters panel of the report. In **Figure 5-25** you can see the new prompt **Enter a number** for the text input control. Because none of the input controls in this example are required, the report can display with blank input controls. Enter values and click **Apply** to modify the report output according to your input.



Sales By Month Report
Checkbox checked
List item 1
User jasperadmin User

November, 2002		12880.00
Sales person	Account	Amount
Sarah Smith	N & W Creek Transportation Corp	600.00
	O & D Rutledge Communications Associat	1900.00
	Wolter-Collins Construction Partners	1630.00
Sally Bronsen	Rutledge-Cavestany Transportation Company	1100.00
	M & Y Planck Communications, Ltd	1330.00
Max Jensen	B & D Wolter Machinery Partners	1930.00

Figure 5-25 Output of the Modified Report

5.6 Localizing Reports

You can adapt reports to global audiences by localizing input controls and field names:

- Input controls – The server supports multi-lingual prompts and static lists of values in reports.
- Field names – The server supports multi-lingual field names in reports.

A `$R`expression that you write in the report design triggers linguistic changes in the report output for different locales. Each `$R` expression refers to a name-value pair (your translations) in a resource bundle. A resource bundle file is a text file that has a `.properties` extension. You create a resource bundle in Jaspersoft Studio or a text editor. You set the base name of the resource bundle in the header of the JRXML file:

```
<jasperReport name=”StoreSales” pageWidth=”595” pageHeight=”842” columnWidth=”515”
leftMargin=”40” rightMargin=”40” topMargin=”50” bottomMargin=”50”
resourceBundle=”simpleTable”>
```

For example, `simpleTable` is the base name of the resource bundle file for this report. If you prefer using a graphical user interface to coding in XML, use Jaspersoft Studio to set the base name of the resource bundle.

5.6.1 Running a Localized Report

In this procedure, you run the Romanian version of the complex report that you added in **“Uploading Undetected File Resources” on page 174**.

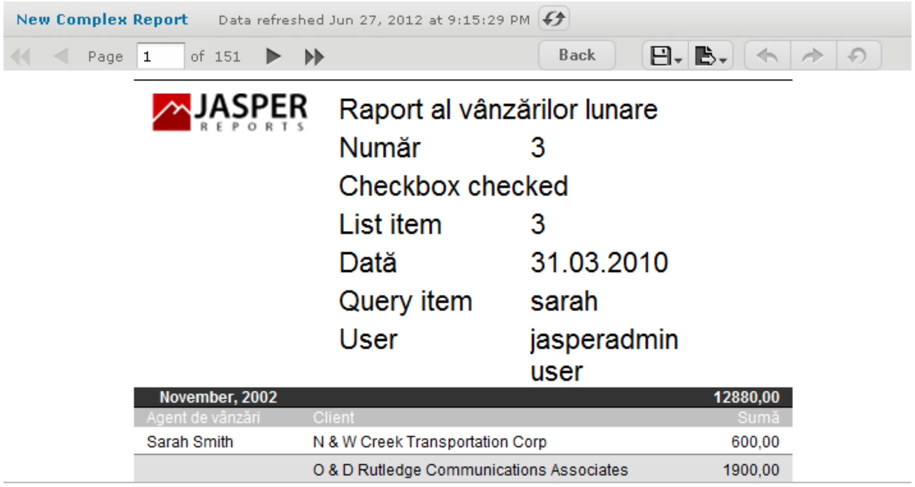
To run the Romanian version of the complex report:

1. Choose the Romanian locale on the login page of the server, and login as an administrator.
2. Click **View > Repository**, and navigate to **Organization > Reports**.
3. Click the name of the complex report, New Complex Report. The Input Controls dialog appears.
4. Enter input control values:
    a. Text Input Control: `3`
    b. Check Box Input Control: Check the check box.
    c. List Input Control: Select **Third Item**.
    d. Date Input Control: Click 📅 and select March 31, 2010.
    e. Query Input Control: Select **Sarah Smith** from the drop-down.
    f. Click **OK**.

    The fields in the title band and column names (sales person, sales account, and sales amount), shown in **Figure 5-26**, appear in the language set by the Romanian resource bundle sales_ro.properties:

```
title=Raport al v\u00E2nz\u0103rilor lunare
sales.person=Agent de v\u00E2nz\u0103ri
sales.account=Client
sales.amount=Sum\u0103
param.number=Num\u0103r
param.date=Dat\u0103
```

The currency and dates in the report output header map to Romanian locale settings.



Raport al vânzărilor lunare

Număr	3
Checkbox checked	
List item	3
Dată	31.03.2010
Query item	sarah
User	jasperadmin user

November, 2002		12880,00
Agent de vânzări	Client	Sumă
Sarah Smith	N & W Creek Transportation Corp	600,00
	O & D Rutledge Communications Associates	1900,00

Figure 5-26 A Report Localized for the Romanian Locale

By default, the web interface elements appear in US English when you choose an unsupported locale, such as the Romanian locale. If you choose a supported language, the web interface elements appear in that language. Supported languages are Chinese (Simplified), French, German, Japanese, and Spanish. You can customize the server to support additional languages. You can translate the web interface into a different language, server property names, and messages in another language. For some locales, you may also need to change the default locale and time zone. For more information about localizing the server, see the *JasperReports Server Administrator Guide*.

5.6.2 Adding Multi-lingual Prompts to Input Controls

This section describes how to create an Ad Hoc report that prompts for input in multiple languages. The tasks are:

- Set the base name of the resource bundles in the JRXML Topic.
- Create resource bundles that contain translations for the prompts.
- Create an Ad Hoc report based on the JRXML Topic.
- Edit an input control to make prompts multi-lingual.
- Upload the resource bundles.
- Run the report and use the localized input control.

The order of these tasks is important: Set the base name of the resource bundle in the JRXML Topic first, then create the Ad Hoc report. If you open the JRXML Topic after creating an Ad Hoc report, Jaspersoft Studio removes grouping or sorting of data if there is any.

To set the base name of the resource bundles in the JRXML Topic:

1. Start Jaspersoft Studio. In Jaspersoft Studio, click **Window >JasperReports Server Repository**. The Repository Navigator appears. This is where you set up a connection to the server.
2. Navigate to **Ad Hoc Components > Topics** and right-click the JRXML topic for this report: **Parametrized Report**.
3. Select **Copy**.

4. Navigate to the **Reports** folder, right-click and select **Paste**. The Parametrized Report topic appears in Reports.
5. To open the topic in the Designer tab, expand the Parametrized Report folder and double-click its main JRXML: **ParametersJRXML_label**. The Shipping Report appears in the Designer tab.
6. Click **Window > Report Inspector**.
7. In the Report Inspector, right-click the root node: **ParamMany** and choose **Properties**. In the main JRXML, ParamMany is the report name.

   In **Figure 5-27**, you can see the ParamMany root node and Properties context menu.

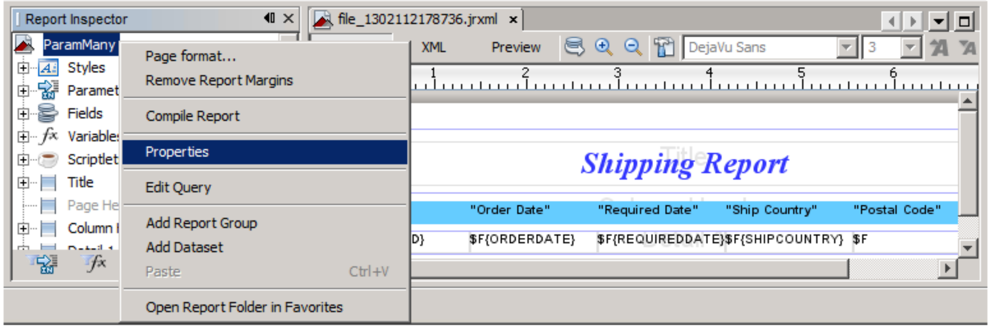
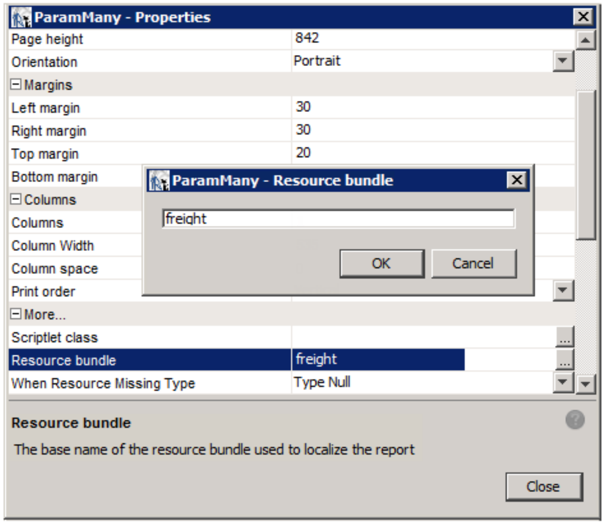


Figure 5-27 Selecting Properties of the ParamMany report

8. In the ParamMany Properties dialog, set the base name of the resource bundle to freight:
   a. Scroll down to the **Resource bundle** property.
   b. Click ....
   c. Enter `freight` and click **OK** .

In **Figure 5-28**, you can see the properties of the ParamMany report.


ParamMany - Properties

Page height	842
Orientation	Portrait
⊟Margins	
Left margin	30
Right margin	30
Top margin	20
Bottom margin	
⊟Columns	
Columns	
Column Width	
Column space	
Print order	
⊟More...	
Scriptlet class	
Resource bundle	freight
When Resource Missing Type	Type Null

ParamMany - Resource bundle

freight

Resource bundle

The base name of the resource bundle used to localize the report

Figure 5-28 Setting the Base Name of the Resource Bundle

9. Click **Close**.
10. Click **File > Save** to save the resource bundle name to the JRXML.
11. In the Repository Navigator, right-click **ParametersJRXML_label** and choose **Replace with Current Document**.

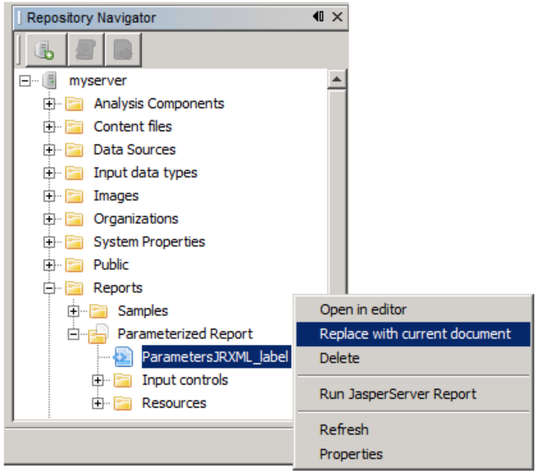


Figure 5-29 Saving the Current Document in iReport to the Repository

The modified JRXML with a base resource bundle name overwrites the Parametrized Report Topic in the repository.

To create resource bundles that contain translations for prompts:

1. In a text editor, create a new file for English translations.
2. Enter these name-value pairs in the file:

```
BundleCountry = Country
BundleDate = Date
BundleOrder= Order
```

3. Save the file as `freight.properties`.

The file name of the default (English) resource bundle consists of the base name of the resource bundle and the properties extension.

4. In a text editor, create a new file for French translations and enter these name-value pairs in the file:

```
BundleCountry = Pays
BundleDate = Date
BundleOrder = Pour ID
```

5. Save the file as `freight_fr.properties`.

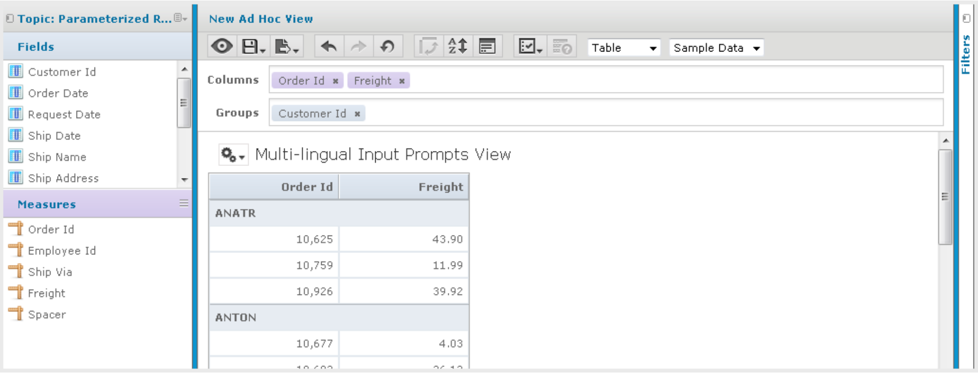
The file name of a localized resource bundle follows this Java naming convention:

<default_file_name>_<locale>.properties

- <default_file_name> is the base name of the resource bundle
- <locale> is a Java-compliant locale identifier

To create an Ad Hoc View based on the JRXML Topic:

1. Log into the server as administrator, and choose **Create > Ad Hoc View**.
2. In the Select Data wizard, click and navigate to **Ad Hoc Components > Topics** and choose **Parametrized Report**.
3. Click **Table.** The Parametrized Report topic (a blank report) opens in the Ad Hoc Editor.
4. In the **Measures** list, double-click these fields:
   - Order ID
   - Freight
5. In the **Fields** list, right-click **Customer Id** and select **Add as Group**.
6. Click **Click to add a title** and enter **Multi-lingual Input Prompts View.**



Multi-lingual Input Prompts View

Order Id	Freight
ANATR	
10,625	43.90
10,759	11.99
10,926	39.92
ANTON	
10,677	4.03

Figure 5-30 Creating an Ad Hoc View

7. Click and select **Save Ad Hoc view As and Create Report**. In the Save As dialog, select the **Ad Hoc Reports** folder, and enter:
   - Data View Name: `Multi-lingual Input Prompts View`
   - Data View Description: `A report that prompts for input in French and English.`
   - Report Name: `Multi-lingual Input Prompts View Report`
8. Browse to a location to save both the view and report.
9. Click **Default Report Template**.
10. Click **Save**.

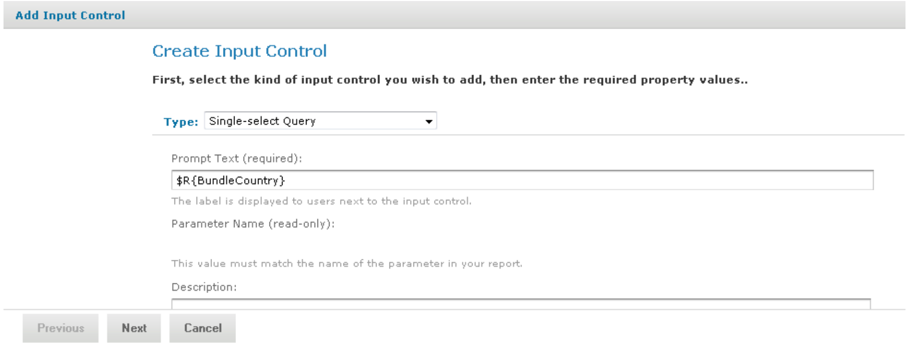
To edit an input control to make prompts multi-lingual:

1. In the server, click **View > Repository.**
2. Locate the Multi-lingual Input Prompts View Report, right-click it, and choose **Edit**. The JasperReport wizard appears.

3. Click **Controls & Resources**. The Controls & Resources page lists these input controls:
   - Country
   - RequestDate
   - OrderID
4. Click the **Country** input control.
5. On the Locate Input Control page, click **Next** to define an input control in the next step.
6. On the Create Input Control page, change the prompt text from `Country` to this expression:

   `$R{BundleCountry}`

   In **Figure 5-31** this expression is entered in the prompt text field.



Add Input Control

Create Input Control

First, select the kind of input control you wish to add, then enter the required property values..

Type: Single-select Query

Prompt Text (required):

$R{BundleCountry}

The label is displayed to users next to the input control.

Parameter Name (read-only):

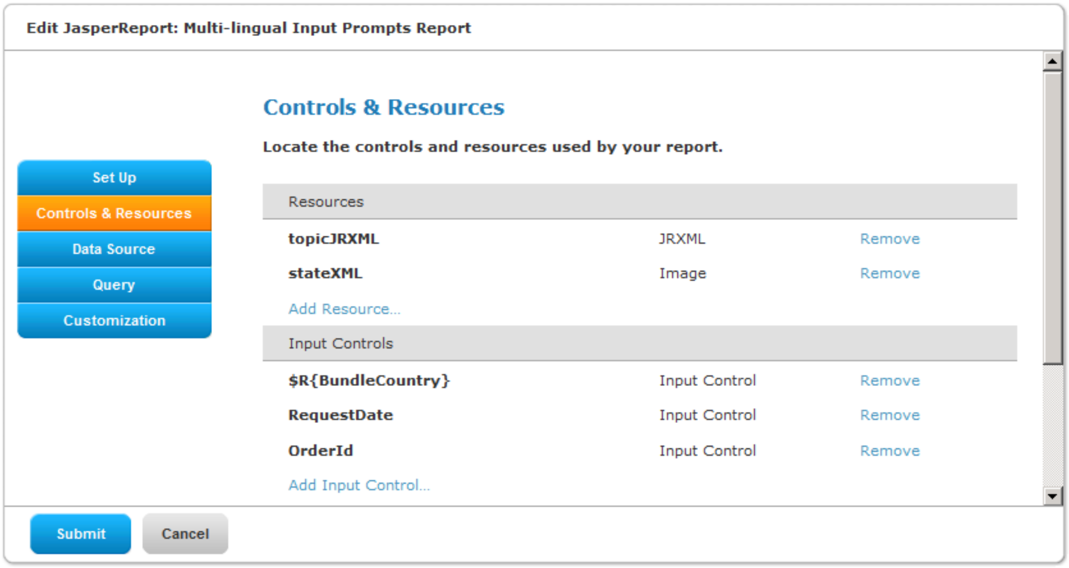
This value must match the name of the parameter in your report.

Description:

Figure 5-31 Entering a $R Expression in the Prompt Text Field

7. Click **Next**.
8. Accept the default settings on subsequent pages by clicking **Next** and **Save:**
   a. On the Locate Query page, click **Next**.
   b. On the Name the Query page, click **Next**.
   c. On the Link a Data Source to the Query page, click **Next**.
   d. On the Define the Query page, click **Save**.
9. On the Set Parameter Values page, click **Submit**.

   The Controls & Resources page now shows the `$R{BundleCountry}` expression instead of Country at the top of the list of input controls.



Edit JasperReport: Multi-lingual Input Prompts Report

Controls & Resources

Locate the controls and resources used by your report.

Resources		
topicJRXML	JRXML	Remove
stateXML	Image	Remove
Add Resource...		

Input Controls		
$R{BundleCountry}	Input Control	Remove
RequestDate	Input Control	Remove
OrderId	Input Control	Remove
Add Input Control...		

Figure 5-32 Input Controls Include One Multi-lingual Input Control

10. Change the prompt text of the other input controls in a similar manner:
    a. Repeat **step 4** through **step 6** to change the RequestDate and OrderID input controls to these expressions:

       RequestDate: `$R{BundleDate}`

       OrderID: `$R{BundleOrder}`
    b. Accept the default settings on subsequent pages of the JasperReport wizard by clicking **Next** and **Save**. The Controls & Resources page now shows the `$R` expressions for all three input controls.

To upload the resource bundles:

1. On the **Controls & Resources** page, click **Add Resource**. The Locate File Resource page appears.
2. Select **Upload a Local File.**
3. Click **Browse**, select the freight.properties file, and click **Open**.
4. On the Locate File Resource page, click **Next**.

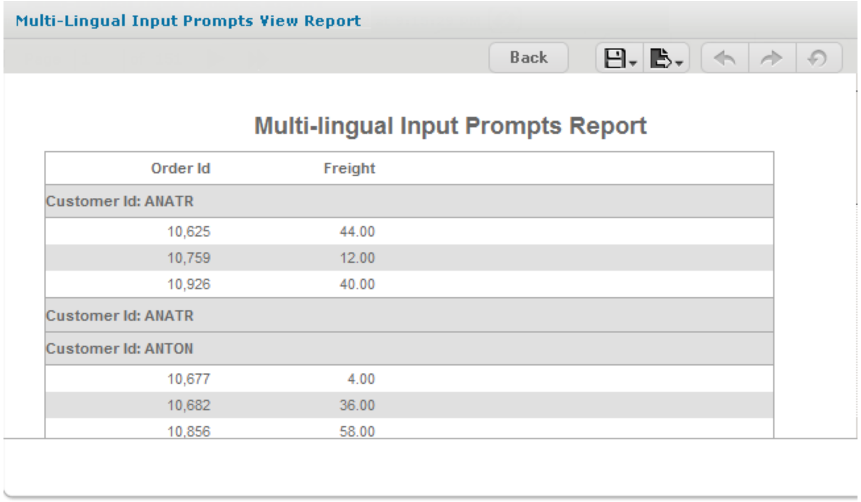
   On the Add a Report Resource page, freight.properties appears as the Selected Resource, indicating that the server automatically detected it as a resource bundle.
5. On the Add a Report Resource page, enter these properties:
   - Name – `freight.properties`
   - Resource ID – `freight.properties`
   - Description – `Default English resource bundle`
6. Click **Next**. The list of resources on the Controls & Resources page now includes the resource bundle freight.properties.
7. On Controls & Resources, click **Add Resource** again, but this time upload the French resource bundle:
   a. On the Controls & Resources page, click **Add Resource** again.
   b. Select **Upload a Local File**, click **Browse**, select the freight_fr.properties file you created, and click **Open**.
   c. In Locate File Resource click **Next**. The **Add a Report Resource** page appears.

d. Enter the following information:
- Name – `freight_fr.properties`
- Resource ID – `freight_fr.properties`
- Description – `French resource bundle`

8. Click **Next**. The Controls & Resources page shows the English and French resource bundles in the resources list.
9. Click **Submit**.

To run the report and use the localized input control:

1. Click **Log Out**.
2. On the Login page click **Show locale & time zone**.
3. Select the **en_US-English (United States)** locale and log into the server as an administrator.
4. Run the Multi-lingual Input Prompts Report. The report appears in the report viewer.



Multi-lingual Input Prompts Report

Order Id	Freight
Customer Id: ANATR	
10,625	44.00
10,759	12.00
10,926	40.00
Customer Id: ANATR	
Customer Id: ANTON	
10,677	4.00
10,682	36.00
10,856	58.00

Figure 5-33 Viewing the Report in English

5. Click **Options**.

   The input controls appear with English prompts.
6. Click **Log Out**.
7. On the Login page, click **Show locale & time zone**.
8. Select the **fr - French** locale, and log in as administrator.
9. Run the report again and click Options.

   In **Figure 5-34**, you can see the input controls that with English and French prompts.

Figure 5-34 Input Control Prompts in English and French

5.6.3 Reusing Resource Bundles

The location of a resource bundle determines whether it's reusable and the conditions under which the server uses it. To resolve a $R expression in a report, the server scans resource bundles at two levels in the order described in this table.

Order	Level	Description
1	Report/ Repository	A report-level bundle declares the resource bundle base name in the header of its JRXML file. The Romanian resource bundle is a report-level bundle. A repository-level bundle is independent of any specific report and can be linked to multiple reports, as described in **"Locale Bundles" on page 277**).
2	Server	A server-wide bundle typically contains server messages and labels of user interface components, as described in the *JasperReports Server Administrator Guide*. This bundle typically resides in the WEB-INF/bundles directory of the server.

First, the server searches the report/repository level and stops scanning resource bundles if it finds a resolution to the $R expression. If the server does not find a resolution, it scans the server level for the resource ID of the field and uses this ID.

5.6.4 Using Default Fonts in JasperReports Server

By default, the server uses three fonts for reports:

- DejaVu Sans
- DejaVu Serif
- DejaVu Sans Mono

Using the DejaVu fonts shipped with the server ensures availability of fonts in all environments; the PDF is pixel-perfect every time.

The DejaVu fonts replace the Java logical fonts used in previous versions of the server:

- SansSerif
- Serif
- Monospaced

SansSerif, Serif, Monospaced can still be used, but are deprecated because these Java logical fonts map to different TTF files in different environments, and run the risk of text being cut when exported to PDF due to font metric mismatches. Also, these Java logical fonts aren't recognized by some browsers, resulting in font substitutions. For example, Firefox in a Windows environment renders the SansSerif logical font as Serif.

When using the DejaVu fonts coming from font extensions, you don't need to set any other font attributes (such as the `pdfXXX` attributes) in the JRXML or specify font mapping. The font extension file that makes these fonts available sets font attributes and mapping.

For more information about DejaVu, refer to its SourceForge project at:

http://dejavu-fonts.org/wiki/index.php?title=Main_Page.

When you upload a TrueType font to the repository, the file name must include the correct extension (.TTF).

CHAPTER 6 CREATING DOMAINS

A Domain is a metadata layer that provides a business view of the data accessed through a data source. A Domain presents the data in business terms appropriate to your audience, and can limit access to data based on the security permissions of the person running the report. A Domain defined in JasperReports Server can be used to create reports, Ad Hoc views, and Domain Topics.

This chapter covers the process of creating a Domain and defining its contents. For instructions about creating Domain Topics and views based on Domains in the Ad Hoc Editor, see **"Creating a View from a Domain" on page 152**. Domains defined in the server can be accessed through Jaspersoft Studio as well.

This chapter contains the following sections:

- **Overview of Domains**
- **Example of Creating a Domain**
- **Example of Creating a Domain Using a Virtual Data Source**
- **Using the Add New Domain Page**
- **Using the Domain Designer**
- **Editing a Domain**

6.1 Overview of Domains

Production databases typically contain data in tables optimized for storage and retrieval. Columns with data relevant to users must be joined across several tables, filtered by business need, secured against unauthorized access, and labeled with user-friendly names. In relational databases, a view can perform some of these functions. In the server, a Domain performs all these functions and more, such as the localization of report text and permissions based on data values.

A Domain is a virtual view, created and stored in the server without modifying the data source. Through a Domain, users see columns that have been joined, filtered, and labeled for their business needs. Security policies limit the data values users can access through a Domain. Administrators define Domains, and users create reports based on Domains using either the Ad Hoc Editor or Jaspersoft Studio.

Typically, database administrators or business analysts create Domains, which are then used by end users who understand the structure of the raw data in the database. In the default server installation, those who create Domains must have organization admin privileges.

Domains, like Topics, are used in the Ad Hoc Editor as a basis for designing reports. Users can save a report based on a Domain for others to run, and can also save the settings in a Domain Topic so others can design similar reports.

Using a Domain instead of a Topic has the following advantages:

- Domains can be created directly through the server user interface.
- Domains can use virtual data sources that combine data residing in multiple JDBC or JNDI data sources into a single data source.
- Domain creators can write SQL queries, joins, filter expressions, and security files to specify exactly what data can be accessed, as well as labels and locale bundles to specify how the data appears.
- When designing a report based on a Domain, the server can optimize database access to allow editing of reports that access huge datasets.
- A Domain design typically includes data filtering, but users who create reports using the Domain can filter data even further. Users can select a subset of columns to appear in the Ad Hoc Editor and give them custom labels. Users can filter the data for each column.
- Users creating reports can design prompts for input from report readers to filter data that appears in the document.

For more information, see **"Creating a View from a Domain" on page 152** and **"Creating Topics from Domains" on page 159**.

6.1.1 Domain Use Cases

Typically, you base the decision to use Domains, Domain Topics, or saved reports on the complexity of the data and your business needs. Consider the following use cases:

- Case 1 – Domains and Domain Topics can be designed to give users great freedom in designing reports plus the security features that prevent unauthorized access to data.
- Case 2 – Administrators can create very targeted Domains and Domain Topics, then use repository access permissions to make sure users cannot modify them.

In the first case, a Domain could contain dozens of tables in several unjoined sets, perform very little filtering but define a strong data-level security policy. Users of the Domain might see only a single set of tables according to their security permissions. They could then perform their own filtering, relabel columns for their own needs, and save their settings as a Domain Topic for reuse in Ad Hoc views. In this case, the organization may have only one domain but many Domain Topics that users have created for specific needs.

In the second case, Domains can be used to perform complex joins, have formulas for calculating new columns, filtering data, and selecting a small set of columns for specific users or specific types of reports. Perhaps reports also need to be internationalized, so the administrator creates the corresponding locale bundles. In this case, there might be many specific Domains, and each would have corresponding locale bundles and a single Domain Topic. Users would have a wide variety of Domain Topics to choose from, each for a specific purpose, but no opportunity to access the Domains to perform their own filtering.

The preceding examples illustrate two extreme cases, but many scenarios combine both to some degree . The number of users in your production environment and their level of proficiency also determine your general use cases. A small number of users who understand their database might be given administrator privileges to define complex Domains as an extension of the Ad Hoc view design process. On the other hand, with large numbers of users or with users who have limited database skills, administrators want to create Domains that help the users meet their business goals.

The table in **"Creating Topics from Domains" on page 159** describes individual use cases for Domains, Domain Topics, and reports based on Domains.

6.1.2 Terminology

This chapter refers to both columns and fields. Conventionally, database tables are composed of columns, and the columns divide each row or record into fields. Some Domain operations refer to columns, others to fields, but these terms refer to the same concept from different perspectives. For example, a calculated field refers to a field that is computed from the other field values in the same row or record. But the effect of a calculated field in every row is to create a new calculated column. Similarly, operations such as joins and filters operate on designated field values in a row, but they are defined by the columns that determine the fields involved.

Within a Domain, columns are called items. Because an item may originate from derived tables or calculated fields, it may not correspond to a single column in the database.

Measures are columns or fields that display numeric values or aggregate values in reports. Measures are the quantitative values of a record, such as an amount, as opposed to qualitative values such as name or location. Having a field designated as a measure determines where the item can be placed in the report. For example, measures appear summarized in the center of crosstabs, whereas only non-measures, simply called fields, can be used as row and column groups of the crosstab. In Domains, all items based on numeric fields are designated as measures by default, but you can set them manually as well. By marking items as measures, Domains help report creators find and use the data they need in the Ad Hoc Editor.

6.1.3 Components of a Domain

A Domain is saved as an object in the repository. Like other repository objects, it has a name, optional description, and folder location specified at creation time. A Domain references the following components:

- A JDBC, JNDI, or virtual data source whose tables and columns are presented through the Domain. Data sources are selected from previously defined data sources in the repository.
- The Domain design that specifies tables and columns in the data source, queries for derived tables, joins between tables, calculated fields, and labels to display the columns.
- Optional locale bundles consisting of localized property files that present labels and descriptions in other languages.
- Optional security file that defines row and column-level access privileges based on users or roles.

The Domain design is either created using the Domain Designer or uploaded from an external XML file. Locale bundles and security files are uploaded from external files. The following sections describe the various dialogs for selecting, creating, or uploading the components of a Domain.

6.1.4 Sample Domains

The following sample Domains are provided in **Organization > Domains** in the default installation:

- Simple Domain – Demonstrates basic features of Domains
    - Based on the SugarCRM JNDI sample data source
    - Includes a design file that you can export, as described in **"The Domain Design File" on page 245**
    - Used to create Simple Domain Topic in the **Organization > Ad Hoc Components > Topics** folder, as described in **"Creating Topics from Domains" on page 159**
    - Shows how a report creator can see the items in a Domain through a Domain Topic

- SuperMart Domain – Demonstrates advanced features of Domains
  - Based on the Foodmart sample data source
  - Represents a complex Domain with many tables and joins
  - Includes a design file that you can export, as described in **"The Domain Design File" on page 245**
  - Includes the Domain security file and locale bundles that you can download and inspect, as described in **"The Domain Design File" on page 245**

You can use the samples to practice designing and using Domains and to see complex reports based on Domains.

6.2 Steps for Creating a Domain

Plan Domain features before you begin the creation process:

- Decide on the data source and define it in the server repository.
- If you need to combine several data sources into a single data source, first define each data source in the repository and then create a virtual data source that combines them.
- Know the elements of your design: tables and columns to choose, joins to perform, filters to define, sets you need, and item properties to expose to users.
- Determine whether the users of the Domain need localized resources, such as translated labels and local formats.
- Finally, determine a security policy for the data retrieved through the Domain.

To launch the Domain Designer:

1. As an administrative user, select **Add Resource > Domain** in the repository to open the Add New Domain page of the Domain Designer.
2. Enter a name and optional description for the Domain.
3. Under Data Source, browse to locate a data source.
4. Optionally define localization bundles or data-level security on the Add New Domain page of the Domain Designer.
5. Click Create with Domain Designer to launch the Domain Designer.

The next two sections include detailed examples of creating Domains.

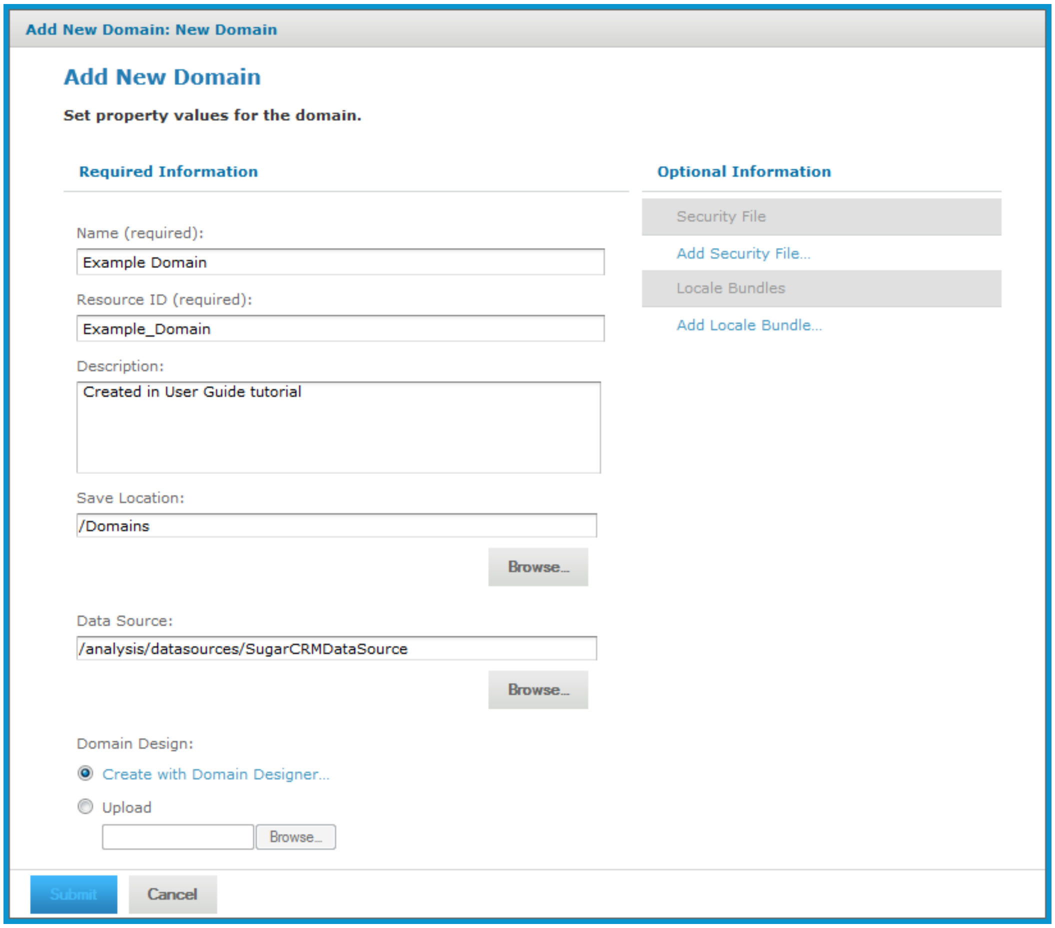
6.3 Example of Creating a Domain

The following example shows how to create a Domain. For complete information about all of the options and settings available for designing reports, see **"Using the Add New Domain Page" on page 220** and **"Using the Domain Designer" on page 223**.

To create a sample Domain:

1. Log into JasperReports Server as an administrator and select **Create > Domain.**

   Alternatively, you can select **View > Repository**, then right-click a folder, such as /Domains, and select **Add Resource > Domain** from the context menu. The Add New Domain page appears.



Add New Domain: New Domain

Add New Domain

Set property values for the domain.

Required Information

Name (required): Example Domain

Resource ID (required): Example_Domain

Description: Created in User Guide tutorial

Save Location: /Domains

Data Source: /analysis/datasources/SugarCRMDataSource

Domain Design:

● Create with Domain Designer...

○ Upload

Optional Information

Figure 6-1 **Add New Domain Page**

2. Enter a name for the Domain and an optional description. In this example:
   - Name – `Example Domain`
   - Resource ID – Accept the automatic ID or change it (no spaces).
   - Description – `Created in User Guide tutorial`
3. Chose the Save Location – Browse to the Domains folder.
4. Choose the Data Source – Browse to select data source and select **SugarCRM Data Source** as shown below.

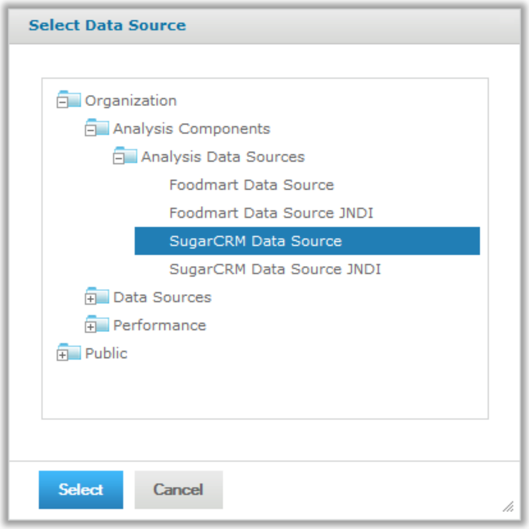


Figure 6-2 Select a Data Source for a New Domain

5. Click the **Create with Domain Designer** link.
6. Select the schema – For data sources that allow schemas (subdivisions of a database), you are prompted to select a schema. The PostgreSQL database that is bundled in the evaluation version supports schemas, so you must select the public schema, as shown in the following figure:

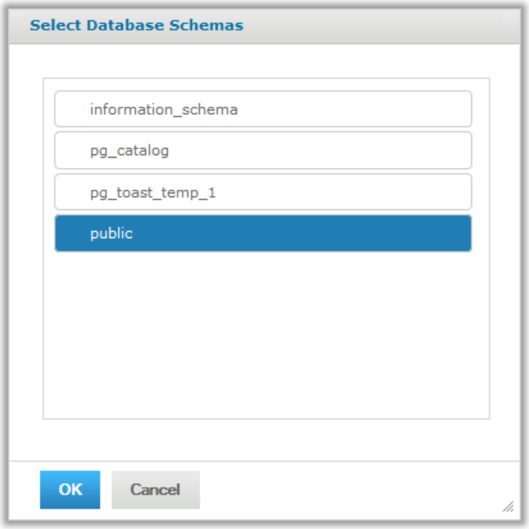


Figure 6-3 Select a Schema for a New Domain

7. Click **OK** in the Select Database Schemas dialog if necessary. The Domain Designer appears.

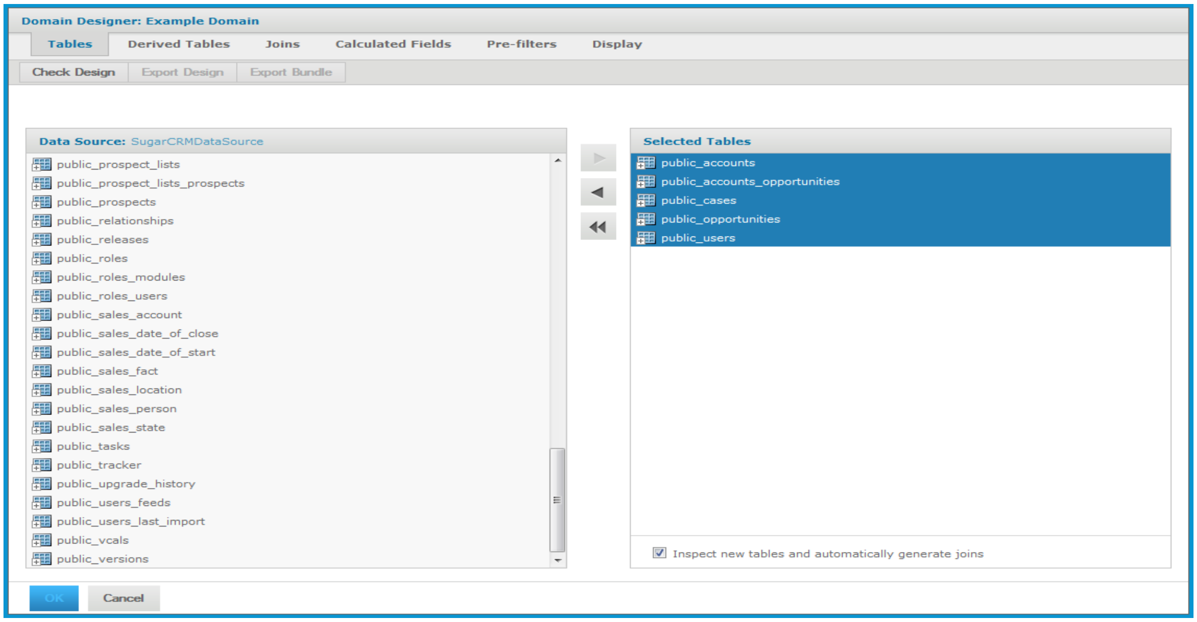
   The **Tables** tab shows the database tables in the data source. You can expand each table to see its individual columns.

In databases that support schemas (subdivisions of a database), table names are prefixed by <schemaName>_ to distinguish tables that may have the same name in separate schemas. This procedure is based on the sample data in a PostgreSQL database, and therefore it uses table names with the public_ prefix.

8. Double-click the following tables in **Data Source** to add them to **Selected Tables**:

   `public_accounts`, `public_accounts_opportunities`, `public_cases`, `public_opportunities`, `public_users`

   Because this example uses the Sugar CRM data source, ignore the check box **Inspect new tables and automatically generate joins**.



Data Source: SugarCRMDataSource

public_prospect_lists
public_prospect_lists_prospects
public_prospects
public_relationships
public_releases
public_roles
public_roles_modules
public_roles_users
public_sales_account
public_sales_date_of_close
public_sales_date_of_start
public_sales_fact
public_sales_location
public_sales_person
public_sales_state
public_tasks
public_tracker
public_upgrade_history
public_users_feeds
public_users_last_import
public_vcals
public_versions

Selected Tables

public_accounts
public_accounts_opportunities
public_cases
public_opportunities
public_users

☑ Inspect new tables and automatically generate joins

Figure 6-4 Tables Tab of the Domain Designer

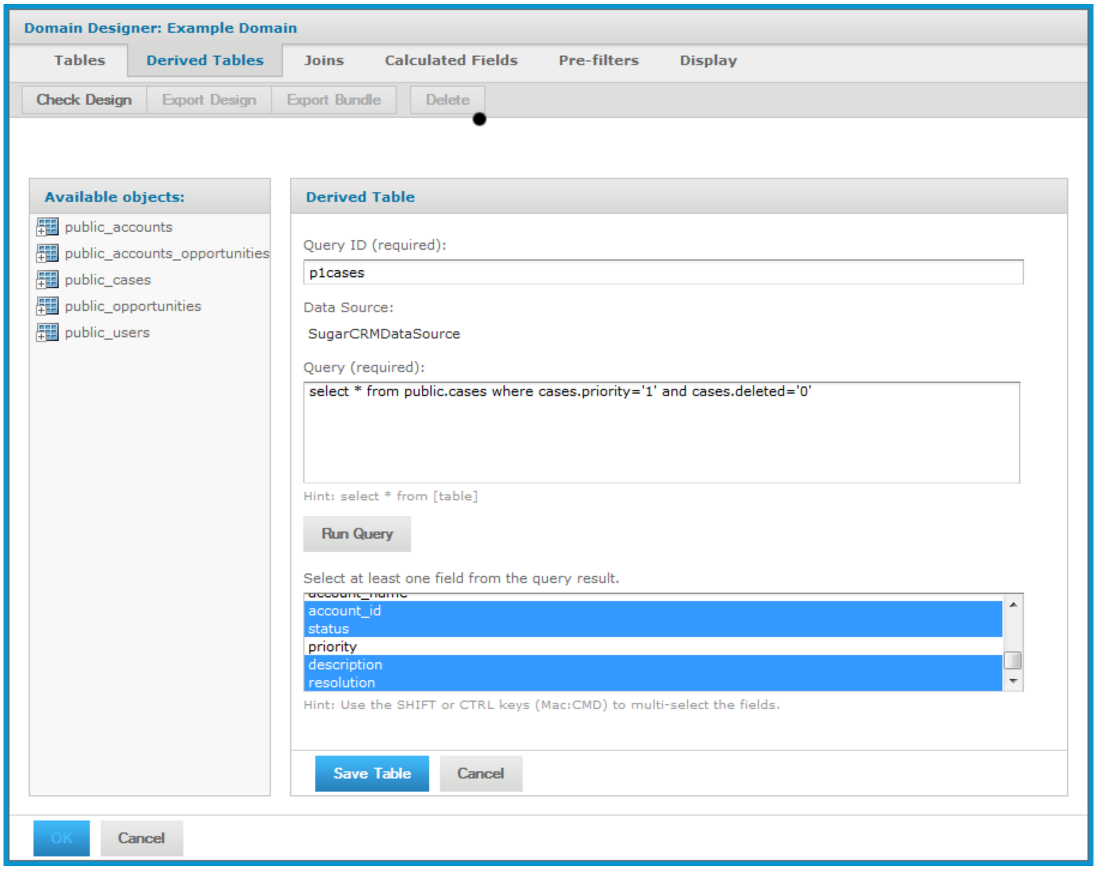
9. Click the **Derived Tables** tab. A derived table is defined by a query and a selection of the columns in the result.
10. Create a derived table:
    a. Enter a meaningful name in the **Query ID** field, in this example: `p1cases`.
    b. Type the following query:

    ```
    select * from public.cases where cases.priority='1' and cases.deleted='0'
    ```

    c. Click **Run Query**.
    d. Ctrl-click to select the following fields from the query result:

id	assigned_user_id	status
case_number	name	description
date_entered	account_id	resolution

The **Derived Tables** tab appears as shown in the following figure.



Query ID (required):

p1cases

Data Source:

SugarCRMDataSource

Query (required):

```
select * from public.cases where cases.priority='1' and cases.deleted='0'
```

Hint: select * from [table]

Select at least one field from the query result.

account_id
status
priority
description
resolution

Hint: Use the SHIFT or CTRL keys (Mac:CMD) to multi-select the fields.

Figure 6-5 Derived Tables Tab of the Domain Designer

e. Click **Save Table**. The new derived table, p1cases, appears in the list of available objects.

11. Click the **Joins** tab. The tables selected on the Tables tab and defined on the Derived Tables tab appear in **Left Table** and **Right Table**.

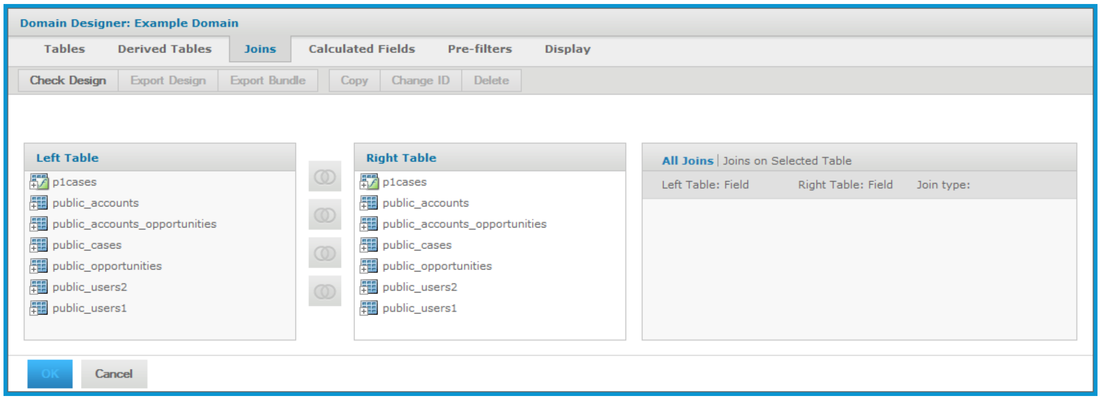


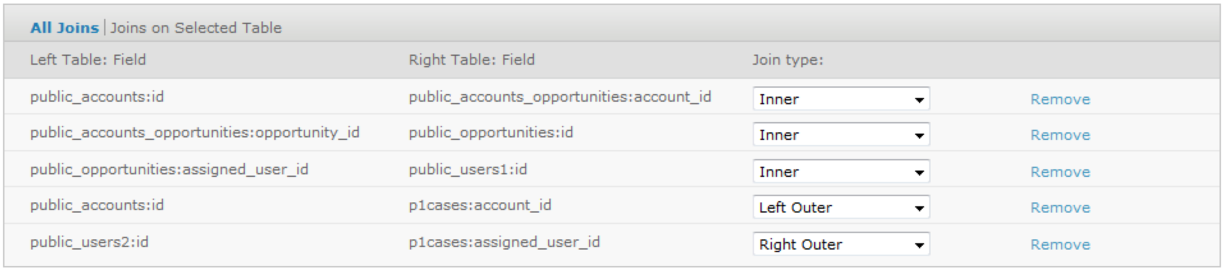
Figure 6-6 Joins Tab of the Domain Designer

12. In the left table, select the `public_users` table and click **Copy** to create the `public_users1` table. Select the same `public_users` table, click **Change ID**, and rename the `public_users` table to `public_users2`. Now there are two table aliases for the users table to avoid circular joins:
13. To specify a join, expand the tables to see column names, select a column in the left table and a column in the right table, then click a join icon. For this example, specify the following joins:

Left Table and Column	Right Table and Column	Join Type
public_accounts: id	public_accounts_opportunities: account_id	Inner
public_accounts_opportunities: opportunity_id	public_opportunities: id	Inner
public_opportunities: assigned_user_id	public_users1: id	Inner
public_accounts: id	P1cases: account_id	Left Outer
public_users2: id	P1cases: assigned_user_id	Right Outer

Each row returned by an inner join contains data from all tables involved in the join. Outer joins return rows that satisfy the join condition plus rows from either the left or right table for which no corresponding rows exist in the other table. In this example, the result of the left outer join includes accounts without P1 cases. The result of the right outer join includes P1 cases without assigned users.

The final list of joins appears as shown below:



All Joins | Joins on Selected Table

Left Table: Field	Right Table: Field	Join type:	
public_accounts:id	public_accounts_opportunities:account_id	Inner	Remove
public_accounts_opportunities:opportunity_id	public_opportunities:id	Inner	Remove
public_opportunities:assigned_user_id	public_users1:id	Inner	Remove
public_accounts:id	p1cases:account_id	Left Outer	Remove
public_users2:id	p1cases:assigned_user_id	Right Outer	Remove

Figure 6-7 Joins for Example Domain

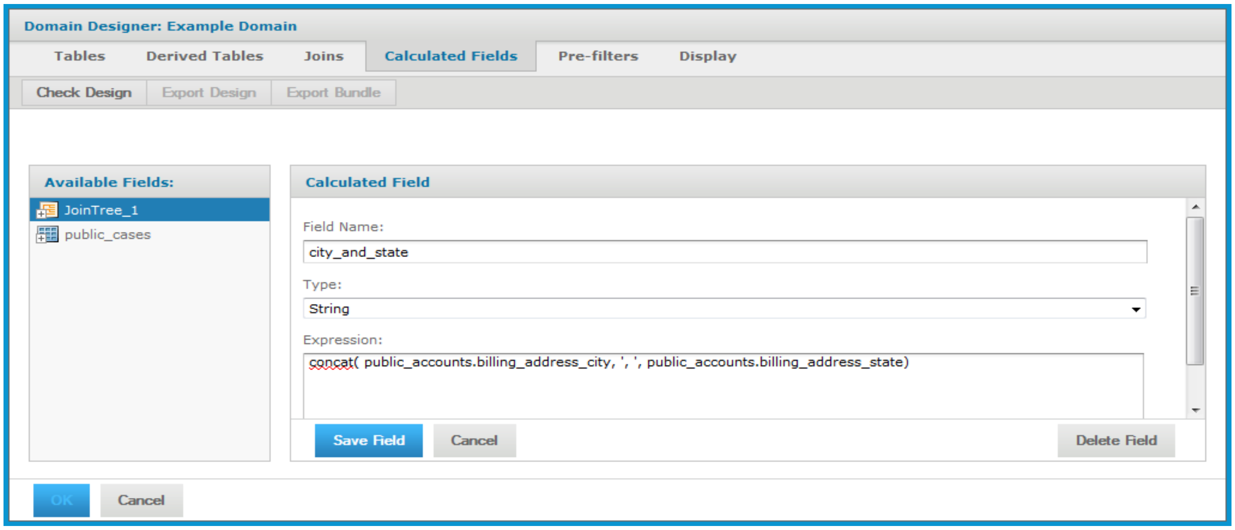
14. Click the **Calculated Fields** tab.

The Available Fields panel shows join trees. These are the joined tables resulting from any joins you defined, and any unjoined tables, such as the `cases` table in this example. The `cases` table was used only to help create the `p1cases` derived table, but was not itself joined to the other tables, as shown in **Figure 6-8**.

15. Enter the following details for a calculated field that creates unambiguous city names:
    - Field Name – `city_and_state`
    - Type – `String`

- Expression – `concat( public_accounts.billing_address_city, ', ', public_accounts.billing_address_state)`

When entering the expression, you can expand the join tree and double-click column names to insert them instead of typing them.



Field Name: city_and_state

Type: String

Expression: concat(public_accounts.billing_address_city, ', ', public_accounts.billing_address_state)

Figure 6-8 Calculated Fields Tab of the Domain Designer

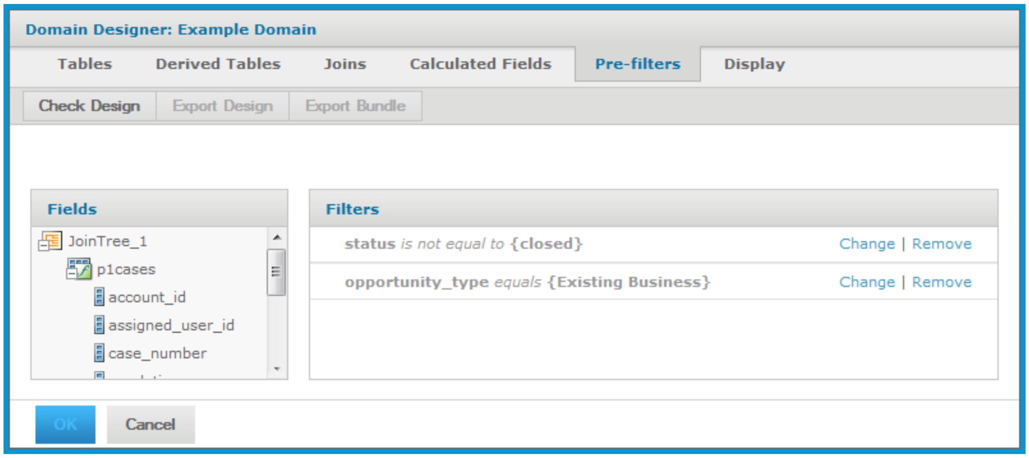
16. Click **Save Field** to validate the expression and add the calculated field to **Available Fields**.
17. Click the **Pre-filters** tab. The Fields and Filters panels appear.
18. Define two filters as follows:
    a. In Fields, expand the join tree and the `public_opportunities` table.
    b. Double-click the `opportunity_type` column to create a filter condition.
    c. Use **Equals** as the comparison operator, and select **Existing Business** from the drop-down.
    d. Click **OK** to create the filter.
    e. Create another filter by expanding the `p1cases` table and double-clicking the `status` column.

    If you select the wrong column, click **Cancel**.

    f. Choose the `is not equal to` comparison operator.
    g. Enter the value `closed` in the search field, but do not click the search icon.

    When the selection list in the Filters pane doesn't contain the value you want, type your value in the search field.

    h. Click **OK** to save the filter. The filters you defined appear in the Filters panel.



Fields:

- JoinTree_1
  - p1cases
    - account_id
    - assigned_user_id
    - case_number

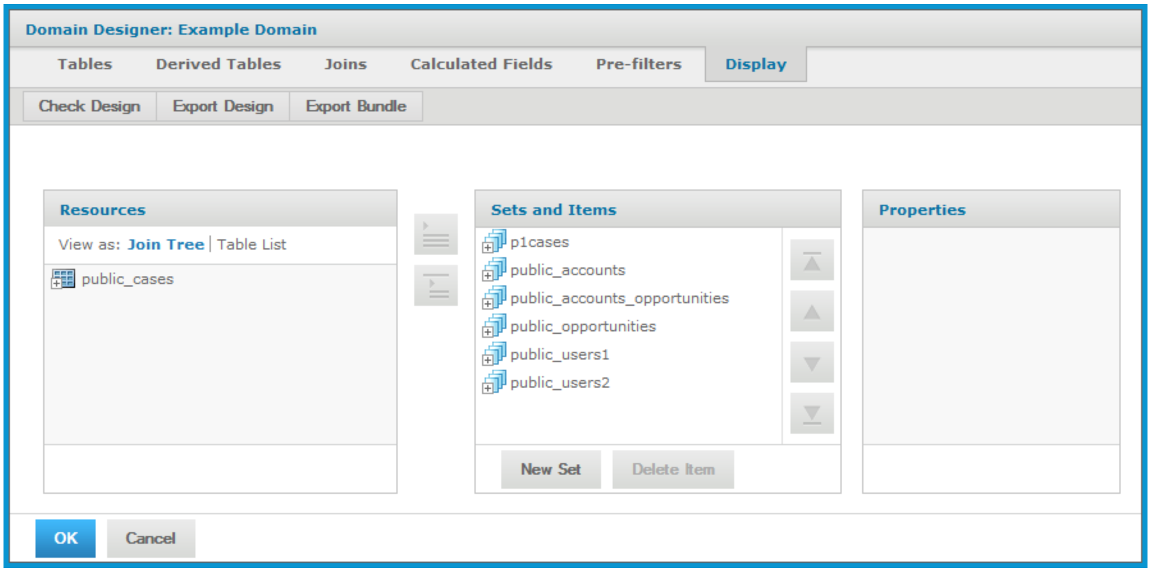
Filters:

status *is not equal to* {closed}

opportunity_type *equals* {Existing Business}

Figure 6-9 Pre-filters Tab of the Domain Designer

19. Click the **Display** tab to specify how your fields should appear to users.



Resources:

- public_cases

Sets and Items:

- p1cases
- public_accounts
- public_accounts_opportunities
- public_opportunities
- public_users1
- public_users2

Figure 6-10 Display Tab of the Domain Designer

20. Create a hierarchy of sets and items from the tables and columns in JoinTree_1:
    a. Select JoinTree_1.
    b. Click ☰ to add all of the JoinTree_1 tables and columns to **Sets and Items**.
21. In the Properties panel, rename the sets and items to give them descriptions specified in the following table.

(Set ID) Set Label Set Description	Item ID	Item Label	Item Description
(public_accounts) Account Customer account information	name1	Customer	Name of customer
	account_type	Type	Account type
	industry	Industry	Primary industry
	annual_ revenue	Revenue Size	Estimated annual revenue
	employees	Employee Size	Estimated number of employees
	city_and_state	City, State	City and state of headquarters
(public_users1) Account Rep Primary account representative	first_name	First Name	Given name
	last_name	Last Name	Surname or family name
(public_opportunities) Opportunity Sales opportunity	date_entered2	Date	Date opportunity opened
	amount	Amount	Anticipated amount of the contract
	probability	Probability	Estimated chance of winning the contract
	description2	Description	Description of the opportunity
	lead_source	Source	Lead source
	sales_stage	Stage	Sales stage
(p1cases) P1 Case High priority support cases	case_number	Case	Case number
	date_entered	Date	Date case opened
	name	Summary	Name or summary of the case
	description	Description	Detailed description of the case
	resolution	Resolution	Description of the case resolution
	status	Status	Current status of the case
(public_users2) Case Rep Support case representative or engineer	first_name1	First Name	Given name
	last_name1	Last Name	Surname or family name

22. Set the data format and summary properties for the following items:
    - Opportunity, Date: data format of Jun 28, 2012
    - Opportunity, Amount: data format of ($1,235) and summary of average
    - P1 Case, Date: data format of Jun 28, 2012

    When used in reports, these items will have the data formats and summary functions defined here as defaults.

    You can also set the Field or Measure setting on any item. By default, numeric fields are set as measures. You may need to change this setting occasionally. For example, a numeric value that you use as an identifier should be set to Field, and a textual ID that you want to use for counting should be set to Measure (and the summary function set to Count All or Count Distinct).

23. Click **OK** to finish creating this Domain. The Add New Domain page appears again.

    Under Domain Design, you can click **Edit with Domain Designer** to launch the Domain Designer again to edit it.

24. Click **Submit** in the Add New Domain page. The new Domain is validated and stored in the location you specified in **step 3**. The new Domain appears in search results when you search the repository for it.

6.4 Example of Creating a Domain Using a Virtual Data Source

The following example shows how to create a Domain based on a virtual data source. The Domain designer interface is very similar to the example in **"Example of Creating a Domain" on page 204**, which covers some of these options in more detail. For complete information about all of the options and settings available for designing reports, see **"Using the Add New Domain Page" on page 220** and **"Using the Domain Designer" on page 223**.

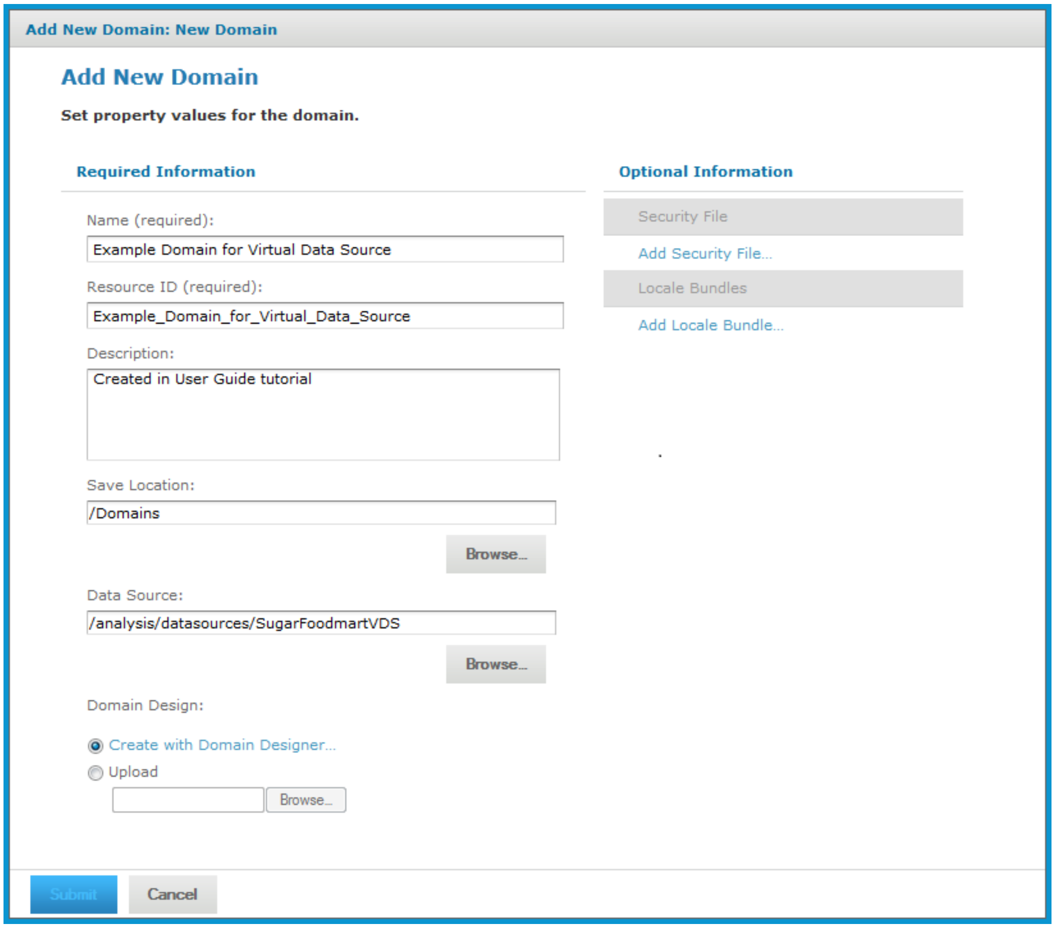
The example in this section uses the SugarFoodMartVDS virtual data source, which combines the Foodmart and SugarCRM JDBC data sources. Normally, you combine data sources that have relationships you want to explore. This example lets you work with SugarCRM sales and Foodmart sales by city, region, or state.

To create a useful Domain using a virtual data source, you need to define relationships that connect the tables of the different data sources. This example shows how to create a Domain that shows the sales by city for each data source. To do this, you create two derived tables, one that aggregates `sales` by city for the Foodmart data source and another that aggregates **amount** by city for the SugarCRM data source. Then you join the two data sources on city to combine them.

To create a sample Domain using a virtual data source:

1. Log in to JasperReports Server as an administrator and select **Create > Domain.**

   Alternatively, you can select **View > Repository** then right-click a folder, such as /Domains, and select **Add Resource > Domain** from the context menu. The Add New Domain page appears.



Add New Domain: New Domain

Add New Domain

Set property values for the domain.

Required Information

Name (required): Example Domain for Virtual Data Source

Resource ID (required): Example_Domain_for_Virtual_Data_Source

Description: Created in User Guide tutorial

Save Location: /Domains

Data Source: /analysis/datasources/SugarFoodmartVDS

Domain Design:

● Create with Domain Designer...

○ Upload

Optional Information

Figure 6-11 **Add New Domain Page Using a Virtual Data Source**

2. Enter a name for the Domain and an optional description. In this example:
   - Name – `Example Domain`
   - Resource ID – Accept the automatic ID or change it (no spaces).
   - Description – `Created in User Guide tutorial`
3. Chose the Save Location – Browse to the Domains folder.
4. Choose the Data Source – Browse to select data source and select **SugarCRM-Foodmart Virtual Data Source** as shown below.

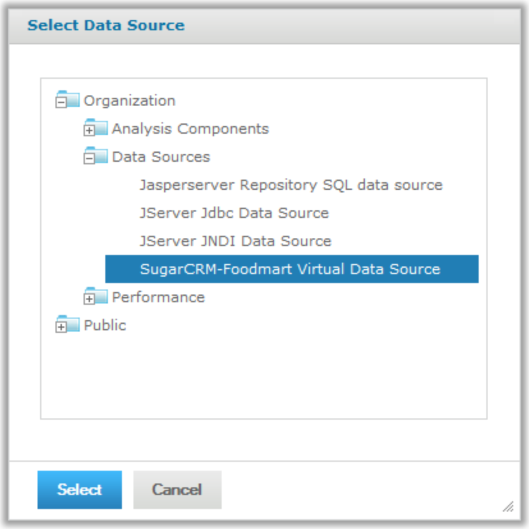
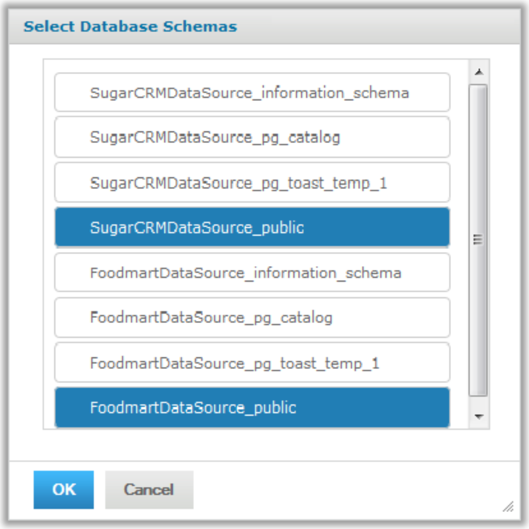


Figure 6-12 Select a Virtual Data Source for a New Domain

The path to the data source appears on the Add New Domain page.

5. Click the **Create with Domain Designer** link.
6. You are prompted to select the database schemas to use. Use Ctrl-Click to select multiple schemas; these can be schemas from different databases or different schemas from the same database. The example in this section uses the schemas as they appear in the PostgreSQL database bundled in the evaluation version. Select the `FoodmartDataSource_public` and `SugarDataSource_public` schemas:



Select Database Schemas
SugarCRMDataSource_information_schema
SugarCRMDataSource_pg_catalog
SugarCRMDataSource_pg_toast_temp_1
SugarCRMDataSource_public
FoodmartDataSource_information_schema
FoodmartDataSource_pg_catalog
FoodmartDataSource_pg_toast_temp_1
FoodmartDataSource_public
OK
Cancel

Figure 6-13 Select Schemas for a New Domain Using a Virtual Data Source

In Domains that use virtual data sources, table and schema names are prefixed by the data source alias to distinguish tables and schemas that have the same name in separate data sources. Aliases are set when the virtual data source is created.

The example in this section uses the schema names as they appear in the PostgreSQL database bundled in the evaluation version.

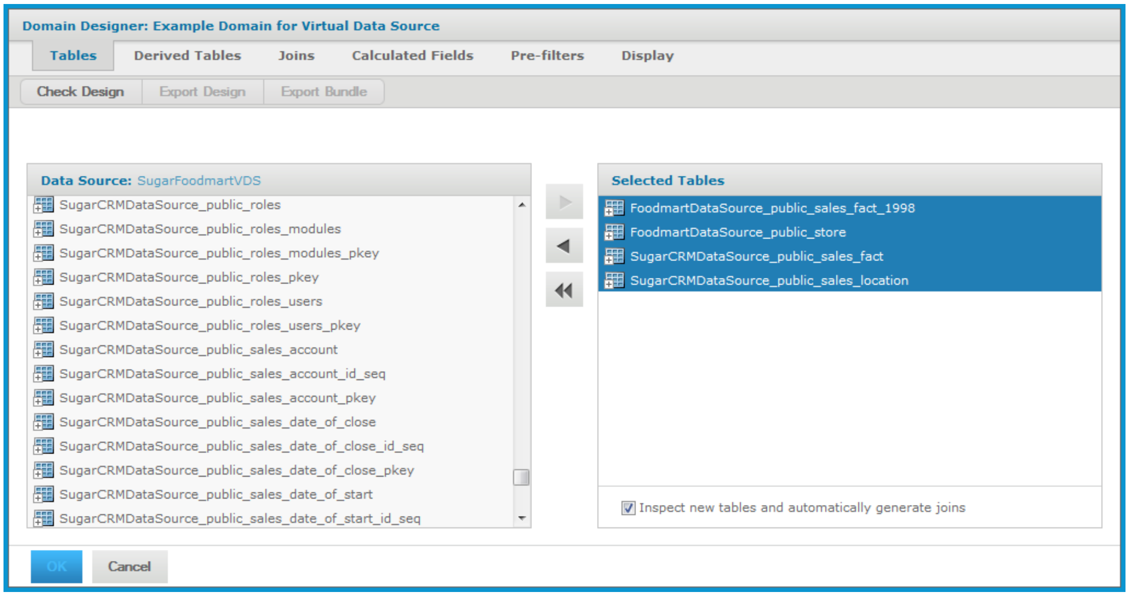
7. Click **OK**. The Domain Designer appears. The **Tables** tab shows the database tables in the data source.

You can create a Domain by joining the two data sources on `City`, `Region`, and `Country`. In this example, a Domain created that way works well for a crosstab, but not so well for a chart or table. This is because the data sources include geographic information at lower level than `City`. To use the data for a chart or table, you need to aggregate the city values for each data source, then join those.

8. Double-click the following tables in **Data Source** to add them to **Selected Tables**:

   `FoodmartDataSource_public_sales_fact_1998`, `FoodmartDataSource_public_store`, `SugarCRMDataSource_public_sales_fact`, `SugarCRMDataSource_public_sales_location`

   Make sure **Inspect new tables and automatically generate joins** is checked, as shown below.



Data Source: SugarFoodmartVDS

- SugarCRMDataSource_public_roles
- SugarCRMDataSource_public_roles_modules
- SugarCRMDataSource_public_roles_modules_pkey
- SugarCRMDataSource_public_roles_pkey
- SugarCRMDataSource_public_roles_users
- SugarCRMDataSource_public_roles_users_pkey
- SugarCRMDataSource_public_sales_account
- SugarCRMDataSource_public_sales_account_id_seq
- SugarCRMDataSource_public_sales_account_pkey
- SugarCRMDataSource_public_sales_date_of_close
- SugarCRMDataSource_public_sales_date_of_close_id_seq
- SugarCRMDataSource_public_sales_date_of_close_pkey
- SugarCRMDataSource_public_sales_date_of_start
- SugarCRMDataSource_public_sales_date_of_start_id_seq

Selected Tables

- FoodmartDataSource_public_sales_fact_1998
- FoodmartDataSource_public_store
- SugarCRMDataSource_public_sales_fact
- SugarCRMDataSource_public_sales_location

☑ Inspect new tables and automatically generate joins

Figure 6-14 Tables Tab of the Domain Designer Using a Virtual Data Source

9. Click the **Derived Tables** tab.
10. Create the following derived table to aggregate `store_sales` for the Foodmart data source by city:
    a. Enter a meaningful name in the **Query ID** field, in this example: `FoodmartSalesByCity`.
    b. Type the following query to aggregate the store sales by city:

```
select store_city FoodCity, sum(store_sales) FoodCitySales from
FoodmartDataSource_public.store join FoodmartDataSource_public.sales_fact_1998 on
(FoodmartDataSource_public.sales_fact_1998.store_id = FoodmartDataSource_
public.store.store_id) group by store_city
```

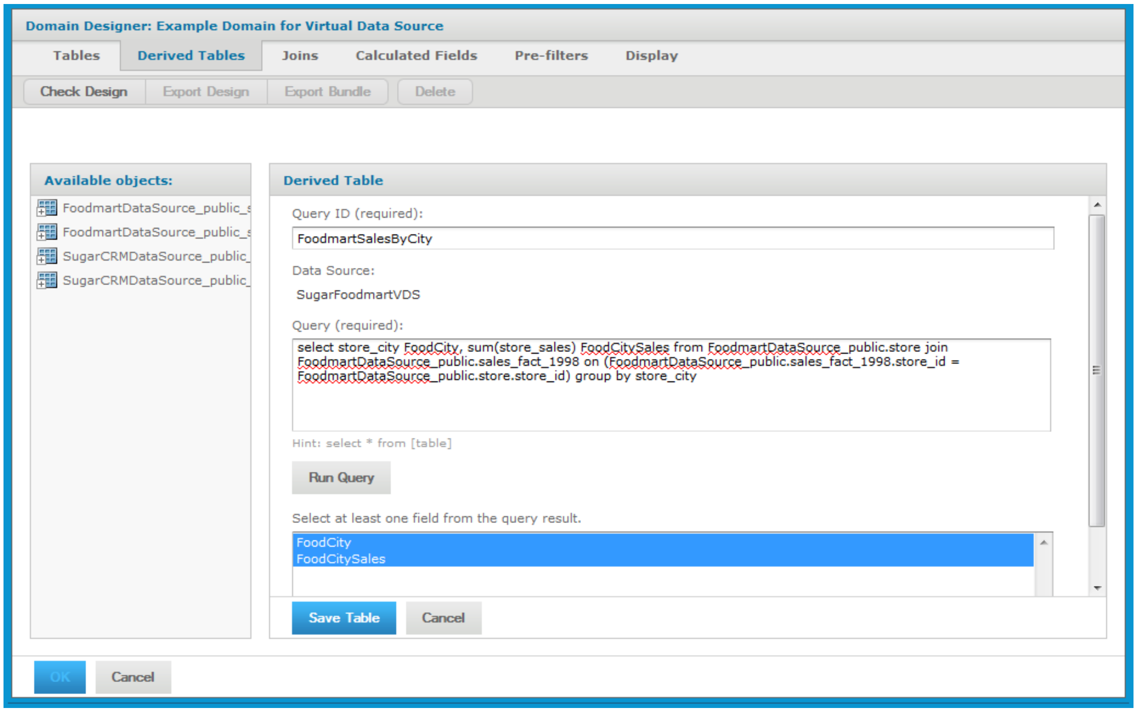
SQL queries for a derived table must be valid with respect to the JDBC driver for the data source. If you are working with a virtual data source, SQL queries are validated against Teiid SQL, which provides DML SQL-92 support with select SQL-99 features. For more information, see the Teiid documentation on the Jaspersoft Support Portal.

c. Click **Run Query**.

d. The fields from the query result are automatically highlighted:

   `FoodCity, FoodCitySales`

   The Derived Tables tab appears as shown in the following figure.


Domain Designer: Example Domain for Virtual Data Source
Tables
Derived Tables
Joins
Calculated Fields
Pre-filters
Display
Check Design
Export Design
Export Bundle
Delete
Available objects:
FoodmartDataSource_public_s
FoodmartDataSource_public_s
SugarCRMDataSource_public_
SugarCRMDataSource_public_
Derived Table

Query ID (required): FoodmartSalesByCity

Data Source: SugarFoodmartVDS

Query (required):

```
select store_city FoodCity, sum(store_sales) FoodCitySales from FoodmartDataSource_public.store join
FoodmartDataSource_public.sales_fact_1998 on (FoodmartDataSource_public.sales_fact_1998.store_id =
FoodmartDataSource_public.store.store_id) group by store_city
```

Hint: select * from [table]

Select at least one field from the query result.

FoodCity
FoodCitySales

Figure 6-15 Derived Tables Tab of the Domain Designer Using a Virtual Data Source

e. Click **Save Table**. The derived table, FoodmartSalesByCity, appears in the list of available objects.

11. Create a second derived table to aggregate the amount for the Sugar data source by city:

    a. Enter a meaningful name in the **Query ID** field, in this example: `SugarAmountByCity`.

    b. Type a query to aggregate the amount by city. If your data source supports database schemas, use the following query:

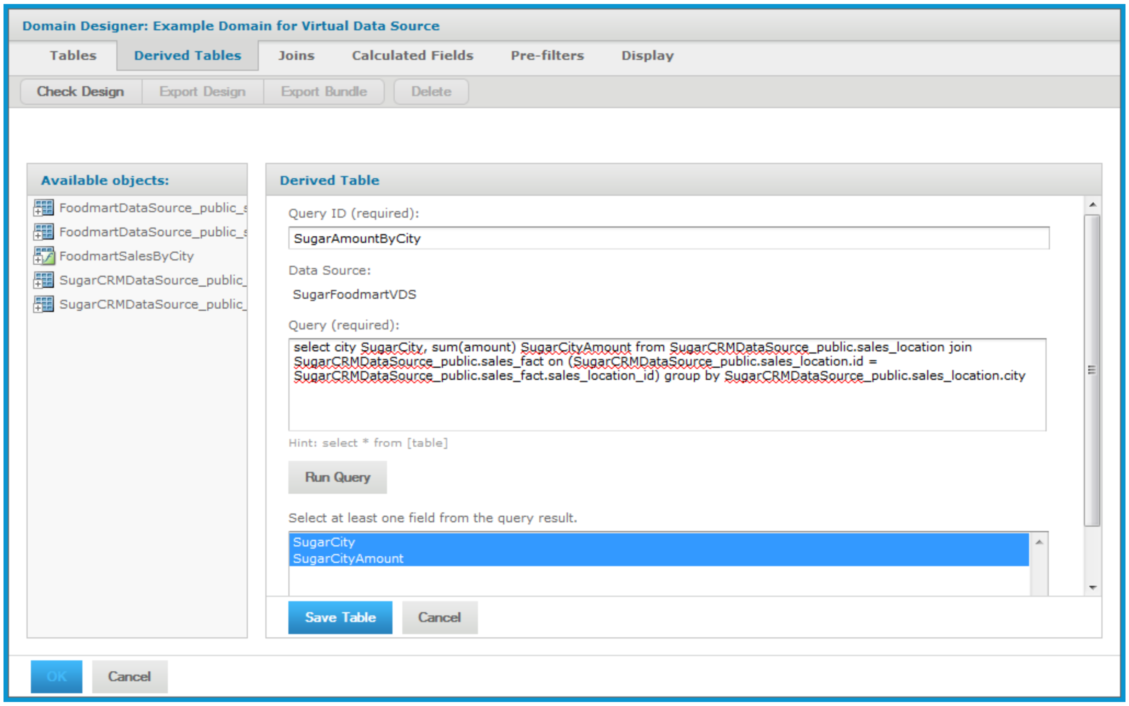
```
select city SugarCity, sum(amount) SugarCityAmount from SugarCRMDataSource_
public.sales_location join SugarCRMDataSource_public.sales_fact on
(SugarCRMDataSource_public.sales_location.id = SugarCRMDataSource_public.sales_
fact.sales_location_id) group by SugarCRMDataSource_public.sales_location.city
```

    c. Click **Run Query**.

    d. The following fields are highlighted in the query result:

       `SugarCity, SugarCityAmount`

The Derived Tables tab appears as shown in the following figure.



Query ID (required):

SugarAmountByCity

Data Source:

SugarFoodmartVDS

Query (required):

```
select city SugarCity, sum(amount) SugarCityAmount from SugarCRMDataSource_public.sales_location join
SugarCRMDataSource_public.sales_fact on (SugarCRMDataSource_public.sales_location.id =
SugarCRMDataSource_public.sales_fact.sales_location_id) group by SugarCRMDataSource_public.sales_location.city
```

Hint: select * from [table]

Select at least one field from the query result.

SugarCity
SugarCityAmount

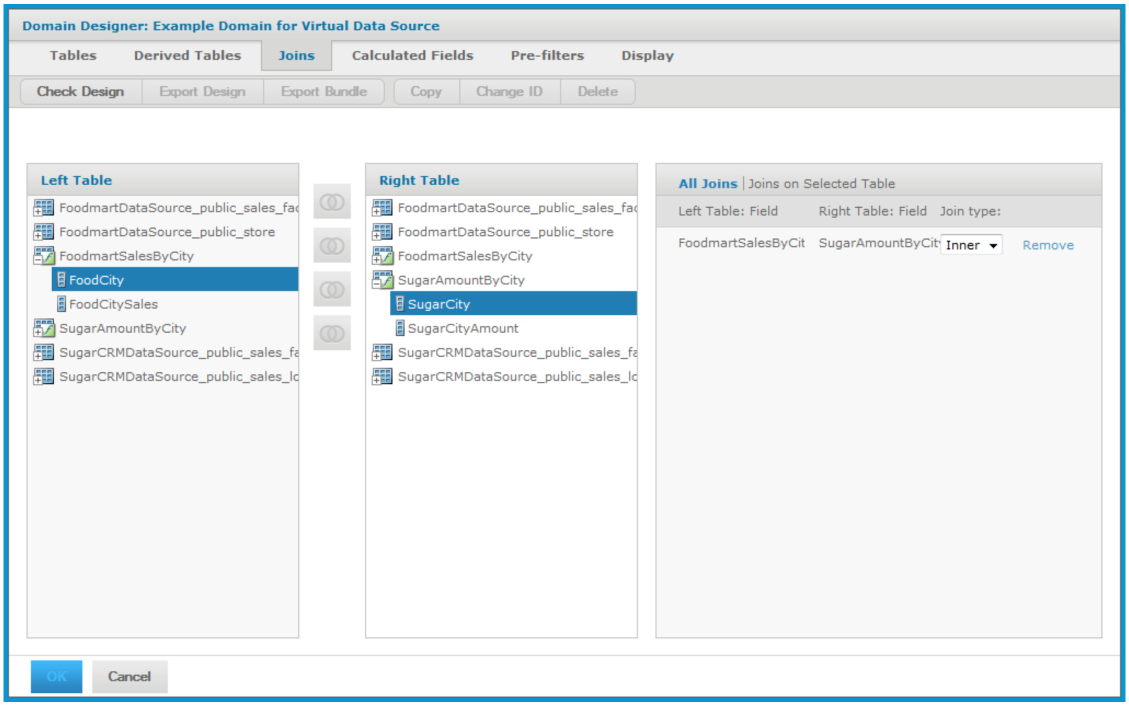
Figure 6-16 Derived Tables Tab of the Domain Designer Using a Virtual Data Source

e. Click **Save Table**. The SugarAmountByCity derived table appears in the list of available objects.

You can also use the Derived Tables tab to combine two tables from different databases or schemas that have identical columns. For example, if you have a virtual data source where Table1 in DataSource1 and Table2 in DataSource2 have identical columns, you can create a derived table that combines these two tables using the following syntax:

```
select * from DataSource1.Table 1 UNION ALL select * from
DataSource2.Table2
```

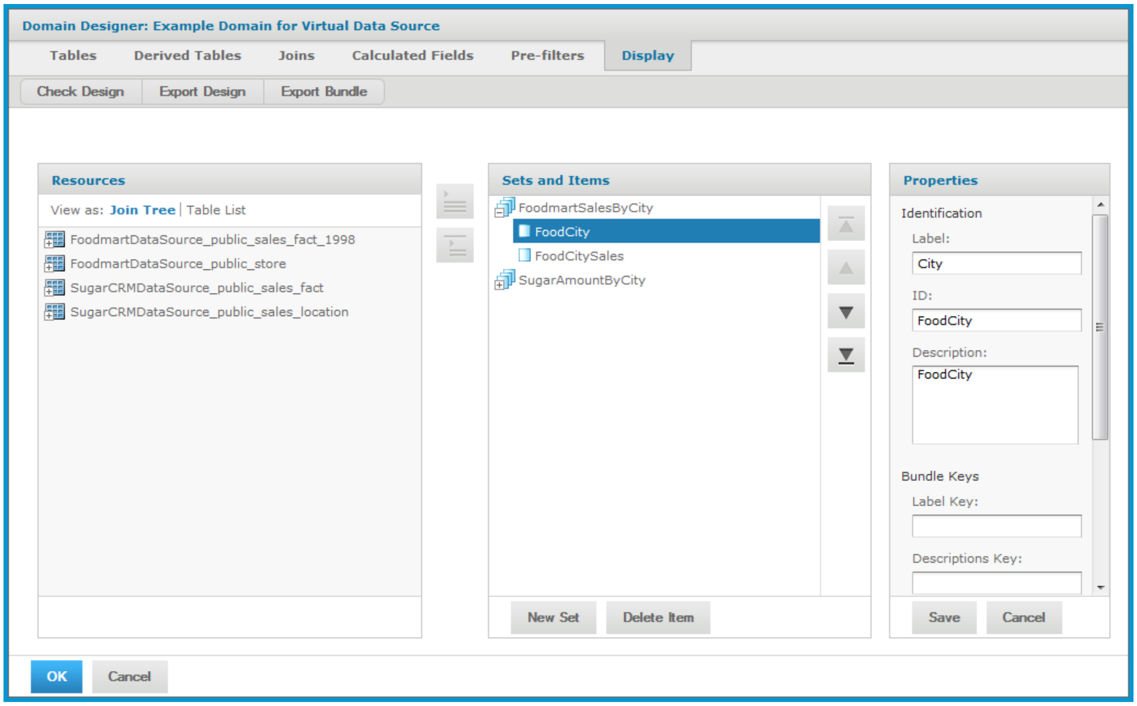
12. Click the **Joins** tab. The tables selected on the Tables tab and defined on the Derived Tables tab appear in **Left Table** and **Right Table**.
13. In the left table, expand the `FoodmartSalesByCity` table and select the `FoodCity` column.
14. In the right table, expand the `SugarAmountsByCity` table and select the `SugarCity` column.
15. Click the Inner icon to specify an inner join.


Domain Designer: Example Domain for Virtual Data Source
Tables
Derived Tables
Joins
Calculated Fields
Pre-filters
Display
Check Design
Export Design
Export Bundle
Copy
Change ID
Delete
Left Table
FoodmartDataSource_public_sales_fac
FoodmartDataSource_public_store
FoodmartSalesByCity
FoodCity
FoodCitySales
SugarAmountByCity
SugarCRMDataSource_public_sales_fa
SugarCRMDataSource_public_sales_lo
Right Table
FoodmartDataSource_public_sales_fac
FoodmartDataSource_public_store
FoodmartSalesByCity
SugarAmountByCity
SugarCity
SugarCityAmount
SugarCRMDataSource_public_sales_fa
SugarCRMDataSource_public_sales_lo
All Joins | Joins on Selected Table
Left Table: Field
Right Table: Field
Join type:
Remove
OK
Cancel

Left Table: Field	Right Table: Field	Join type:
FoodmartSalesByCit	SugarAmountByCit	Inner

Figure 6-17 Joins Tab of the Domain Designer Using a Virtual Data Source

16. Click the **Display** tab. Here you can specify the way your fields will appear to users.



Label: City

ID: FoodCity

Description: FoodCity

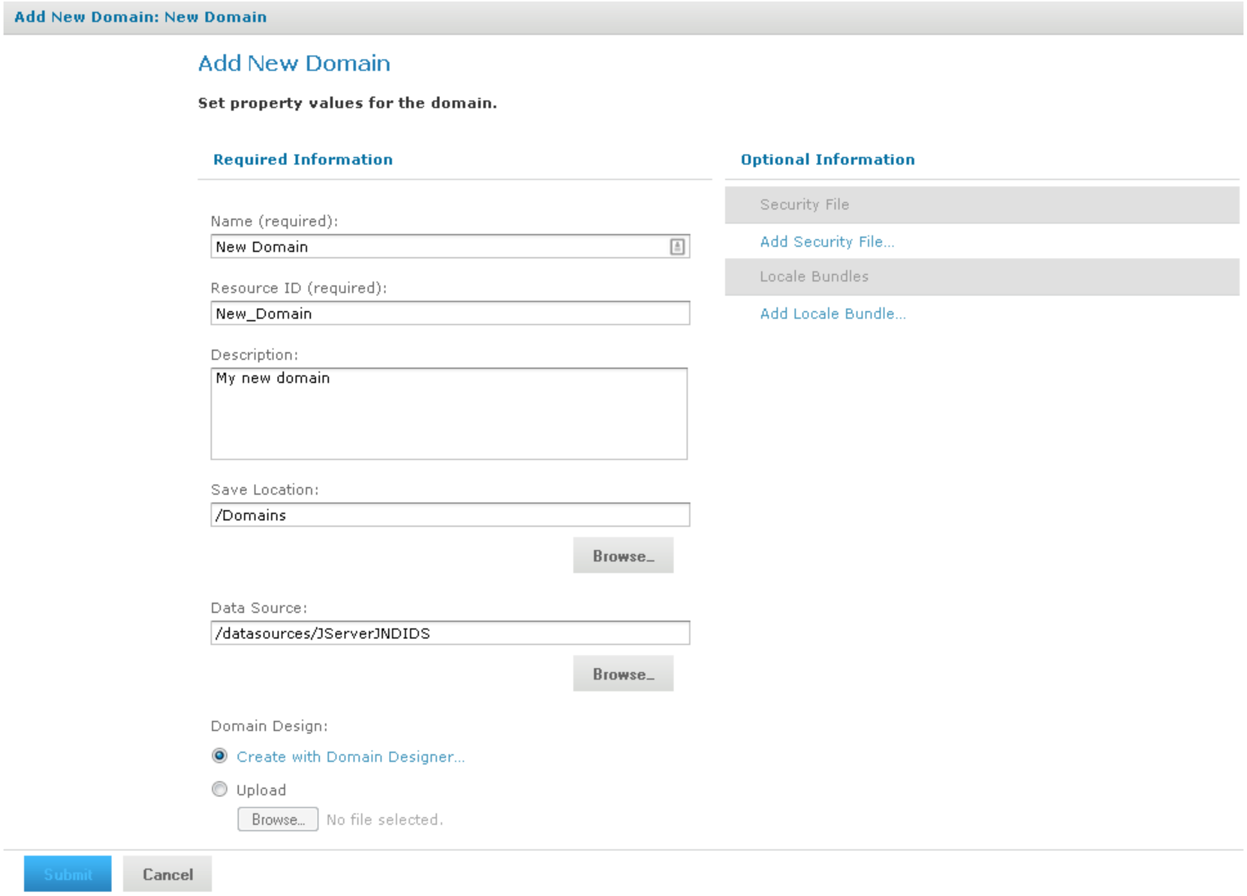
Figure 6-18 Display Tab of the Domain Designer Using a Virtual Data Source

17. Create a hierarchy of sets and items from the tables and columns in JoinTree_1:
    a. Expand JoinTree_1.
    b. Select FoodmartSalesByCity and click ▸☰ to add it to **Sets and Items**.
    c. Select SugarAmountsByCity and click ▸☰ to add it to **Sets and Items**.
18. Select FoodCity under FoodMartSalesByCity in the **Sets and Items** panel. Click **Edit** in the **Properties** panel and change the Label to City. Click **Save**.
19. Click **OK** to finish creating this Domain. The Add New Domain page appears again.
20. Click **Submit**. The new Domain is validated and stored in the location you specified in **step 3**.

    The new Domain appears in search results when you search the repository for it. You can use this Domain to create Ad Hoc views based on this combined information.

6.5 Using the Add New Domain Page

The Add New Domain page appears when you click **Create** > **Domain** or when you right-click a repository folder and choose **Add Resource** > **Domain**. On the Add New Domain page, you set up basic properties of the Domain and launch the Domain Designer.



Add New Domain: New Domain

Add New Domain

Set property values for the domain.

Required Information

Name (required): New Domain

Resource ID (required): New_Domain

Description: My new domain

Save Location: /Domains

Data Source: /datasources/JServerJNDIDS

Domain Design:

● Create with Domain Designer...

○ Upload

No file selected.

Figure 6-19 Add New Domain Page

Enter basic Domain properties:

- **Name** – Enter the domain name users will see in the repository.
- **Resource ID** – Accept the automatic resource ID or enter a new one (no spaces).
- **Description** – Enter a description. This will appear in the repository and in the Ad Hoc Editor when this Domain is used to create a report. Optional, but recommended.
- **Save Location** – Browse to the repository folder where you want to save the Domain.

The server locates Domains by their repository object type, not by their location. The default location is the last folder selected in the repository.

- **Data Source** – Browse the repository to choose a data source.
  - Use only data sources in the repository and for which the user has at least execute permission.
  - Domains must use a JDBC, JNDI, or virtual data source.
  - If you installed the sample data, you can use the sample JDBC, JNDI, and virtual data sources in **Organization > Data Sources** and **Organization > Analysis Components > Analysis Data Sources**.
- **Domain Design** – Create the design using the Domain Designer console or upload a Domain design file from outside the repository. See **“Advanced Domain Features” on page 245** for more information about Domain design files.

- **Optional Information** – Add a security file and one or more locale bundles.

Before you can submit and save a Domain to the repository, you must specify a display name, resource ID, save location, data source, and Domain design. The **Submit** button is grayed out until you have created or specified the design for your domain.

- The **Cancel** button exits the Add New Domain page. For new Domains, no Domain is created; when modifying an existing Domain, it remains unchanged.
- The **Submit** button validates and saves the Domain components. For more information, see **“Using the Domain Designer” on page 223**. The location for Domains in the sample data is the **Organization > Domains** folder, but you may use any location.

6.5.1 Add Security File and Add Locale Bundle Options

The **Add New Domain** page includes options to upload security and locale bundle files for the Domain. You can add, replace, or remove any previously uploaded files for the Domain.

Figure 6-20 Resources Page of the Add New Domain Dialog

Adding security and local bundle files is documented in **“Security and Locale Information for a Domain” on page 274**.

6.5.2 Selecting a Schema

If your data source supports database schemas, like Oracle RDBMS, you need to specify which schema to use. You also need to select schemas when using a virtual data source. The Select Database Schemas dialog box appears when your database supports schemas and you click **Create with Domain Designer**.

Figure 6-21 Select Database Schemas for Regular and Virtual Data Sources

If you're using a JDBC or JNDI data source, select a single schema from the available schemas in the data source. If you're using a virtual data source, Ctrl-click to select multiple schemas; these can be from the same data source or different data sources.

All the tables in the schemas you choose appear in the Data Source panel on the Tables tab. For more information, see **"The Domain Design File" on page 245**.

6.6 Using the Domain Designer

You use the Domain Designer to define all the components of a Domain. On the Add New Domain page click **Create in Domain Designer**. On the Edit Domain page click **Edit in Domain Designer**. Use the tabs at the top of the Domain Designer to configure various aspects of the design:

- Tables tab – Select all tables whose columns you want to use in the Domain, either directly or indirectly.
- Derived Tables tab – Enter queries whose results appear as derived tables available in the Domain.
- Joins tab – Define inner and outer joins between all the tables and derived tables.
- Calculated Fields tab – Enter expressions whose results appear as calculated fields.
- Pre-filters tab – Specify conditions on field values to limit the data accessed through the Domain.
- Display tab – Organize the visual aspects of the Domain and change the display properties of tables, columns, sets, and items exposed to Domain users.

From the Tables tab, you can navigate to any other tab. To navigate among the tabs, click the tab name at the top of the Domain Designer. Before you can save the design, you must choose which sets and items to expose to users of the Domain. The **OK** button on the console validates the design and saves it in the location you specified earlier. For more information, see **"Domain Validation" on page 235**. After saving a Domain, you can modify it using the Domain Designer.

6.6.1 Tables Tab

The Tables tab contains two panels:

- Data Source – Shows the tables and columns in the data source you chose for the Domain.
- Selected Tables – Shows the tables and columns you use to design the Domain. Initially, this panel is blank.

Typically, you move the following tables from Data Source to Selected Tables:

- Tables that need to be joined
- Tables that you want to reference in the Domain design, even if their columns do not appear directly in the Domain.
  For example, select the tables containing columns that you want to use in a derived table or calculated field.

An understanding of the logical design of tables in the data source is critical to selecting the tables to be joined.

To select tables for use in designing the Domain, first expand the table icons beside the table names to inspect the columns of a table. Double-click or drag a table name from the **Data Source** panel to the **Selected Tables** panel. Alternatively, use ▶ or ◀ to move a table from one panel to the other.

On the Tables tab, you can select only entire tables for the Domain. On the Display tab, you can make column-level selections.

To remove a table, double-click or drag it out of Selected Tables. You can also click ◀◀ to clear Selected Tables.

The **Inspect new tables and automatically generate joins** check box at the bottom of Selected Tables creates joins only if the database has been configured with referential constraints (foreign keys). Otherwise, selecting it has no effect.

If applicable, the generated joins appear on the Join tab.

The Tables tab does not detect changes to the database tables and columns in real-time. To update a Domain after making changes to the database structure, click **OK** to close the Domain Designer, then launch it again.

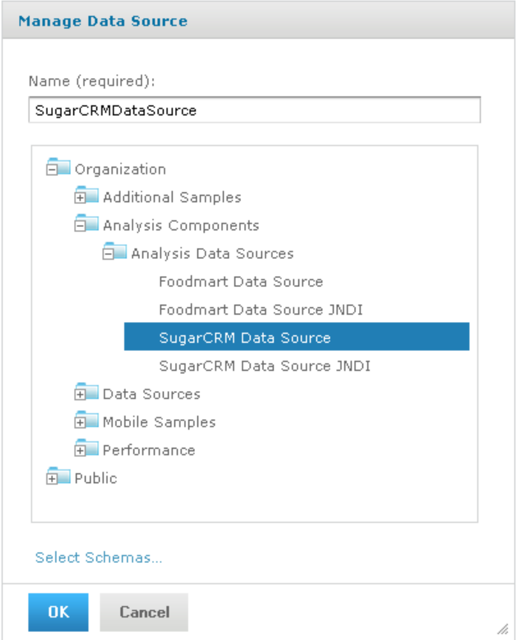
Keep the following points in mind regarding Domain updates:

- New tables and columns appear in the Data Source panel; new columns appear under their table name.
- To add a new column to the Domain, move its table to Selected Tables. The Tables tab works only with entire tables.
- On Selected Tables, respond affirmatively to the prompt to remove tables and columns deleted from the database. The Domain Designer removes those columns or tables from the Tables tab.

If you had selected dropped columns for display, you must manually remove them from the Display tab, otherwise the Domain issues a validation error. For information about removing columns that were displayed through the Domain, see **"Maintaining Referential Integrity" on page 237**.

6.6.2 Manage Data Source Dialog Box

The name of the data source that you choose for the Domain appears at the top of the **Data Source** panel on the Tables tab. Click the name of the data source to make changes to it. The **Manage Data Source** dialog appears.



Manage Data Source

Name (required): SugarCRMDataSource

Figure 6-22 Manage Data Source Dialog Box of the Tables Tab

The name of the data source defaults to the name of its repository object. Domain users don't see the data source name. You can change it as needed. However, if you select a new data source for the Domain, the name in the design remains unchanged. In this case, you may want to rename the data source in the design to match the new data source.

If the database changes servers, you need to create a new data source object and use it to replace the previous one in the Domain. To change the data source, select a new one and click **OK**.

When you change the data source, settings in the Domain Designer that don't conform to the new data source are lost without prompting. Before you change the data source of a Domain, copy the Domain as described in **"Copying a Domain" on page 242**. For more information about switching the data source, see **"Switching the Data Source of a Domain" on page 281**.

If the database supports schemas such as PostgreSQL, Oracle RDBMS, or virtual data sources, click **Select Schemas** at the bottom of the Manage Data Source dialog box to choose among available schemas in the data source. The tables in the schemas that you choose appear in the Data Sources panel.

The Data Source panel shows columns that have a supported type listed in **Structure of the Design File**. If the data source has special datatypes like CLOB or NVARCHAR2, or if you access synonyms on an Oracle database, you need to configure the server to recognize them. See the configuration chapter in the *JasperReports Server Administrator Guide*.

6.6.3 Derived Tables Tab

You create a derived table in a Domain by first building a custom query using the objects you selected on the Tables tab. Because Domains are based on JDBC and JNDI data sources, you write the query in SQL. You run the query, and then select columns from the query result for use in the Domain design.

The clauses in a query determine the structure and content of the table returned by the query. For example, the WHERE clause may contain conditions that determine the rows of the derived table.

To define a derived table:

1. Type a name for the table in the **Query ID** field.
2. Enter a valid SQL query in **Query**. The query may refer to any table or column in **Available objects**, as long as their tables are properly joined in the query as well. Only queries that begin with the `select` statement are allowed. Do *not* include a closing semi-colon (`;`).

   Expand the tables in **Available objects**. Double-click column names to add them to the query.

3. When the query is complete, click **Run Query** to test it.
4. By default, all columns in the result are selected. Ctrl-click to change the selection. If you want only a few columns out of many, it is easier to specify the column names in the `SELECT` clause of the query.
5. Click **Save Table** to add the derived table to **Available objects**. A distinctive icon identifies it as a derived table.

6.6.4 Joins Tab

Joins are associations between tables enabling their rows may be presented together in the same report. Multiple joins associate columns across many tables to create powerful data visualizations when used in reports. The number of tables and joins in the Domain depends on your business needs, as described in **"Domain Use Cases" on page 202**. The Domain Designer supports the four most common join types, all based on equality of values between column.

You can create advanced joins using XML in the design file. Advanced joins are not supported in the Domain Designer. If you have a Domain that uses advanced joins, and you open it in the Domain Designer and then save it, your advanced joins will be lost or changed.

Both **Left Table** and **Right Table** panels display the list of selected and derived tables. Expand tables in both panels, select a column in the left table and one in the right table with the same logical meaning and compatible formats, then select a join:

Join Inner – The result contains only rows where the values in the chosen columns are equal. In the support case example in **"Example of Creating a Domain" on page 204**, the result of an inner join contains only support cases that have been assigned to a support engineer.

Join Left Outer – The result contains all the rows of the **Left Table**, paired with a row of the **Right Table** when the values in the chosen columns are equal or contain blanks. If the support cases are in the Left Table of the example, the result of a left outer join contains all support cases even if they do not have an assigned engineer.

Join Right Outer – The result contains all the rows of the Right Table, paired with a row of the Left Table when the values in the chosen columns are equal or contain blanks. If the users are in the Right Table of the example, the result of a right outer join contains all the users and the support cases assigned to each engineer, if there are any. In this example, a user might also appear several times if different support cases refer to the same support engineer user ID.

Full Outer Join – The result contains all rows from both tables, paired when the joined columns are equal, and filled with blanks otherwise.

The new join appears in **All Joins | Joins on Selected Table** panel.

In order to create a join between two tables, they must each have a column that's compatible with a column in the other table. For example, a table with data for support cases has a column for the assigned engineer user ID that can be joined with the table of user data that has a user ID column.

In some cases, you may need to duplicate a table in order to join it several times without creating a circular join, or in order to join it to itself. You can also duplicate a table so it may be joined with different tables for different uses. Click the following icons above the Right Table to make a copy, change the name, or delete a table:

- **Copy** – Copies the selected table and gives it a name with sequential numbering. The copy appears in both Left Table and Right Table.
- **Change ID** – Changes the name of the selected table. The new name becomes the ID of the table throughout the Domain, and is updated everywhere it appears in the Domain Designer.
- **Delete** – Removes the table from both lists. If the deleted table was the only instance of a table, removing it on the Joins tab also removes it from the list of selected tables on the Tables tab.

You use the **All Joins | Joins on Selected Table** panel to see the defined joins, to remove a join, and to change the join type:

- **All Joins** – Lists all joins defined for the Domain.
- **Joins on Selected Table** – Lists only joins defined on the table you select, simplifying the view you have of many joins.
- **Join Type** – Changes the type of join.
- **Remove** – Removes the selected join definition from the list of joins and from the Domain design.

When you've created joins, one or more join trees appear on the **Calculated**, **Pre-Filters** and **Display** tabs. For example, if you join tables A and B, B and C, then join tables D and E, the result is two join trees. Columns of table A and table C may appear in the same report because their tables belong to the same join tree. Tables A and D are said to be unjoined; their columns may not be compared or appear in the same report. Tables that are not joined appear individually along with the join trees.

6.6.5 Calculated Fields Tab

You create a calculated field for the Domain by writing an expression that computes a value based on the data in one or more columns of the same join tree. All columns in the expression for a calculated field must be from the same join tree.

To create a calculated field:

1. Select the Calculated Fields tab.
2. In **Field Name**, enter a short name for the calculated field. This name becomes the field's ID in the Domain.

Later you can give the field a more meaningful label and full description.

3. In **Type**, select a datatype for the calculated field. The expression you write must return a value of this type.

   Generally, this datatype matches the datatype of the columns in the expression. Therefore, you need to be familiar with the datatypes of columns in the data source.

4. Enter an expression to compute the value of the calculated field.

   Write expressions using the Domain Expression Language, fully described in **"The DomEL Syntax" on page 268**.

   To insert a reference to the value of another column:

   a. Expand the join-tree to find its table and double-click the column name.

      The column name appears in the expression at the cursor, qualified by its table name.

   Calculated fields may be used to compute other calculated fields.

   b. Double-click the calculated field name to insert a reference to it in the expression.

   Do not insert a column reference from unjoined trees. The Domain Designer does not validate expressions as they are written, and the unjoined trees will not be available.

5. Click **Save Field** to save the new calculated field.

After saving the calculated field, the Domain Designer validates the expression and warns you of any errors. If there are errors, use the indications of the error message to help correct the expression. After validation, a calculated field appears in the table or join tree. In **Available Fields**, you'll recognize calculated fields by this icon .

An expression that uses no columns has a constant value. For example, you might create an integer field named Count that has the value 1 and later has a default summary function to count all occurrences. Constant fields are independent of join trees and automatically appear in a set called Constants.

To view, edit, or delete the definition of a calculated field:

1. Cancel any input in the calculated field editor, then click the name of the field in Available Fields.
2. Modify its name, its type, or its expression.
3. Click **OK** to save the new definition, or click Cancel if you just viewed the field definition.

   Click **Delete Field** to remove the calculated field from the Domain design.

6.6.6 The Pre-filters Tab

A filter on one or more columns reduces unwanted data in reports. For example, financial reports for the current fiscal year may need data from the previous fiscal year for comparison, but nothing earlier. It's good practice to filter out irrelevant data to reduce the size of query results and processing time.

Also, reports based directly on the Domain can define their own filters. Putting often-used filters in the Domain design avoids the need for each user to define filters independently and also reduces the chance for errors.

You can define a filter on a column you don't plan to expose in the Domain. The filter remains active and only data that satisfies all defined filters appears to report users. For example, you can filter data to select a single country, in which case it doesn't make sense for the column to appear in a report. However, you should clearly document such data restrictions in the description of the Domain, so users understand what data is accessible through the Domain.

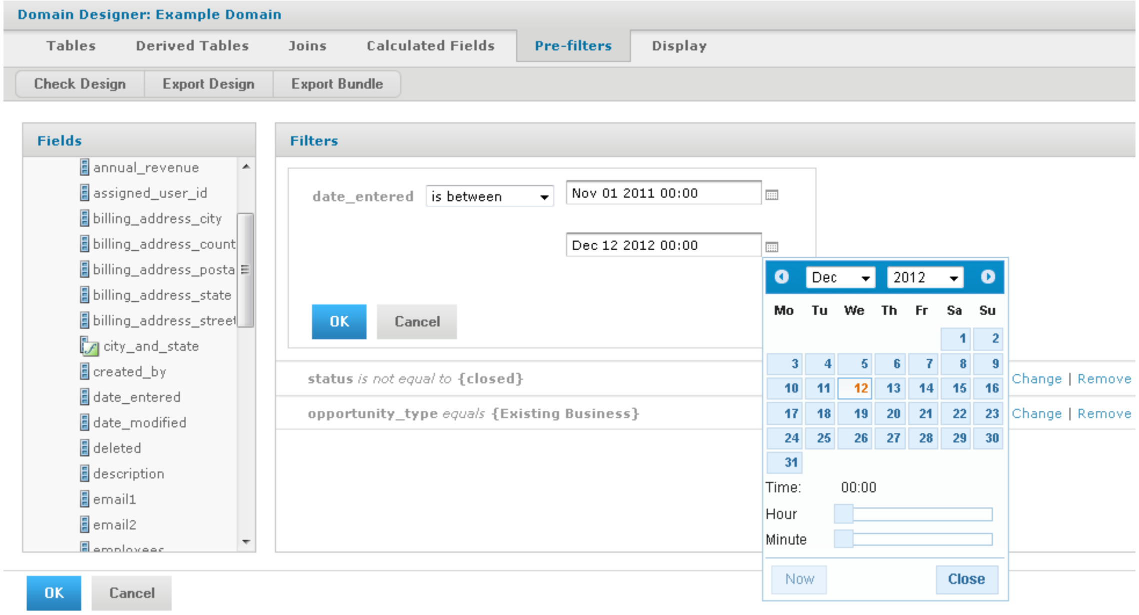
To define a filter:

1. Double-click a column in **Fields**.

   Alternatively, you can drag a column to the Filters panel. The column appears in the **Filters** panel with a list of conditional operators you can apply to that column.

2. Choose the comparison operator and filter value from the drop-down.

   In the Filters panel, the choice of comparison operators depends on the column's datatype. For example, strings offer a choice of search operators and dates offer time comparison operators. The filter value depends on the datatype and comparison operator. For example, if you select a date column with the `is between` operator, the Filters panel displays two calendar widgets for specifying a date range:



Domain Designer: Example Domain

date_entered is between Nov 01 2011 00:00 Dec 12 2012 00:00

status *is not equal to* {closed}

opportunity_type *equals* {Existing Business}

Figure 6-23 Filters Panel of the Domain Designer

Text columns have both substring comparison operators such as `starts with` or `contains` and whole string matching such as `equals` or `is one of`. When you select a whole string matching operator, the panel displays a list of all existing values for the chosen column, retrieved in real-time from the database. If more than 50 values are available, use search 🔍 to narrow the list. For multiple value matching, double-click the available values to select them. You may perform multiple searches and select values from each list of results.

To define a filter that compares two columns of the same datatype, Select both columns, and drag them to the Filters panel.

3. Click **OK** to define the filter.

   Filters shows all the filters you have defined. The overall filter applied to the data is the logical AND of all conditions you defined.

In Filters, click **Change** to modify a filter you defined. Click **OK** to save the changes. To remove a row from the list, select it and click **Remove**.

6.6.7 Display Tab

On the Display tab, you specify the columns and calculated fields to expose through the Domain and how they appear. Typically, you select only columns that are useful in building or filtering reports, and omit other columns for simplicity. You can also modify display properties like the labels and descriptions of tables and sets and items for each specified column. Using meaningful labels and descriptions helps users.

The Display tab contains three panels:

- **Resources** – Here you can view resources as either a **Join Tree** or a **Table List**:
  - Join Tree – Displays joined, unjoined, and calculated tables for this Domain. Joined tables appear as join trees.
  - Table List – Displays the selected and calculated tables for this Domain.
- **Sets and Items** – Lists resources that appear to report creators who use the Domain.
- **Properties** – Displays and modifies properties of selected sets and item.

In the Table List view, you can use the **Delete Table** button to remove the selected table from the **Resources** panel and from the Domain design. The table no longer appears on the Joins tab, and any joins to the table are deleted. Use carefully.

To specify which columns and calculated fields are exposed to users of the Domain, move them from the **Resources** panel to the **Sets and Items** panel:

- Set – A group of items independent of the tables in which the columns originate.
- Item – A column or calculated field, along with its display properties, that you want to appear in the Domain.

All items in a set correspond to columns in a join tree. Sets are optional. You can create a list of items outside of any sets. If you want to display only a few of the columns from the join tree, first create a new set in Sets and Items. Next, add items from the join tree in Resources.

To move a resource to Sets and Items, you can drag it from Resources and drop it in Sets and Items, or use one of the following buttons:

Add to Sets – Select a resource in Resources and click this icon to add it to Sets and Items. Tables are added as sets. Columns are added as items outside of any set.

Add to Selected Set – Select a destination set in Sets and Items, select a resource in Resources, and click this icon. Tables are added to Sets and Items as subsets. Columns are added as items within the destination set. You can also expand sets in Sets and Items and drag tables and columns from Resources to the destination set.

New Set – Creates a new subset of a selected set or a top-level set if no set is selected.

Delete – Removes the selected set or item. You can also double-click the object or drag and drop it in a blank area of Resources. Items removed from Sets and Items automatically reappear in Resources.

To move most of the tables and columns in Resources to Sets and Items:

1. Double-click the join tree name in Resources, or drag-and-drop it from Resources to Sets and Items. All tables are created as sets, and columns are created as items within the sets.
2. Remove any unwanted sets or items individually from Sets and Items using the **Delete Item** button.

To change the order of items within a set in Sets and Items:

1. Select the item.
2. Reposition the item using one of the following buttons:

   ⏫ Move to top.

   ▲ Move up.

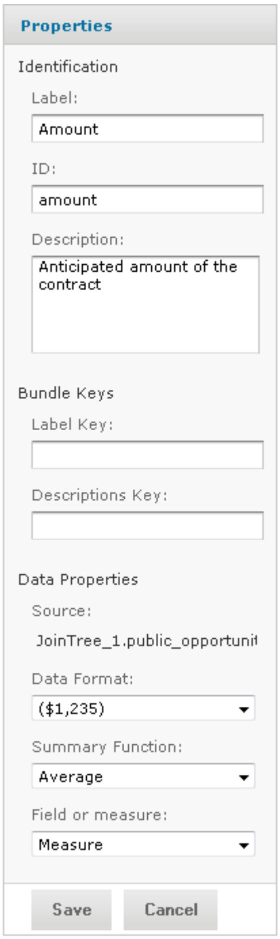
   ▼ Move down.

   ⏬ Move to bottom.

To move items and subsets into other sets:

1. In Sets and Items, select the item or subset you want to move.
2. Click **Delete Item**.
3. Select the destination set.
4. In Resources, select the item or subset deleted from Sets and Items.
5. Click ⇥ to add the item or subset from Resources to the destination set. Subsets always appear after the items in a set.

6.6.8 The Properties Panel



Properties

Identification

Label: Amount

ID: amount

Description: Anticipated amount of the contract

Bundle Keys

Label Key:

Descriptions Key:

Data Properties

Source: JoinTree_1.public_opportunit

Data Format: ($1,235)

Summary Function: Average

Field or measure: Measure

Figure 6-24 Table of Properties on the Display Tab of the Domain Designer

In the Properties panel of the Display tab, you can view, edit, and save the properties of a set or item. You select an item in the Sets and Items panel, and click the **Edit** button. For example, open the Simple Domain example in the repository for editing. On the Display tab, select the Opportunities: Amount item, then click the **Edit** button in Properties. Set the Data Format and Summary Function, as shown in the previous figure. Click the **Save** button. The property settings determine how sets and items appear to users of the Domain.

The following table shows the properties and possible actions you can perform on them, depending on what you select in the other panels. For example, if you select the ID of a table in the Resources panel, you can view and edit the ID.

Panel	Selection	Properties	Possible actions
Resources (Table List view)	Table	• ID of the table • Name of the data source • Name of the table in the data source	• View and edit • View only • View only

Panel	Selection	Properties	Possible actions
Resources (Table List view)	Field (column)	• ID of the field • Name of the data source • Name of the field's table in the data source • Java datatype of the field	• View only • View only • View only • View only
Resources (Join Tree view)	Table, field, or join tree	None	None
Sets and Items	Set	• Label of the set • ID of the set • Description of the set • Internationalization keys for the set	• View and edit • View and edit • View and edit • View and edit
Sets and Item	Item	• Label of the item • ID of the item • Description of the item • Internationalization keys for the item • Source, table.field • Data format • Default summary function • Field or measure	• View and edit • View and edit • View and edit • View and edit • View only • View and edit • View and edit • View and edit

Labels and descriptions are visible to users of the Domain. Descriptions of sets and items appear as tooltips in the Ad Hoc Editor to help report creators understand their purpose. The internationalization keys are the property names of strings in locale bundles. Selecting an item in Sets and Items displays the internationalization keys, if any, for the item under **Bundle Keys**.

After defining the list of sets, subsets and items, refine the Domain's appearance by renaming and providing descriptions for sets, subsets, and items. Click **Edit** in the Properties panel to modify properties. The following table describes each of the properties in detail:

Property	Appears On	Description
ID	Table, Field Set, Item	An identifier used within the Domain. Default table and field IDs are based on the names in the data source, but you can change the ID of a table as long as it remains unique. Set and item IDs are a separate namespace in which each ID must be unique, although based on table and field IDs by default. The ID property value must be alphanumeric and cannot start with a digit. You should not change IDs for a Domain that has been used to create Topics and reports. See **"Maintaining Referential Integrity" on page 237**.
Data Source	Table, Field	Alias of the data source for the selected field or table.

Property	Appears On	Description
Source Table	Table, Field	Name of the selected table, or of the field’s table, in the data source. Does not change when the ID property of a table is modified.
Datatype	Field	Java datatype of the selected field.
Label	Set, Item	User-friendly name displayed in the Data Chooser and the Ad Hoc Editor.
Description	Set, Item	User-friendly description displayed as tooltip on the label in the Ad Hoc Editor. The description helps the report creator understand the data represented by this set or item.
Label Key Description Key	Set, Item	Internationalization keys for the label and description properties; locale bundles associate this key with the localized text for the label or description. Keys may use only characters from the ISO Latin-1 set, digits, and underscores (_).
Source	Item	References the Domain names of the table and field associated with this item; the syntax is `Domain_jointree.Domain_table.datasource_field`.
Data Format	Item	Default numerical format (such as number of decimal places) for the item when used in a report.
Summary Function	Item	Default summary function of the item when used in a report. The available functions depend on the field's data type (Boolean, date, numeric, or text). See **Summary Calculations** for more information.
Field or Measure	Item	Designates the item as either a qualitative value (field) or quantitative value (measure). By default, all numeric types are assumed to be measures, and all non-numeric items are plain fields. Use this setting to override the default. For more information, see **“Terminology” on page 203**.

You can use the **Export** menu to export the design or locale bundle, as described in the next section.

6.6.9 Designer Tool Bar

The tool bar buttons operate on the Domain design in its current state, regardless of which tab is selected.

- **Check Design** – Validates the Domain as described in the next section.
- **Export Design** – Exports the XML design for the Domain. Do this to edit the XML in an external editor, for example to duplicate settings with copy-paste or to enter a complex formula. Exporting the XML design from the Domain Designer is easier than writing it from scratch. For more information, see **“The Domain Design File” on page 245**.
- **Export Bundle** – Generates the internationalization keys and saves them to a Java properties file that serves as a template for the locale bundles. Do this to create a template for the locale bundles after you've defined the sets and items on the Display tab. You can select a check box to automatically generate keys based on the set and item IDs. This option is visible only after you define localized sets or items. For more information, see **“Locale Bundles” on page 277**.

You can save time editing the properties for a large number of sets and items by using the exported XML file instead of using the properties table on the Display tab. For more information about saving and uploading XML, see **"The Domain Design File" on page 245**.

Before exporting a bundle, make sure the set and item IDs are finalized because they're used to generate the keys. The generated keys are added to the Domain design and appear in the table of properties. For more information about the internationalization keys, see **"Locale Bundles" on page 277**.

6.6.10 Domain Validation

Validating a Domain ensures the consistency of all its components. The Domain Designer checks the syntax of files when they're uploaded, but overall consistency must be checked when saving a new or edited Domain.

When you click **Check Design**, the Domain Designer does the following:

1. Verifies that the tables and columns of the Domain design exist in the data source.

   In special cases where you need to create a design before the data source is available, this step can be omitted by setting a parameter in the server configuration file. See the *JasperReports Server Administrator Guide*.

2. Verifies that all items in each defined set originate in the same join tree.
3. Verifies that all items reference existing columns.
4. Verifies that derived tables have valid SQL queries.
5. If a security file has been uploaded, verifies that all items and sets in the security file exist in the Domain design.

If validation fails, the Domain Designer remains open and a message appears to help you correct the error. Make the necessary changes to the settings and save again. If the settings are in the uploaded files, edit the files and upload them again.

Validation is important because the Domain design may include derived table queries and calculated field expressions entered by the user.

Validation occurs at the following times:

- When opening the Domain Designer. This check detects any inconsistencies in Domain designs from uploaded files.
- When navigating from tab to tab under certain circumstances. This check detects problems on the tab where they occur.
- When changing the data source.
- When exporting the Domain design file.
- When clicking **Check Design** in the Domain Designer.
- When clicking OK to exit the Domain Designer.

Unless you click Check Design, no message appears when validation succeeds. When validation fails, however, a message appears to help you correct the error.

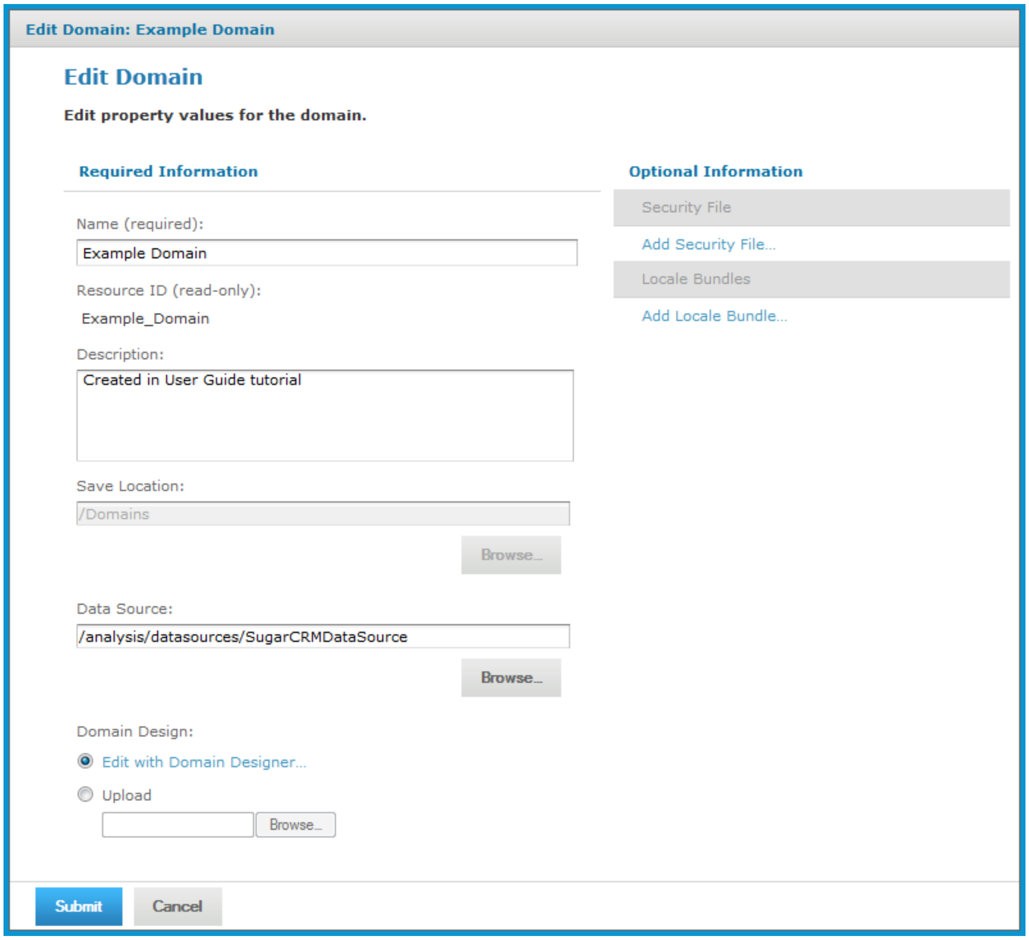
6.7 Editing a Domain

You can add, change, and delete domain components.

- Use extreme caution when editing Domains that might have been used for reports and Domain Topics. A Domain specifies the data source for Domain Topics and reports. These Domain Topics and reports might fail if you edit the underlying Domain; if you delete the underlying Domain, dependent reports fail.
- Domains that include advanced joins should not be edited in the Domain Designer. The Domain Designed will not prevent you from attempting to edit such a Domain, but if you have a Domain that uses advanced joins, and you open it in the Domain Designer and then save it, your advanced joins will be lost or changed.

To edit a Domain:

1. Log into the server as an administrator and select **View > Search Results**.
2. Locate the Domain.
   a. In the Filters panel, under All Types, click **More choices**.
   b. Click Domains.
3. Right-click a Domain and select **Edit** from the context menu. The Edit Domain page is similar to the Add New Domain page documented in **“Using the Add New Domain Page” on page 220**.



Edit Domain: Example Domain

Edit Domain

Edit property values for the domain.

Required Information

Name (required): Example Domain

Resource ID (read-only): Example_Domain

Description: Created in User Guide tutorial

Save Location: /Domains

Data Source: /analysis/datasources/SugarCRMDataSource

Domain Design:

● Edit with Domain Designer...

○ Upload

Optional Information

Security File

Add Security File...

Locale Bundles

Add Locale Bundle...

Figure 6-25 Edit Domain Page

4. Change the Domain's **Name** and **Description**.
5. Browse to locate a new **Data Source**. Because the Domain design relies extensively on the data source, changing the data source makes sense only in certain cases. Examples include:
   - Switching to a data source with a different schema but with the same tables, for example, when moving from staging to production.
   - Switching from a regular data source to a virtual data source that contains the original data source. In this case, JasperReports Server attempts to locate the original data source inside the virtual data source and create the correct prefix. Derived tables and calculated fields are not preserved.
   - Switching from a virtual data source to one of the underlying data sources. In this case, any resources that are not in the selected data source are deleted along with any dependent information, such as derived tables, joins, calculated fields, or labels that depend on the deleted resources.

If you change to a data source with a different database, the definitions in the Domain design become invalid and you can't save the Domain.

Before you switch the data source for a Domain, back up the Domain and export a Domain design file. See **"Switching the Data Source of a Domain" on page 281** for more information.

6. Click **Edit with Domain Designer**. For instructions, see section **"Using the Domain Designer" on page 223**. Alternatively, select the **Upload** option, and browse to locate a Domain design file. You can import the XML file of the Domain design after exporting it and making modifications in an external editor. See "**Maintaining Referential Integrity**" below if you need to remove items in the Domain.
7. To add or replace the security file or locale bundles for the Domain, click **Add Security File** or **Add Locale Bundle** in Optional Information. The Resources page is further documented in section **"The Domain Design File" on page 245**.
8. Click **Submit** to update the Domain. To close the Domain Designer without modifying the Domain stored in the repository, click **Cancel**.

After modifying a Domain, you must clear the Ad Hoc cache of all queries based on the Domain. This removes any data that was based on the old instance of the Domain and avoids inconsistencies in new reports. For instructions, see the *JasperReports Server Administrator Guide*.

6.7.1 Maintaining Referential Integrity

When editing a Domain, you're responsible for maintaining referential integrity between the items in the Domain and any Domain Topics or reports that have been created from that Domain. referential integrity means that at the time a Domain Topic or report is opened or run, all the items that the Domain references are still defined in the Domain. If you modify a Domain by removing sets or items, you must be sure that no Domain Topics or reports are based on those sets or items. Even if the underlying tables and columns still exist in the database, the sets and items referenced in the Domain must still exist in the Domain.

Domain items are identified by their IDs and the IDs of the sets in which they are located. Changing the ID of an item or moving it to a different set also makes it unavailable to any Topics and reports that referenced the ID. To change the name of an item or set, edit its label property, not its ID (see **"The Properties Panel" on page 232**). Moving an item is considered the same as deleting it from its current location and adding it elsewhere.

Any report that references a deleted item no longer runs, nor can you open it in the Ad Hoc Editor again. A Domain Topic that references a deleted item causes errors when used in the Ad Hoc Editor. You can, however, open a Domain Topic for editing and remove references to deleted items. That action, however, allows only new reports to use the Domain Topic, and does not fix the broken reports based on the items deleted from the Domain Topic.

The granularity of referential integrity is at the individual set and item level. This means that changes to sets and items not used in a given report or Topic do not affect the report or Topic. For example, if you delete an item used by Topic A but not Topic B, then Topic A fails and reports based on Topic A that included the item fail. But Topic B and its reports are unaffected, as are any reports based on Topic A that did not include the deleted item.

6.7.2 Fixing Referential Integrity Problems

Generally, to maintain referential integrity, avoid deleting items from a Domain that reports need to use. Sometimes, removing an item or set from a Domain is unavoidable. For example, you included an item that exposes confidential data to unauthorized users, or the underlying database has changed, making the item invalid.

In a typical scenario, an administrator or data analyst creates a Domain and many end-users create Topics and reports based on it. When deleting Domain items in such a situation, tell users beforehand to modify reports to prevent reports from using the item. Inevitably, a user fails to modify a report as requested, and needs to fix the broken report. The server provides a method for fixing the report. The user can replace the deleted item with a placeholder, open the report, and edit it, even after the original item has been removed.

The following procedure assumes a user created a report based on a Domain. The Domain is similar to the one described in **"Example of Creating a Domain" on page 204**. The Domain exposes the `accounts_opportunities` table that contains no useful columns and is necessary only to join the `accounts` and `opportunities` tables. Unfortunately, some users were confused by the additional items exposed by the Domain, and used the `date_modified` item in their reports. You want to delete the `accounts_opportunities` table from Sets and Items, so it is no longer exposed to users.

To fix referential integrity problems:

1. Locate the problematic Domain in the repository, right-click, and choose **Edit**. The Edit Domain page appears.
2. Click **Edit with Domain Designer.**
3. In the Domain Designer, click the **Display** tab and select the public`_accounts_opportunities` set in **Sets and Items.**

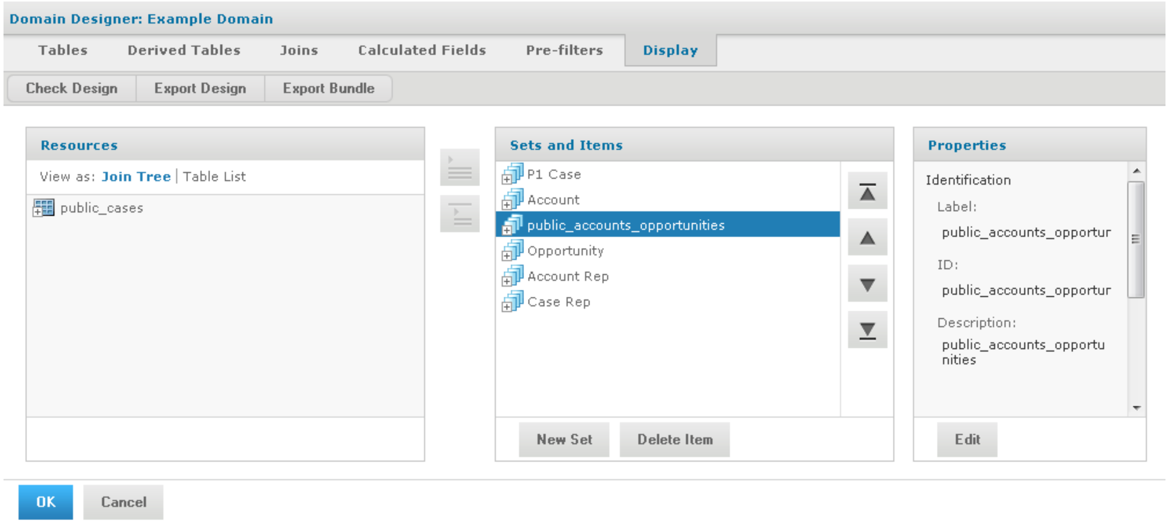
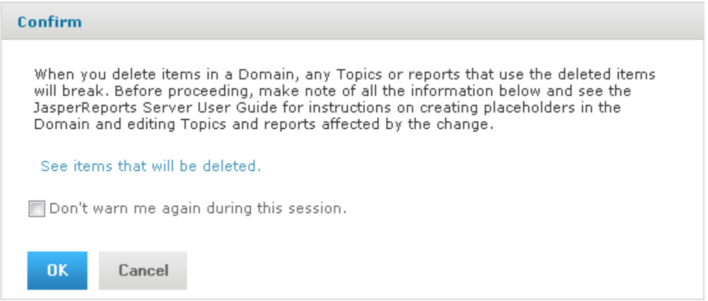


Figure 6-26 Deletion of Items in Use in Reports from a Domain

4. Click **Delete Item**. The **Confirm** dialog box appears, indicating that the item is in use and that a potential referential integrity problem exists. Click **See items that will be deleted** to access the **Detail** dialog.


Confirm

When you delete items in a Domain, any Topics or reports that use the deleted items will break. Before proceeding, make note of all the information below and see the JasperReports Server User Guide for instructions on creating placeholders in the Domain and editing Topics and reports affected by the change.

See items that will be deleted.

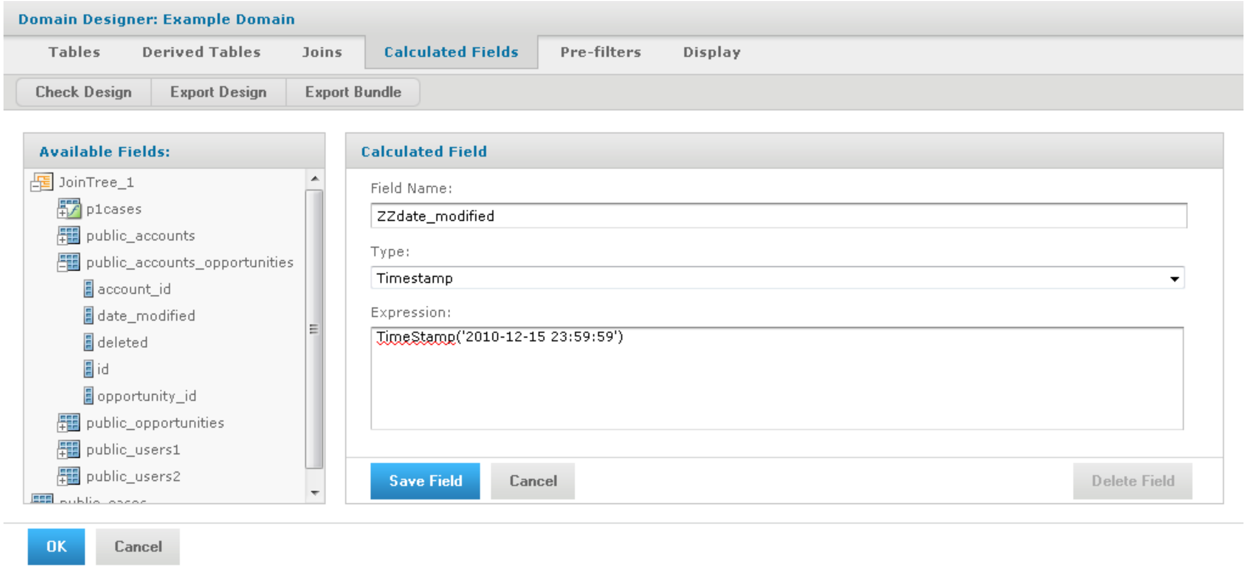
☐ Don't warn me again during this session.

Figure 6-27 Indication of Referential Integrity Problem

5. Close the Detail dialog box and click OK in the Confirm dialog box.
6. Click the **Calculated Fields** tab.

Because a calculated field is a custom column of a user-selected type, it can serve as a placeholder. The value of the custom column is irrelevant, but it needs to be recognized as a placeholder when it appears in a report.

7. Enter a field name, type, and an expression that returns a constant value of the same type as the deleted item, as shown in **Figure 6-28**, and click **Save Field**.



Field Name: ZZdate_modified

Type: Timestamp

Expression: TimeStamp('2010-12-15 23:59:59')

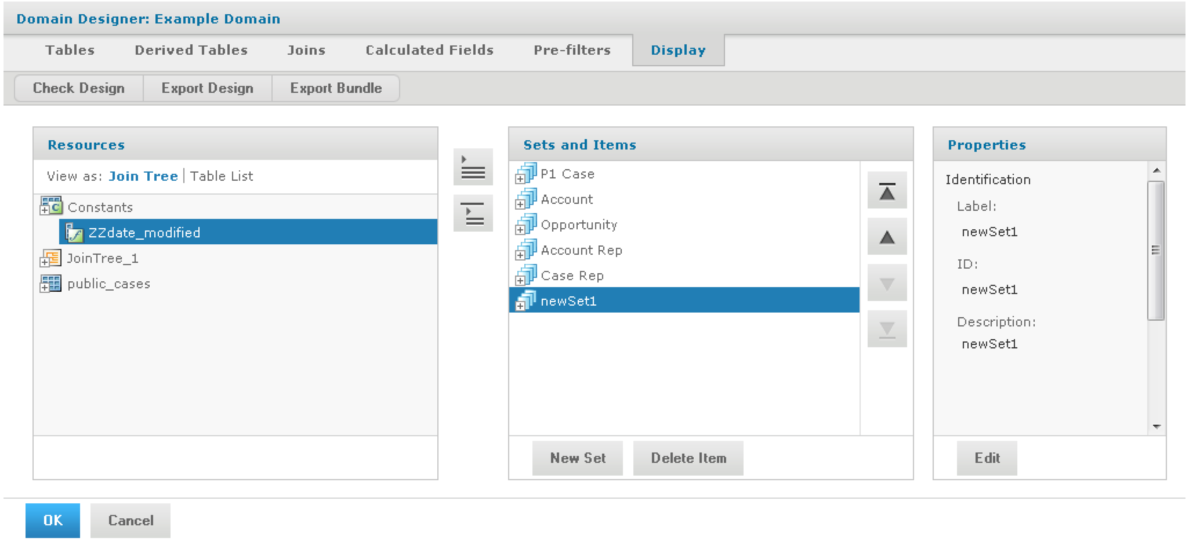
Figure 6-28 Creation of Calculated Field as Placeholder for a Deleted Item

The ZZ prefix in Name distinguishes the placeholder from other columns in the Domain. The value in Type must match the datatype of the deleted item, in this case a timestamp. For information about calculated field datatypes, see **"Datatypes" on page 268**.

8. Click the **Display** tab.

A placeholder item must be located in the placeholder set to mimic the structure of the deleted item, and generally, in exactly the same path of nested sets as the original item. You deleted the accounts_ opportunities set, so you need to create a set for the placeholder item.

9. In Sets and Items, click **New Set** to create the placeholder set.
10. In Resources, expand the list of Constants and select `ZZdate_modified`.



Identification

Label:
newSet1

ID:
newSet1

Description:
newSet1

Figure 6-29 Calculated Field Set Up as a Placeholder

11. Click ⇥ to add the ZZdate_modified to the new set.
12. In Sets and Items, select the placeholder set, newSet1.
13. In Properties, click **Edit** and change these properties:

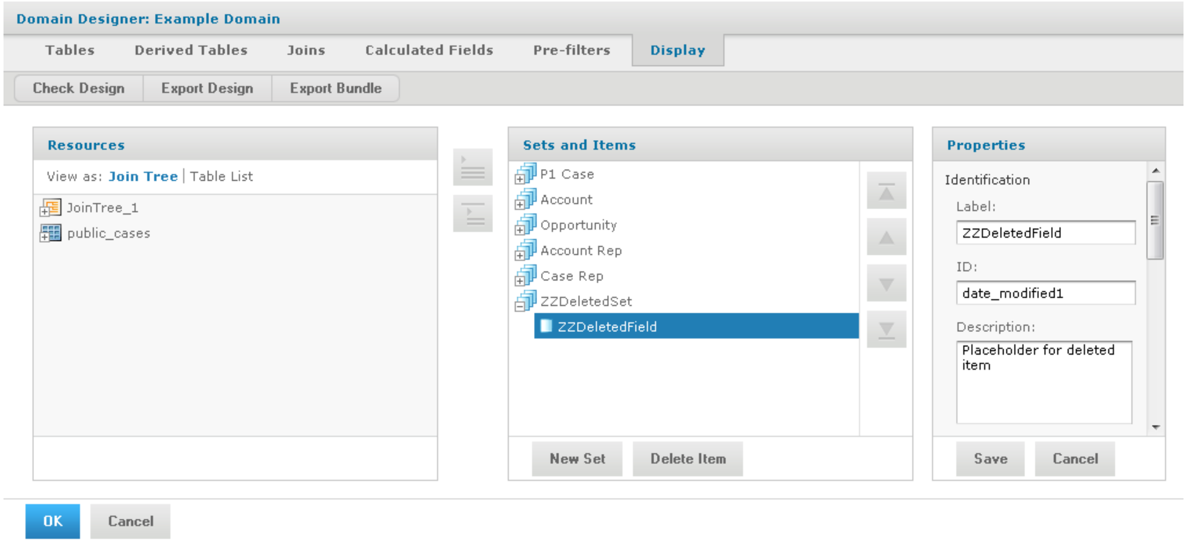
Property	New Value
ID	`accounts_opportunities`
Label	`ZZDeletedSet`
Description	`Placeholder for deleted set`

14. In Sets and Items, select ZZdate_modified.
15. In Properties, click **Edit** and change these properties:

Property	New Value
ID	`date_modified1`
Label	`ZZDeletedField`
Description	`Placeholder for deleted item`

Distinguishing the placeholders from real items by using the ZZDeletedSet and ZZDeletedField labels discourages their use in building new Ad Hoc views. The IDs, although invisible to users, must be identical to those of the deleted item and set, so the server can open and run the view.

You can see the modified properties for the placeholder item in **Figure 6-30**.



Identification

Label: ZZDeletedField

ID: date_modified1

Description: Placeholder for deleted item

Figure 6-30 Modification of ID Properties to Match the Deleted Item

16. Click **OK** in the Domain Designer and then **Submit** on the Edit Domain page.

Alternatively, you can edit the XML design file of a Domain to fix referential integrity problems. The modifications are the same as described in this procedure, but the order is slightly different.

To fix referential integrity problems in the XML design file of a Domain:

1. Instead of deleting the item, first create the constant calculated field. You can do this in the Domain Designer even before exporting the design file. You must be familiar with the design file syntax to create the calculated field with the correct type and expression, as described in **"Structure of the Design File" on page 249**.
2. Next, locate the item definition and change its `resourceID` so that it references the newly created calculated field. This removes the reference to the unwanted column, and replaces it with an item that references the placeholder field.
3. Update the other properties of the item and its enclosing set, using the values shown in **Figure 6-30**. For more information, see **"Representing Sets and Items in XML" on page 265**.
4. Save the design file and upload it as described in **"Uploading a Design File to a Domain" on page 248**.

6.8 Copying a Domain

In some cases, for example, when you are making extensive changes to a Domain using the Domain designer, you may want to create a copy of a Domain. Because each Domain has a Resource ID, you must copy the Domain to a different folder in the repository.

To copy a Domain via the UI:

1. Log in to the server as an administrator and select **View > Search Results**.
2. Locate the Domain.
   a. Choose **View > Search Results**.
   b. In the Filters panel, under Types, click **More choices**.

c. Click Domains.

3. Right-click an existing Domain and select **Copy** from the context-menu.
4. Select **View > Repository** from the menu.
5. Right-click a folder that does not contain the original Domain and select **Paste** from the context-menu.

To change the Resource ID of the Domain, export the Domain design file as described in **"Exporting the Design File from a Domain" on page 246**, manually edit the Domain's resource ID, then upload the design file as described in **"Uploading a Design File to a Domain" on page 248**.

Chapter 7 Advanced Domain Features

The security file and locale bundle are optional information in a Domain. When included, these options provide data access permissions and localized strings for reports based on a Domain. The data security and localized strings are defined in external files that Domain creators must upload to the Domain. Security and locale options take effect in the Ad Hoc Editor when creating the report and in the final output when running the report.

The Domain design can be exported to an XML file and edited outside of JasperReports Server. This gives Domain creators an alternative to using JasperReports Server to modify the design and simplifies sharing of Domains between systems. This chapter documents the syntax for the security file and locale bundle and discusses important considerations in writing or modifying file contents. The discussion begins with the Domain design file because the security file and locale bundles rely on certain values it contains.

For a definition of the terms column, field, and item as used in Domains, see **"Terminology" on page 203**.

This chapter contains the following sections:

- **The Domain Design File**
- **Structure of the Design File**
- **The DomEL Syntax**
- **Security and Locale Information for a Domain**
- **The Domain Security File**
- **Locale Bundles**
- **Internationalized Databases**

7.1 The Domain Design File

The Domain design specifies the tables in the data source, derived tables, joins, calculated fields, and filters, as well as how those elements appear to users. In JasperReports Server, the Domain design can be created interactively through the Domain Designer, however there is also an XML file format for exporting and uploading the settings. The text file containing a Domain design in XML is called a design file.

The XML in a design file is a hierarchy of elements and attributes that specifies all the settings in the Domain. The elements and attributes are defined by an XML schema provided in an XSD file. In addition, there are constraints on a Domain design that are not expressed in the XML schema. A design file can be modified or written from scratch in an editor and uploaded to the server, as long as it conforms to the XML schema and the design constraints.

XML is not the native format of the Domain design. The XML file is only a representation from which the design can be inferred. A design has additional constraints that are not mapped in the XML format.

Here are several common use cases for working with design files:

- Completing the elements of a new design. Use the Domain Designer to define as much of the design as possible, then export the design file and add your handwritten code to the exported file. For example, you can enter the SQL query for a derived table or complex expressions for a calculated field.
- Working with very large Domains. The Domain Designer makes it easy to select all tables and columns and expose them as sets and items, but editing the labels and descriptions of dozens of items is faster when they appear in a single design file.
- Making repetitive changes to an existing Domain design. If the database changes or you want to move a design file to a different system, you may need to edit each table or item in the design in the same way. Using search-and-replace on an external editor does this quickly.
- Creating locale bundles and security files as described in the other sections of this chapter. These optional files refer to elements of the Domain design, and it is often more convenient to copy-paste them between external files.
- Creating a Domain design from scratch. It is possible to write a valid XML file that meets the constraints of JasperReports Server and defines a Domain design. However, due to the complexity of a valid design file, it is much easier to begin with a basic design file exported from the Domain Designer or to modify an existing design file.

After editing, a design file can be uploaded, validated, and used to define the design of a new or existing Domain. When you open the design in the Domain Designer again, the modifications appear and are editable in the Designer. For example, a description you added in the XML design file appears in the Properties table of the Display Tab, and you can edit it again in the Designer. Other elements of the XML file appear on some or all of the Designer tabs.

7.1.1 Exporting the Design File from a Domain

You can export the Domain design file from the Domain Designer and save it as an XML file.

To export a Domain design file:

1. Log into the server as an administrator and locate the Domain you want to export.
2. Right-click the Domain and select **Edit** to launch the Domain Designer.
3. Click **Edit with Domain Designer**.
4. Click **Export Design**.
5. If your data source supports schemas (subdivisions of a database), you're prompted to include the schema name in the table name. Select **Yes** if you intend to use this Domain design file with the same data source and schema. Select **No** if you intend to re-use this Domain design with other data sources or other schemas.
6. The browser usually gives you the choice of viewing or saving the file. To save the design file, select a location and give it a name. Be sure to keep the .xml suffix.

   The server validates the design before exporting the XML file. If errors are found, you can cancel the export. For more information, see **"Domain Validation" on page 235**.

7.1.2 Working With a Design File

The relationship of item definitions, column definitions, and actual database columns is the essence of the Domain itself and must be maintained when editing the design file. In order to be usable when uploaded to a Domain, a design file must meet the following conditions:

- It must be well-formed XML. This means that all syntax, spelling, and punctuation is correct so that the file contains a hierarchy of elements, attributes, and content.
- It must be valid with regard to the XML schema. The XML schema defines allowed element and attribute names and how they're nested in a hierarchical structure. This ensures that the elements and attributes are the same ones with the same meaning used by JasperReports Server. The following allowed versions of the XML schema of a Domain design are given in the XSD files located in <install-dir>/samples/domain-xsd/:

  <install-dir>/samples/domain-xsd/schema_1_0.xsd – Supports all schema elements up through JasperReports Server 6.1.

  <install-dir>/samples/domain-xsd/schema_1_2.xsd – Adds support for options introduced in JasperReports Server 6.1 to give you influence over which join paths are chosen when multiple options are possible. Also supports all previous schema elements. Domain Designer will display a warning when a user opens a schema_1_2.xsd design file that uses the join path schema elements.
- The design file must be internally consistent and define all the necessary elements of a Domain design. These are constraints that cannot be expressed in the XSD file because they are outside the scope of an XML schema.
- The tables and columns in the design must be consistent with their external definition in the data source of the Domain. The design must reference valid table and column names in the data source; in particular, table names for a design based on an Oracle RDBMS must include the database schema name. The data source also defines the datatype of a column, and the design must use that column accordingly. As a result, a design file is specific to a given data source and most likely fails when used in a Domain with a different data source.

In addition, the more complex elements of a design file have further constraints:

- SQL queries for a derived table must be valid with respect to the JDBC driver for the data source. For a virtual data source that combines data from data sources with different JDBC drivers, SQL queries are validated against Teiid SQL; for more information, see the Teiid Reference Guide under the Documentation link on the Jaspersoft Support Portal. Also, the tables and columns in the query must exist in the data source, and the columns in the results must match those declared in the design.
- Expressions for filters and calculated fields must be valid programmatic expressions in a local format called Domain Expression Language (DomEL). This format is documented in **"The DomEL Syntax" on page 268**.

The design of a Domain is stored internally in the repository. The XML is only a representation from which the design can be inferred, and it may have some validity errors that cannot be detected. As a result, the Domain resource and the XML may not remain totally synchronized through several cycles of exporting to XML, editing, and uploading. For example, the Domain Designer sometimes renames the result of a join (JoinTree_1), which affects the security file.

However, the Domain Designer also has limitations and cannot create some valid designs. For example, in a design file, you may select the columns of a table whereas you can only select whole tables on the Tables tab. Furthermore, the Domain Designer cannot read some valid designs, in which case you must not open the uploaded design file in the Domain Designer. These rare cases are documented in the following sections.

A design file is plain text and can be edited in any text editor. The server exports well-formatted XML, and if you want to make only a few changes or simple additions, a text editor is sufficient. For editing the content of the design file, a specialized XML editor ensures that the design file is well-formed, so you don't introduce other errors. If you want to make structural changes or write a design file from scratch, use an XML editor that understands the XML schema in the schema_1_0.xsd file. By loading both the XML and XSD files, this type of XML editor lets you insert elements and attributes only where they're allowed to ensure that the design file is valid.

However, no editor can enforce the internal and external constraints on a design file. The following section explains all of the possible elements and attributes of an XML design file and the various constraints that apply to each of them.

7.1.3 Uploading a Design File to a Domain

After you have modified an XML design file, you can upload it through the Add New Domain page. Alternatively, you can create a new Domain based on a modified file or even on a design file created from scratch.

To upload an XML design file:

1. Log into the server as an administrator and select **View > Search Results**.
2. Locate the Domain.
   a. Choose **View** > **Search Results**.
   b. In the Filters panel, under Types, click **More choices**.
   c. Click Domains.
3. To update a Domain, right-click the Domain and select **Edit** from the context-menu.

   To create a new Domain, select **Create > Domain** from the main menu.

   The Domain appears in the Data and Design page of the Edit Domain or Add New Domain dialog. If creating a Domain, you must select a data source before you can proceed.
4. Select **Upload** under Domain Design, then click **Browse** to find the XML design file. In the File Upload window, click **Open**.

   The design file overwrites any existing design without prompting. If you make a mistake or upload the wrong file, click **Cancel** on the Data and Design page and start over.

   The server validates the uploaded XML file. If there are syntax or semantic errors, the current design is not replaced. You can make changes to the XML file and upload it again until there are no errors.
5. If you used only supported features in the design file, verify the uploaded Domain design by selecting **Edit in Domain Designer**. Make sure the settings you made in the XML file appear as expected on the various tabs of the Domain Designer. If there are any errors or inconsistencies, you should make changes to the XML file, upload it again, and verify it again. The results of editing a design in the Domain Designer based on an inconsistent XML file are unpredictable. If you can't resolve the problem, you should click **Cancel** on the Data and Design page so the uploaded design is not saved.

   After the design appears correctly in the Domain Designer, you can make further modifications on any of the tabs.

   If you intentionally use syntax in the design file that Domain Designer does not support, do **not** launch the Domain Designer after uploading the file. The Domain Designer can have unpredictable results with XML designs it does not support.
6. Click **OK**, then click **Submit** to update the Domain in the repository.

7. If you modified an existing Domain, you must clear the Ad Hoc cache of all queries based on the Domain. This removes any data that was based on the old instance of the Domain and avoids inconsistencies in new reports. For instructions, see the *JasperReports Server Administrator Guide*.

7.2 Structure of the Design File

This section explains each of the XML elements and attributes in a design file and how they relate to the settings in the Domain Designer. The sample XML code is based on **"Example of Creating a Domain" on page 204**, with table names that do not include database schema names. Some elements have been added to show structures that did not appear in the example.

Because certain XML elements correspond to objects in the Domain design, this section refers to XML elements that *contain* Domain objects such as sets. This is a short-hand description that means the XML elements contain other XML elements that represent the Domain objects.

7.2.1 Top-level Elements of a Design File

The top-level container elements of a design file are:

- schema (no relationship to the database schema)
- itemGroups
- items
- resources

The following schema representation shows the top-level structure of the design file:

```
<schema xmlns="http://www.jaspersoft.com/2007/SL/XMLSchema" version="1.0"
    schemaLocation="schema_1_0.xsd">
 <itemGroups>...
 </itemGroups>
 <items>...
 </items>
 <resources>...
 </resources>
</schema>
```

- schema – The outer-most container element of a design file.
  When exported from the Domain Designer, the schema element includes the xmlns and version attributes.
  - The xmlns attribute specifies an XML namespace for all element names.
    This string must be unique to Jaspersoft, but it does not correspond to a valid URL. For more information, see http://www.w3.org/TR/REC-xml-names.
  - The version attribute gives the version of the XSD used to create this design file.
  - The schemaLocation attribute is often added by XML editors to locate the XSD file.
- itemGroups – Contains all the sets and items within sets in the Domain.
  Along with items, this element corresponds to all the sets and items defined on the Display tab of the Domain Designer. Therefore, itemGroups and items determine what users see when they create a report

based on this Domain. The sets and items defined under `itemGroups` and `items` must be internally consistent with the tables and columns under `resources`.

- `items` – Contains all the items that are not within sets.
  These correspond to the items at the root level of the Display tab. When all items are contained in sets, this element is absent.
- `resources` – Contains all the definitions in the design:
  - columns
  - tables
  - derived tables
  - joins
  - calculated fields
  - filters

  These definitions in the design correspond to the first five tabs of the Domain Designer:

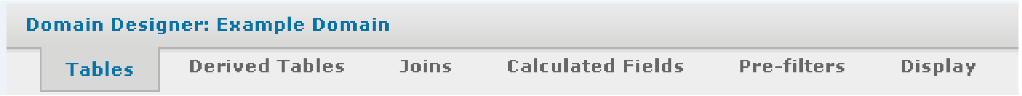


Figure 7-1 Domain Designer Tabs

Because the elements under `resources` refer to database objects, they must be externally consistent with the data source intended for this Domain.

although the `itemGroups` appear first in the design file, this section documents the `resources` first present design elements in the same order as the tabs of the Domain Designer.

The `resources` element contains the `jdbcTable` and `jdbcQuery` elements to represent database tables and derived tables, respectively. Join trees are represented as a `jdbcTable` element with additional contents to define the joins.

```
<resources>
 <jdbcTable datasourceId="SugarCRMDataSource" tableName="accounts" id="accounts">
  ...
 </jdbcTable>
 <jdbcTable datasourceId="SugarCRMDataSource" tableName="accounts_opportunities"
       id="accounts_opportunities">
  ...
 </jdbcTable>
 <jdbcTable datasourceId="SugarCRMDataSource" tableName="opportunities"
       id="opportunities">
   ...
  <filterString>opportunity_type == 'Existing Business'</filterString>
 </jdbcTable>
 <jdbcQuery datasourceId="SugarCRMDataSource" id="p1cases">
  <fieldList>
   <field id="account_id" type="java.lang.String"/>
   <field id="assigned_user_id" type="java.lang.String"/>
   <field id="case_number" type="java.lang.Integer"/>
   <field id="date_entered" type="java.sql.Timestamp"/>
   <field id="description" type="java.lang.String"/>
   <field id="id" type="java.lang.String"/>
   <field id="name" type="java.lang.String"/>
   <field id="resolution" type="java.lang.String"/>
   <field id="status" type="java.lang.String"/>
  </fieldList>
  <filterString>status != 'closed'</filterString>
  <query>select * from cases where cases.priority=1 and cases.deleted=false</query>
 </jdbcQuery>
 <jdbcTable datasourceId="SugarCRMDataSource" tableName="users" id="users1">
  ...
 </jdbcTable>
 <jdbcTable datasourceId="SugarCRMDataSource" tableName="users" id="users2">
  ...
 </jdbcTable>
 <jdbcTable datasourceId="SugarCRMDataSource" tableName="accounts" id="JoinTree_1">
  ...
 </jdbcTable>
</resources>
```

7.2.2 Representing Tables in XML

In the Domain Designer, you can select only entire tables, not individual columns. In the XML design file, however, you can specify any subset of columns that you need.

- jdbcTable – Represents a table or a copy of a table in the data source. A Domain design must reference all the tables it needs to access. The jdbcTable element is also used to describe join trees, but this case is documented separately in **7.2.4, "Representing Joins in XML," on page 253**. All three attributes of jdbcTable are required:
    - datasourceId – Alias that identifies the data source. When created in the Domain Designer, this is the data source alias defined in **"Derived Tables Tab" on page 226**. When creating a design file, this alias may be any name you choose, but it must be identical for all tables and derived tables. When uploading the file, the datasourceId automatically becomes the alias associated with the data source defined for the Domain.
    - tableName – Literal name of the table in the data source. Depending on the data source type and the database, this name may include additional information:
        - For PostgreSQL and Oracle-based data sources, the table name includes the database schema name in the form schema.table, if you opted to include schema names when exporting the design.
        - For PostgreSQL, Oracle, and other schema-based data sources within a virtual data source, the table name includes the data source prefix (defined when the virtual data source was created) and the database schema name in the form dataSourcePrefix_schema.table, if you opted to include schema names when exporting the design. The table name includes the data source prefix in the form dataSourcePrefix.table if you opted not to include schema names when exporting the design.
        - For MySQL and similar data sources within a virtual domain, the table name includes the data source prefix and the database name in the form dataSourcePrefix_database.table.
    - id – Table ID used to reference the table in the Domain design. If you copy a table to join it multiple times, each copy has the same datasourceId and tableName but must be given a different id.
- fieldList – A container for field elements. Required on jdbcTable elements because it would not make sense to have a table without columns in the Domain design. You must reference all the columns you use in the Domain.

- `field` – Represents a column of a table in the data source. All the columns you want to reference in the Domain defined with this element. Both attributes of `field` are required:
  - `id` – Literal name of the column in the data source. As in the JDBC model that the data source is based on, the `id` must be unique within the `jdbcTable`, but not necessarily within the Domain.
  - `type` – The Java type of the column, as determined from the data source by the JDBC driver. The available types are shown in the following table.

`java.lang.Boolean`	`java.lang.Float`	`java.lang.String`	
`java.lang.Byte`	`java.lang.Integer`	`java.math.BigDecimal`	`java.sql.Timestamp`
`java.lang.Character`	`java.lang.Long`	`java.sql.Date`	`java.util.Date`
`java.lang.Double`	`java.lang.Short`	`java.sql.Time`	

Unless you know the name and type of every column in the data source, it is often easier to select and export tables from the Domain Designer. The Domain Designer accesses the data source to find the names of all tables and columns, as well as their types. You may then export the XML design file with this information and refine your design.

If you have proprietary types in your database, the server may not be able to map its Java type from the JDBC driver. You can configure the mapping for proprietary types, as described in the *JasperReports Server Administrator Guide*.

Alternatively, you can override any mapping by specifying the `type` attribute for any given field in the XML design file. The server uses this Java type for the field, regardless of its mapping. If your proprietary type can't be cast in the specified type, the server raises an exception.

7.2.3 Representing Derived Tables in XML

Derived tables are similar in structure to tables, but they use the `jdbcQuery` element that contains the `query` element:

- `jdbcQuery` – Represents a derived table that results from an SQL query. Both attributes of `jdbcQuery` are required:
  - `datasourceId` – Alias that identifies the data source of the Domain. In the design file, this alias must be identical for all tables and derived tables.
  - `id` – Table ID used to reference the derived table in the Domain design. Any reference to the `id` of a `jdbcTable` may also reference the `id` of a derived table.
- `fieldList` – A required container of the `field` elements.
  When the derived table is created in the Domain Designer, the set of columns corresponds to the selection of columns in the query result on the Derived Tables tab.
- `field` – Represents a column in the results of the query.
  Only the columns represented by a `field` element are available for reference by other elements. The columns of a derived table must be among those returned by the query.
  - `id` – Literal name of the column in the query result. If the query gives an alias to the column in a `SELECT AS` statement, the `id` is the same as the alias. The `id` must be unique within the query results, but not necessarily within the Domain.
  - `type` – The Java type of the column, as determined from the data source by the JDBC driver.

- query – The SQL query sent to the database server.
  You can use any valid SQL that returns results, as long as the tables and columns in the query exist in the data source, and the columns in the result match the id and type of all field elements of the derived table given in the fieldList. The syntax for a valid SQL query does *not* include a closing semi-colon). If you add a derived table in the Domain Designer, it runs the query and generates columns based on the result set. You can then export the design file containing the generated column list.

SQL queries for a derived table are with respect to the JDBC driver for the data source. If you are working with a virtual data source, SQL queries are validated against Teiid SQL, which provides DML SQL-92 support with select SQL-99 features. For more information, see the Teiid Reference Guide under the Documentation link on the Jaspersoft Support Portal.

The following sample query in PostgreSQL selects some columns, including a field calculated in the SQL, from the result of a join with sorted results. In this case, only exp_date, store_id, amount, currency, conv, and as_dollars can be exposed as columns of this derived table.

```
<query>
 select e.exp_date, e.store_id, e.amount, c.currency, c.conversion_ratio conv,
  amount * c.conversion_ratio as_dollars
 from expense_fact e join currency c
  on (e.currency_id = c.currency_id and date(e.exp_date) = c.date)
 order by e.exp_date
</query>
```

The preceding example is not valid for SQL Server. SQL queries for a derived table are resolved using the JDBC driver for the data source and the derived table becomes a subquery in the generated SQL. SQL Server requires a TOP or FOR XML clause in any subquery that uses ORDER BY.

A derived table provides an alternative way to create joins and calculated fields. Here are some things to keep in mind when deciding how to implement the Domain:

- Unlike joins defined in the Domain, joins within a derived table are not restricted to equality comparisons when uploaded to the Domain Designer. For more information, see **Representing Joins in XML**.
- Unlike calculated fields in DomEL, calculated fields within derived tables may use any function call recognized by the RDBMS. See **"The DomEL Syntax" on page 268** for restrictions on function calls in calculated fields.
- The Domain mechanism applies filters, aggregation, and joins to derived tables by wrapping the SQL in a nested query, which may be less efficient on some databases than the equivalent query generated for a non-derived table.

7.2.4 Representing Joins in XML

A join is represented in the design file as a special jdbcTable element. There are several ways to represent joins in the Domain schema file:

- basic join syntax – Supports joins based on equality comparisons only. Available in all versions of JasperReports Server that use Domains. This is the only syntax that is fully supported in Domain Designer.
- advanced join syntax – In addition to joins based on equality comparisons, advanced join syntax supports joins based on inequalities, joins in a set or range of fields or values, and joins between a foreign key and a constant. Available in JasperReports Server 6.0 and later.

- advanced joins syntax in schema version 1.2 – In JasperReports Server 6.1.1, elements and attributes were added to give you influence over which join paths are chosen.

Advanced joins can be created only with XML. They are not supported in the Domain Designer. However, the Domain Designer will not prevent you from opening Domains with advanced joins. If you have a Domain that uses advanced joins, and you open it in the Domain Designer and then save it, your advanced joins will be lost.

You cannot combine basic and advanced syntax in a single data island. Each data island must use either all basic syntax or all advanced syntax for their joins. We recommend using the same join syntax for your entire Domain.

If you use the `jdbcQuery` element to define derived tables, you may have additional data islands in your Domain.

The Domain Designer automatically exposes all columns of all tables in a join, but in the design file you need to specify only those you want to reference elsewhere in the Domain.

7.2.4.1 Top-level Syntax for Joins

Both types of joins use the same tags to specify the tables and fields used in the join and the output table for the join results.

- `jdbcTable` – Represents the results of one or more joins between tables. If not all tables are joined together, there is one `jdbcTable` representing each join tree and containing only the join expressions for that tree. To define the join, the attributes and elements have different meanings than for a regular table.
  - `datasourceId` – Alias that identifies the data source for the Domain. The alias designates the data source where the join is performed. In the design file, this alias must be identical to that for all tables and derived tables.
  - `id` – ID used to reference the join results in the Domain design. In the Domain Designer, each join tree is automatically given the ID `JoinTree_n`, where `n` is a sequential number. In the design file, you can give the join any name, as long as it is unique among all tables and derived tables.
  - `tableName` – Literal name of the first table in the join. This table name is combined with those in the `join`, `joinInfo` and `joinedDataSetList` to define the `join` expressions.
    - For PostgreSQL and Oracle-based data sources, the table name includes the database schema name in the form `schema.table`, if you opted to include schema names when exporting the design.
    - For PostgreSQL and Oracle-based data sources within a virtual data source, the table name includes the data source prefix defined when the virtual data source was created and the database schema name in the form `dataSourcePrefix_schema.table`, if you opted to include schema names when exporting the design. The table name includes the data source prefix in the form `dataSourcePrefix.table` if you opted not to include schema names when exporting the design.
    - For MySQL and similar data sources within a virtual domain, the table name includes the data source prefix and the database name in the form `dataSourcePrefix_database.table`.
- `fieldList` – A required container for the `field` elements in the join tree.
- `field` – Represents a column in the join tree. The table of each column is identified by a prefix on the `id` attribute. When created in the Domain Designer, the design file includes every column in every table of the join. When you create your own design file, only the columns you want to reference are needed. Both attributes of `field` are required:
  - `id` – Field ID composed of the ID of the table in the design and the literal name of the column in the data source. The syntax is `table_ID.field_name`.

  - `type` – The Java type of the column, identical to the type in its table definition.
- `joinInfo` – Gives the table ID and alias for the table given by the `tableName` attribute. The table ID and alias are used as the first table in the join definition. This element and its two attributes are required even if they are identical.
  - `alias` – Alternative name within the join expression for the table identified in `referenceId`. By default, the alias is the same as the `referenceID`. If you use a distinct alias, you must be careful to use the alias throughout the `joinString` element that defines the join.
  - `referenceId` – Table `id` from the `<jdbcTable>` element of the table within the design whose data source name is given in `tableName`.

7.2.4.2 Basic Join Syntax

The basic join syntax supports joins based on equality comparisons only. This is the format produced by the Domain Designer. This format uses the `joinInfo` and `joinedDataSetList` elements to define the actual join. It also contains a list of the columns exposed through the Domain, each with a prefix on the field `id` attribute to identify its originating table. The basic join syntax is available in all versions of JasperReports Server that use Domains.

Table 7-1 Example of Basic Joins in XML

```
<jdbcTable datasourceId="SugarCRMDataSource" id="JoinTree_1" tableName="accounts">
 <fieldList>
  <field id="accounts_opportunities.account_id" type="java.lang.String"/>
  ...
 </fieldList>
 <joinInfo alias="accounts" referenceId="accounts"/>
 <joinedDataSetList>
  <joinedDataSetRef>
   <joinString>join accounts_opportunities accounts_opportunities on
    (accounts.id == accounts_opportunities.account_id)</joinString>
  </joinedDataSetRef>
  <joinedDataSetRef><joinString>join opportunities opportunities on
    (accounts_opportunities.opportunity_id == opportunities.id)</joinString>
  </joinedDataSetRef>

  <joinedDataSetRef><joinString>join users1 users1 on
    (opportunities.assigned_user_id == users1.id)</joinString>
  </joinedDataSetRef>
  <joinedDataSetRef><joinString>left outer join p1cases p1cases on
    (accounts.id == p1cases.account_id)</joinString>
  </joinedDataSetRef>
  <joinedDataSetRef><joinString>right outer join users2 users2 on
    (users2.id == p1cases.assigned_user_id)</joinString>
  </joinedDataSetRef>
 </joinedDataSetList>
</jdbcTable>
```

- `joinedDataSetList` – Container for the list of join statements.
- `joinedDataSetRef` – Container for the join statement.
- `joinString` – A string expressing a DomEL join statement in the following format:

  join_type `join` *table_ID table_alias* `on` *expression*

  Where:

  - join_type – One of `right outer`, `left outer`, or `full outer`. Inner join is the default if no join type is specified.
  - table_ID – The ID of a table within the design.

- table_alias – Alternative name to use for the table_ID within the join expression. By default, the alias is the same as the table_ID, but you may supply a true alias in handwritten design files.
- expression – Expression that compares the columns on which the join is made, in the form

  ```
  left_table_alias.field_name == right_table_alias.field_name
  ```

The order of joinedDataSetRef elements is important. The first one must contain a join expression between the table_alias it defines and the alias in the joinInfo element. The subsequent ones may reference only the table_alias they define and ones that appear in joinString elements before them.

7.2.4.3 Advanced Joins in XML

In JasperReports Server 6.0, a new join syntax was added to the XML in the design file. This syntax uses a joinList element instead of joinedDataSetList. It supports all joins supported by the basic join syntax along with the following additional types of joins:

- comparison operators – in addition to equal to (==), the advanced syntax supports less than (&lt;), less than or equal to (&lt;=), greater than (&gt;), greater than or equal to (gt;=), and not equal to (!=).

  You have to use the character entities for less than (&lt;) and greater than (&gt;)

- Boolean operators – AND and NOT.
- IN operator with support for sets and ranges.
- joins between a table foreign key and a constant value.

Examples of each syntax type are given below.

Advanced joins can only be created using XML in the design file. They are not supported in the Domain Designer. If you have a Domain that uses advanced joins, and you open it in the Domain Designer and then save it, your advanced joins will be lost.

When you add or modify joins in the Domain design file, make sure that all tables in a join element are actually connected. If you include a table that is not actually joined to any other tables, the unjoined table will be included in any Ad Hoc view that uses that data island. In this case, the fields in the unjoined table will show up in the Ad Hoc view. You will receive errors in your Ad Hoc view when you add unconnected fields from a poorly-configured join.

The following table shows how to express the basic joins in the advanced join syntax. This syntax gives the same join structure as in **Table 7-1, "Example of Basic Joins in XML," on page 255**

```
<jdbcTable datasourceId="SugarCRMDataSource" id="JoinTree_1" tableName="accounts">
   <tableRefList>
      <tableRef tableId="accounts_opportunities" alias="accounts_opportunities"/>
      <tableRef tableId="opportunities" alias="opportunities"/>
      <tableRef tableId="p1cases" alias="p1cases"/>
      <tableRef tableId="users1" alias="users1"/>
      <tableRef tableId="users2" alias="users2"/>
   </tableRefList>
 <fieldList>
  <field id="accounts_opportunities.account_id" type="java.lang.String"/>
  ...
 </fieldList>
 <joinInfo alias="accounts" referenceId="accounts"/>
```

```
 <joinList>
   <join left="accounts" right="accounts_opportunities" type="inner"
     expr="(accounts.id == accounts_opportunities.account_id)"/>
   <join left="accounts_opportunities" right="opportunities" type="inner"
     expr="(accounts_opportunities.opportunity_id == opportunities.id)"/>

   <join left="opportunities" right="users1" type="inner"
     expr="(opportunities.assigned_user_id == users1.id)"/>
   <join left="accounts" right="p1cases" type="leftOuter"
     expr="(accounts.id == p1cases.account_id)"/>
   <join left="p1cases" right="users2" type="rightOuter"
     expr="(users2.id == p1cases.assigned_user_id)"/>
 </joinList>
</jdbcTable>
```

- joinOptions element – Child element of jdbcTable. Allows you to influence the joins that are pushed down to SQL in an Ad Hoc view. A jdbcTable element can contain only one joinOptions element. joinOptions has the following attributes:
  - suppressCircularJoins (Boolean; default = false) – When this attribute is set to true for a join set/data island, an Ad Hoc view created from the data island uses a minimum join. When suppressCircularJoins is set to true, you can optionally assign weights to the joins in the join set. For most situations, the best practice is to set this option to true to avoid circular joins. Set it to false only if you need backwards compatibility with schema_1_0.xsd.
  - includeAllDataIslandJoins (Boolean; default = false) – Available for backwards compatibility. When this attribute is set to true for a join set/data island, an Ad Hoc view created from the data island includes all joins specified for that data island in the Domain. For most situations, the best practice is to set this option to false (default).
- <tableRefList> – Container for a list of tables and aliases used in the join. Must include a tableRef tag for every table used, except the table in the joinInfo tag.
  - <tableRef> – A table and its alias.
    - table – The ID of a table within the design.
    - tableAlias – Alternative name for the table used within the join expression.
    - alwaysIncludeTable (version 1.2 of the schema only) – Include this table in all Ad Hoc views created from this data island.
- <joinList> – Container for the join statement. The left join in the first join in the joinList must be the table given by the tableName attribute of jdbcTable.
  - <join> – A string expressing an SQL join statement. The following attributes are required; an additional optional weight attribute was added in version 1.2 of the schema:

    left="*left_table_ID*" right=*right_table_ID*" type="*join_type*" expr="*expression*"

  Where:

  - left_table_ID – The first table in the join definition. This table must have been declared in the jdbcTable element, either in the <joinInfo> element or in another join element that precedes it in the joinList.
  - right_table_ID – The second table in the join definition.
  - join_type – One of inner, rightOuter, leftOuter, or fullOuter.
  - expression – Expression that compares the columns on which the join is made. Each column used in the expression must come from the two tables in the current join element. When two columns are compared directly – for example, public_store.store_id == public_employee.store_id – the first column must come from the left table and the second column must come from the right table.

- The following expressions are supported:
  - Boolean operators – AND, OR, and NOT. See the other expressions for examples of how these are used.
  - comparison operators – Supports equal to (==), less than (&lt;), less than or equal to (&lt;=), greater than (&gt;), greater than or equal to (gt;=), and not equal to (!=). Operators other than == must be used in conjunction with ==. For example:

    ```
    expr="(public_store.store_id == public_employee.store_id) AND
    (public_store.coffee_bar != public_store.salad_bar)"
    ```

    ```
    expr="(public_employee.employee_id == public_salary.employee_id) AND
     (public_salary.overtime_paid &gt; 2) AND (public_salary.overtime_paid &lt;=
    2.1)"
    ```

  - IN operator – Must be used in conjunction with ==. Supports the following:
    - a set of strings or values, separated by commas. Strings are enclosed in single quotes. For example:

    ```
    expr="(public_store.region_id == public_region.region_id) AND (public_
    store.store_city IN ('San Francisco','Portland', 'Seattle'))"
    ```

    ```
    expr="(public_store.store_id == public_employee.store_id) AND (NOT public_
    store.coffee_bar IN ('true'))"
    ```

    - a range of values, separated by a colon. For example:

    ```
    expr="(public_employee.employee_id == public_salary.employee_id) AND (NOT
    public_employee.salary IN (7000 : 8000))"
    ```

  - joins between a table foreign key and a constant value. Must be used in conjunction with ==. For example:

    ```
    expr="(public_employee.employee_id == public_salary.employee_id) AND (public_
    employee.salary == 5000)"
    ```

- `weight` (optional; schema_1_2 only) – Boolean that specifies a weight to use when calculating the minimum join path.

  If you are using a field of type Date with an Oracle database, use the JasperReports Server `Date()` function in your join expression. For example:

  ```
  <join left="FOODMART_STORE" right="FOODMART_EMPLOYEE" type="inner"
  expr="(FOODMART_STORE.STORE_ID == FOODMART_EMPLOYEE.STORE_ID)
  AND (FOODMART_EMPLOYEE.BIRTH_DATE &gt; Date('1914-02-02'))"/>
  ```

  This applies to Date literals only; it is not necessary for fields with date formats but other field types, for example, Timestamp.

7.2.4.4 Working With Advanced Joins

Keep the following in mind when working with advanced join syntax:

- Advanced joins are not supported in the Domain Designer. If you have a Domain that uses advanced joins, and you open it in the Domain Designer and then save it, your advanced joins will be lost.
- You cannot combine advanced joins and basic joins in the same data island (`jdbcTabl>` tag). We recommend that you do not combine advanced joins and basic joins in the same Domain.
- If you are using advanced joins with version 1.0 of the schema, the number of joins must be less than the number of tables within each `jdbcTable` tag. That is, if you have N tables, use at most N-1 joins. In JasperReports Server 6.1.1, additional elements were added to help you influence join paths when you have more than N-1 joins. See **7.2.5, "Minimum Paths and Circular Joins in Version 2.0 of the Schema," on page 259** for more information.

- When you add or modify joins in the Domain design file, make sure that all tables in a join element are actually connected. If you include a table that is not actually joined to any other tables, the unjoined table will be included in any Ad Hoc view that uses that data island. In this case, the fields in the unjoined table will show up in the Ad Hoc view. You will receive errors in your Ad Hoc view when you add unconnected fields from a poorly-configured join.
- When you use a `join` tag, the `left` table must have been defined previously. You can define the table either in the `joinInfo` tag or as the `right` table in a previous join in the same `joinList`. Follow these guidelines for your joins:
  - When you have a `join` that depends on the `right` table from another join in the `joinList`, make sure that the dependent `join` appears after the join it depends on.
  - When you have a `join` that refers to a table that has not yet been defined, make sure that the new table is referenced as the `right` table in the join.
- In some cases, you can start your Domain creation using the Domain Designer. To do this:
  a. Set up the rest of your Domain, such as derived tables and calculated fields, in the Domain Designer.
  b. Create a join that uses the same tables that you want to join in your Domain. This will set up the `jdbcTable` and `fieldList` tags for you.
  c. Export the schema from the Domain Designer.
  d. Manually edit the `tableRefList` and `joinInfo` sections of the exported design file to create the join you want.

7.2.5 Minimum Paths and Circular Joins in Version 2.0 of the Schema

A new schema, schema_1_2.xsd, added in JasperReports Server 6.1.1 includes syntax that gives you influence over which join paths are chosen when multiple options are possible. These attributes are primarily used to avoid circular joins. To use these options, your Domain design file must reference the updated schema, schema_1_2.xsd, and you must use the advanced join syntax.

schema_1_2.xsd supports the original basic and advanced join syntax. In addition, the following new elements and properties have been added to the advanced syntax:

- `joinOptions` element – Child element of `jdbcTable`. Allows you to influence the joins that are pushed down to SQL in an Ad Hoc view. A `jdbcTable` element can contain only one `joinOptions` element. `joinOptions` has the following attributes:
  - `suppressCircularJoins` (Boolean; default = `false`) – When this attribute is set to `true` for a join set/data island, an Ad Hoc view created from the data island uses a minimum join. When `suppressCircularJoins` is set to `true`, you can optionally assign weights to the joins in the join set. For most situations, the best practice is to set this option to `true` to avoid circular joins. Set it to `false` only if you need backwards compatibility with schema_1_0.xsd.
  - `includeAllDataIslandJoins` (Boolean; default = `false`) – Available for backwards compatibility. When this attribute is set to `true` for a join set/data island, an Ad Hoc view created from the data island includes all joins specified for that data island in the Domain. For most situations, the best practice is to set this option to `false` (default).
- `weight` attribute (positive integer; default = `1`) – New attribute for the `join` element; only available when `suppressCircularJoins` is set to `true`. You can assign a weight to one, several, or all of the `join` elements in a join set that has `suppressCircularJoins` enabled. Higher weights indicate costlier, less-desirable joins.
- `alwaysIncludeTable` attribute (Boolean; default=`false`) – Available for backwards compatibility. Attribute of `tableRef`. When this attribute is set to `true`, the table is included in all Ad Hoc views that use this data island. For most situations, the best practice is to set this option to `false` (default).

7.2.5.1 Understanding Circular Joins

The Domain Designer and Domain Design file do not impose any limits on the number of joins you can specify for a given number of tables. However, when you define joins among N tables, it is good design to use N-1 or fewer joins, in order to avoid circular joins.

Circular joins occur, for example, when table A is joined to table B and table B is joined to table C which is in turn joined to table A.

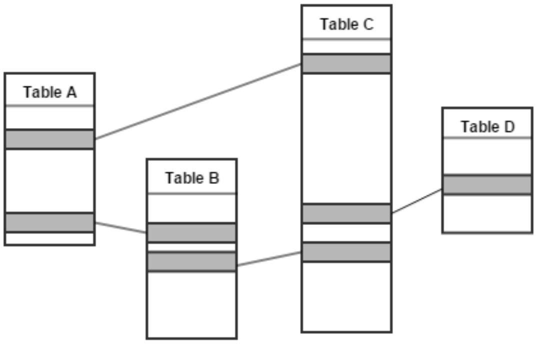


Figure 7-2 Circular Joins in a Domain

Suppose you create an Ad Hoc view using table A and table B from such a Domain. JasperReports Server attempts to avoid circular joins in the SQL by choosing a path between the tables. However, if you are using schema_1_0.xsd, you cannot be sure which join path the Ad Hoc view will use: the direct join from A–B or the indirect join A–C–B.

In this situation, you can use `joinOptions` with the `suppressCircularJoins` set to `true`. What this does is assign a default weight of 1 to each join, as shown in the following simplified diagram.

The best practice is to avoid circular joins in your Domain by creating aliases for one or more tables in the join. For an example of creating a table alias, see **step 12**in **6.3, “Example of Creating a Domain,” on page 204**.

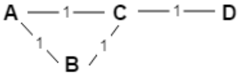


Figure 7-3 Circular Join with Default Weights

Now, when you create an Ad Hoc view using table A and table B, the Ad Hoc Designer calculates the weight of the possible join paths from A to B by summing the weights of their subpaths. The direct path A–B has total weight=1, while the indirect path A–C–B has weight 1+1=2. The path with the smallest total weight is chosen, in this case, the direct path from A to B.

Note, however, that this does not cover all cases. For example, suppose you have four joins as shown in the following table, where table A is joined to tables C and D, and table B is also joined to tables C and D. If you create an Ad Hoc view with tables A and B, both possible paths A–C–B and A–D–B have weight 1+1=2, so again, you do not know which join is being used.

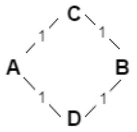


Figure 7-4 Circular Joins with Equal Weights Between A and B

To exert even more control over the joins used by your Ad Hoc view, you can add an optional `weight` to one or more of the `join` elements when you are using `suppressCircularJoins` set to `true`. Adding a weight increases the cost of a particular join, and reduces its desirability. The higher you set a weight the more you dislike the join. For example, you can add a weight of 2 to the join between table B and table D. Then you can ensure that an Ad Hoc view with tables A and B will use the A–C–B join path, which only has a total weight of 2, and not the A–D–B join path, which now has a total weight of 3. Note that, in this situation, an Ad Hoc view with tables B and D will still use the direct path with weight 2, instead of the path B–C–A–D, which has a total weight of 3.

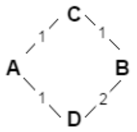


Figure 7-5 Circular Joins with Custom Weight

7.2.5.2 suppressCircularJoins Syntax

The following example shows how you might use `suppressCircularJoins` in a Domain file, giving the second join in the list a `weight` of 10:

```
<schema xmlns="http://www.jaspersoft.com/2007/SL/XMLSchema" version="1.2">

<jdbcTable id="JoinTree_1" datasourceId="FoodmartDataSourceJNDI" tableName="public.customer">
  <fieldList>
    <field id="public_customer.account_num" fieldDBName="account_num" type="java.lang.Long" />
    ...
  <fieldList>
<joinInfo alias="public_store" referenceId="public_store" />
<joinOptions suppressCircularJoins="true"/>

        <tableRefList>
          <tableRef tableId="public_store" tableAlias="public_store"/>
          <tableRef tableId="public_customer" tableAlias="public_customer"/>
          <tableRef tableId="public_region" tableAlias="public_region"/>
        </tableRefList>

       <joinList>
         <join left="public_store" right="public_customer" type="inner"
                      expr="public_store.region_id == public_customer.customer_region_id"/>
         <join weight="10" left="public_store" right="public_region" type="inner"
                      expr="public_store.region_id == public_region.region_id"/>
         <join left="public_region" right="public_customer" type="inner"
                      expr="public_region.region_id == public_customer.customer_region_id"/>
    </joinList>

</jdbcTable>
```

7.2.5.3 alwaysIncludeTable

The alwaysIncludeTable property of the tableRef element lets you specify that any Ad Hoc view created from your Domain must include the indicated table. You can use this property whether suppressCircularJoins is true or false.

For example, suppose you have four joins for four tables, where table A is joined to tables C and D, and table B is also joined to tables C and D and you create an Ad Hoc view using tables A and B. As mentioned above, there are two possible join paths, A–C–B and A–D–B. If you set the alwaysIncludeTable property of table D to true and suppressCircularJoins is false or unspecified, the Ad Hoc view will use the A–D–B path.

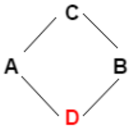


Figure 7-6 Including a Table

When suppressCircularJoins is true, you can use both the alwaysIncludeTable property and the weight property to further control the join path. For example, suppose instead of the default weights, you set the weight on the join between A and D to 100. In this case, the join path with the minimum weight is actually the join that goes from A–C–B–D, avoiding the join from A to D.

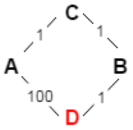


Figure 7-7 Using alwaysIncludeTable and Weighted Joins

The following example shows how you might set alwaysIncludeTable as well as a join weight:

```
<schema xmlns="http://www.jaspersoft.com/2007/SL/XMLSchema" version="1.2">

<jdbcTable id="JoinTree_1" datasourceId="FoodmartDataSourceJNDI" tableName="public.customer">
  <fieldList>
    <field id="public_customer.account_num" fieldDBName="account_num" type="java.lang.Long" />
    ...
  <fieldList>
       <joinInfo alias="public_store" referenceId="public_store" />
       <joinOptions suppressCircularJoins="true"/>

        <tableRefList>
          <tableRef tableId="public_store" tableAlias="public_store"/>
          <tableRef tableId="public_customer" tableAlias="public_customer"/>
          <tableRef alwaysIncludeTable="true" tableId="public_region" tableAlias="public_region"/>
        </tableRefList>

       <joinList>
         <join left="public_store" right="public_customer" type="inner"
                         expr="public_store.region_id == public_customer.customer_region_id"/>
         <join left="public_store" right="public_region" type="inner"
                         expr="public_store.region_id == public_region.region_id"/>
        <join weight="10" left="public_region" right="public_customer" type="inner"
                      expr="public_region.region_id == public_customer.customer_region_id"/>
      </joinList>

</jdbcTable>
```

7.2.5.4 includeAllDataIslandJoins

The `includeAllDataIslandJoins` property of the `joinOptions` element lets you specify that any Ad Hoc view created from your Domain must include all joins in your data island. This option can only be set to `true` when `suppressCircularJoins` is `false`.

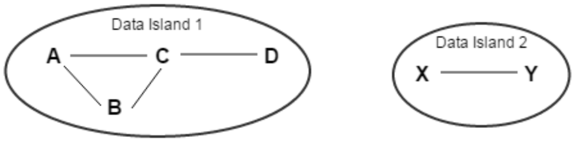


Figure 7-8 Joins in a Data Island

In the figure above, if you set `includeAllDataIslandJoins` to `true` for Data Island 1, then an Ad Hoc view containing tables A and B will include all joins between tables A, B, C and D. However, the join between X and Y is in a different data island, and is not included.

The following example shows how to use `includeAllDataIslandJoins` in a Domain file:

```
<joinOptions suppressCircularJoins="false" includeAllDataIslandJoins="true"/>
```

You can include at most one `joinOptions` element in a specific `jdbcTable` element.

7.2.5.5 Tips for Using Version 2.0 Joins

- To use the features in version 2.0, set your domain schema version to 1.2 and use the `<joinsOptions>` syntax and the `<tableRef>` element.
- To avoid join cycles in queries set `<joinOptions suppressCircularJoins=true/>`. This will find the lowest weight join set.
- To configure your Domain to use the fewest number of joins required to reach a given set of chosen fields, use `<joinOptions suppressCircularJoins=true/>` AND set all `<join weight="W">` values to be the same value. In this case, the lowest weight join set will be a set that has the minimum possible number of joins. You do not have to set the join weight explicitly. If the weight is not set, it defaults to 1.
- To give preferential treatment to joins that involve indexed columns for better join performance, assign lower join weights to the preferred joins and higher join weights to less desirable joins.
- To make sure that one or more specific table(s) always get included in your results, set `alwaysIncludeTable=true`. For example, to make sure tableA is always included, use

  ```
  <tableRef tableId=tableA tableAlias=tableA alwaysIncludeTable=true/>
  ```

7.2.6 Representing Calculated Fields in XML

Calculated fields are defined as regular columns in a `field` element with an additional attribute. Calculated fields that rely only on columns of the same table appear in `jdbcTable` for that table, as well as in the join tree.

Calculated fields that rely on columns from different tables that are joined appear only in the join tree.

The following example shows the XML for a calculated expression in the `accounts` table. Because it references only the columns of `accounts`, it appears in that table and in the join tree.

```
<jdbcTable datasourceId="SugarCRMDataSource" id="accounts" tableName="accounts">
 <fieldList>
  ...
  <field dataSetExpression="concat( billing_address_city, ', ',
      billing_address_state )" id="city_and_state" type="java.lang.String"/>
  ...
 </fieldlist>
</jdbcTable>
...
<jdbcTable datasourceId="SugarCRMDataSource" id="JoinTree_1" tableName="anything">
 <fieldList>
  ...
  <field dataSetExpression="concat( accounts.billing_address_city, ', ',
      accounts.billing_address_state )" id="accounts.city_and_state"
      type="java.lang.String"/>
  ...
 </fieldlist>
</jdbcTable>
```

The attributes of the `field` element have a different meaning when defining a calculated field:

- `dataSetExpression` – Expression that calculates a value based on other columns. The syntax for the expression, including how to reference columns, is documented in **"The DomEL Syntax" on page 268**.
- `id` – User-defined name of the calculated field. The format of the `id` is dependent on how the calculated field appears in the design file:

1. If the expression references columns in the same table:
    a. The field appears in the table and the `id` is a simple column name.
    b. The field also appears in a join tree that uses the table, the `id` has the form `table_ID.field_name`.
2. When the expression references columns in different tables, the field appears only in the join tree of those tables, and the `id` has the form `jointree_ID.field_name`.
3. When the expression computes a constant value. The field appears in a table named `Constant`, and the `id` is a simple column name. Constant fields are further explained below.

- `type` – The Java type of the value calculated by the expression, for example `java.lang.String`. This type must be compatible with the result of the DomEL expression and among the JDBC-compatible types listed in **7.2.2, "Representing Tables in XML," on page 251**.

A special case of a calculated field occurs when the expression does not reference any column names. The calculated field always has the same value and is said to be a constant. In the Domain Designer, constant fields are automatically grouped in a table named `Constant` and may be used in other calculated fields, filters, or even as an item. Because constant fields are not dependent on any column values, they may be used in any join tree and exposed to the user along with the items from any join tree. When editing a design file, you must treat constant calculated fields in the same way.

7.2.7 Representing Filters in XML

Filters are defined as optional `filterString` elements inside of `jdbcTable` and `jdbcQuery` elements. They impose a condition on any results returned for that table, query, or join tree, thereby limiting the number of rows returned when accessing the data source. Whereas other settings mainly determine which columns are available for use in a report. A filter determines which rows are available when running the report.

- `filterString` – Expression that evaluates to true or false when applied to each row of values in the data source. The expression refers to columns using their `id` attribute. Thus, a filter on a table or derived table refers to the simple column name, but a filter on a join tree refers to the `table_ID.field_name`. The full syntax for the expression is documented in **"The DomEL Syntax" on page 268**.

Filters defined in the Domain Designer are limited to conditions on one column or comparisons of two columns, with more complex filters created by the conjunction (logical AND) of several conditions. Other filter expressions are not supported. You can upload a design file with more complex filters, but they are overwritten or cause errors if you open the design in the Domain Designer.

For example, the following filters are defined in the example Domain on **page 211**.

```
<jdbcTable datasourceId="SugarCRMDataSource" id="opportunities"
                                        tableName="opportunities">
       <fieldList>
               ...
               <field id="opportunity_type" type="java.lang.String"/>
       </fieldList>
       <filterString>opportunity_type == 'Existing Business'</filterString>
</jdbcTable>
<jdbcQuery datasourceId="SugarCRMDataSource" id="p1cases">
       <fieldList>
               ...
               <field id="status" type="java.lang.String"/>
       </fieldList>
       <filterString>status != 'closed'</filterString>
       <query>...</query>

</jdbcQuery>
<jdbcTable datasourceId="SugarCRMDataSource" id="opportunities"
                                        tableName="opportunities">
       <fieldList>
               ...
               <field id="opportunity_type" type="java.lang.String"/>
       </fieldList>
       <filterString>opportunity_type == 'Existing Business'</filterString>
</jdbcTable>
<jdbcQuery datasourceId="SugarCRMDataSource" id="p1cases">
       <fieldList>
               ...
               <field id="status" type="java.lang.String"/>
       </fieldList>
       <filterString>status != 'closed'</filterString>
       <query>...</query>
</jdbcQuery>
```

7.2.8 Representing Sets and Items in XML

Now that all the table and field IDs have been defined, look at the definitions of sets and items exposed through `itemGroups` and `items` elements at the top of the Domain design file. The `itemGroups` and `items` elements are equivalent to the selection of sets and items on the Display tab of the Domain Designer. They create a hierarchy of sets, subsets and items and hold attributes that define all the properties available on sets and items. For a description of each possible property, see **“The Properties Panel” on page 232**. The following example shows two levels of sets, with items inside each level and at the root, outside of any set.

```
<itemGroups>
       <itemGroup label="outerset" ...  >
<itemGroups>
       <itemGroup label="innerset" ... >
       <items>
               <item label="innersetitem1" ... />
               <item label="innersetitem2" ... />
       </items>
       </itemGroup>

</itemGroups>
       <items>
               <item label="outersetitem1" ... />
               <item label="outersetitem2" ... />
       </items>
       </itemGroup>
</itemGroups>
       <items>
               <item label="outsideitem1" ... />
               <item label="outsideitem2" ... />
       </items>
       <items>
               <item label="outersetitem1" ... />
               <item label="outersetitem2" ... />
       </items>
       </itemGroup>
</itemGroups>
<items>
       <item label="outsideitem1" ... />
       <item label="outsideitem2" ... />
</items>
```

- `itemGroups` – A container for `itemGroup` elements.
- `itemGroup` – Represents a set. The `itemGroup` element may contain an `itemGroups` element, an `items` element, or both, representing its subsets and items, respectively. The attributes of `itemGroup` are the properties of the set it represents:
  - `id` – The unique identifier of the set among all set and item IDs. This attribute is required.
  - `label` – The set’s name, visible to users of the Domain. If the label is missing, the Ad Hoc Editor displays the `id`.
  - `description` – The optional description of the set, visible to users as a tooltip on the set name in the Ad Hoc Editor.
  - `labelId` – The internationalization key for the label in the Domain’s locale bundles.
  - `descriptionId` – The internationalization key for the description in the Domain’s locale bundles.
  - `resourceId` – A reference to the table on which the set is based. This attribute is required, but it has no meaning on a set and is not significant in the design.

  When an internationalization key is defined for the label or description, the label or description is replaced with the value given by the key in the local bundle corresponding to the user’s locale. See **“Locale Bundles” on page 277**.
- `items` – A container for `item` elements.
- `item` – Represents an item. The attributes of `item` are the properties of the item it represents:
  - `id` – The unique identifier of the item among all set and item IDs. This attribute is required.

- `label` – The item's name, visible to users. If the label is missing, the Ad Hoc Editor displays the `id`.
- `description` – The optional description of the item, visible as a tooltip on the item name in the Ad Hoc Editor.
- `labelId` – The internationalization key for the label in the Domain's locale bundles.
- `descriptionId` – The internationalization key for the description in the Domain's locale bundles.
- `resourceId` – A reference to the column on which the item is based. This attribute is required because it defines the connection between what the user sees and the corresponding data in the data source. The `resourceId` has the form `table_ID.field_ID`. When the item refers to a column in a join tree, the `resourceID` corresponds to `jointree_ID.table_ID.field_name` because the field ID in a join tree includes the table ID.
- `dimensionOrMeasure` – Corresponds to the **Field or measure** setting in the user interface. Its possible values are `Dimension` (equivalent to field) or `Measure`. By default, all numeric fields are treated as measures in the Ad Hoc Editor. This attribute is optional and necessary only when overriding the default behavior, for example to make a numeric item explicitly not a measure. See **"Terminology" on page 203**.
- `defaultMask` – A representation of the default data format to use when this item is included in a report. The possible values depend on the `type` attribute of the column referenced by the `resourceId`. See the table below.
- `defaultAgg` – The name of the default summary function (also called aggregation) to use when this item is included in a report. The possible values for the `defaultAgg` depend on the `type` attribute of the column referenced by the `resourceId`. The following table gives the possible data formats and summary functions based on the column type. The appearance columns show the equivalent setting in the properties table of the Display tab, as shown in the following table.

Field Type	Default Data Formats		Default Summary Functions	
	Attribute Value	Appearance	Attribute Value	Appearance

Integer	`#,##0`
	`0`
	`$#,##0;($#,##0)`
	`#,##0;(#,##0)`
		`-1,234`
		`-1234`
		`($1,234)`
		`(1234)`
			`Highest`
			`Lowest`
			`Average`
			`Sum`
			`DistinctCount`
			`Count`
				Maximum
				Minimum
				Average
				Sum
				Distinct Count
				Count All

Double	`#,##0.00`
	`0`
	`$#,##0.00;($#,##0.00)`
	`$#,##0;($#,##0)`
		-1,234.56
		-1234
		($1,234.56)
		($1,234)		

Date	`short` hide medium hide long medium	3/31/09 Mar 31, 2009 March 31, 2009 23:59:59	`DistinctCount` `Count`	Distinct Count Count All
All others	*Not allowed*			

The following example shows the use of the `itemGroup` and `item` elements to represent the sets and items from **"Example of Creating a Domain" on page 204**. The design file was exported from the Domain Designer.

```
<itemGroups>
 ...
 <itemGroup id="users1" label="Account Rep" description="Primary account
       representative" labelId="" descriptionId="" resourceId="JoinTree_1">
  <items>
   <item id="first_name" label="First Name" description="Given name"
      labelId="" descriptionId="" resourceId="JoinTree_1.users1.first_name"/>
   <item id="last_name" label="Last Name" description="Surname or family name"
      labelId="" descriptionId="" resourceId="JoinTree_1.users1.last_name"/>
  </items>
 </itemGroup>
 <itemGroup id="opportunities" label="Opportunity" description="Sales opportunity"
       labelId="" descriptionId="" resourceId="JoinTree_1">
  <items>
   <item id="date_entered1" label="Date" description="Date opportunity opened"
      labelId="" descriptionId="" defaultMask="short,hide"
      resourceId="JoinTree_1.opportunities.date_entered"/>
   <item id="amount" label="Amount" description="Estimated contract Amount"
      labelId="" descriptionId="" defaultMask="$#,##0;($#,##0)"
      defaultAgg="Average" resourceId="JoinTree_1.opportunities.amount"/>
   <item id="probability" label="Probability" description="Chance of closing the
      contract" labelId="" descriptionId=""
      resourceId="JoinTree_1.opportunities.probability"/>
```

Labels and descriptions may contain any characters, but the ID property value of both `itemGroup` and `item` elements must be alphanumeric and not start with a digit.

7.3 The DomEL Syntax

Various components of a Domain need to compute values based on some expression involving constants, field values, and environment variables. The Domain Expression Language (DomEL) was created to fill this need. Currently, the following features in XML design files are expressed in DomEL:

- The `on` and `where` clauses of derived tables
- The `on` clause of join statements
- Calculated fields
- Filter expressions in Domains and Domain topics (equivalent to `where` clauses)
- Access to attribute values
- Row-level security (see **7.5, "The Domain Security File," on page 276**)

A DomEL expression is a shorthand way of writing a complex query. When processing a report based on a Domain, the server interprets DomEL expressions to generate parts of the SQL expression that perform the desired query. Depending on the data policy, either the augmented SQL is passed to the data source, or the server performs a simpler query and applies the DomEL expressions to the full dataset in memory.

7.3.1 Datatypes

The following simple datatypes may be declared as constants, used in expressions, and returned as values:

Simple Type	Description	Example of Constant
boolean	Expressions such as comparison operators return boolean values, but `true` and `false` constants are undefined and cannot be used.	none
integer	Whole numbers.	`123` or `-123`
decimal	Floating point numbers. Decimal separator must be a period (`.`); other separators such as comma (`,`) are not supported.	`123.45` or `-123.45`
string	Character string entered with single quotes (`'`); double quotes (`"`) are not supported.	'hello world'
date	ANSI standard date.	`d'2009-03-31'` or `Date('2009-03-31')`
timestamp	ANSI standard date and time.	`ts'2009-03-31` 23:59:59`'` or TimeStamp('2009-03-31 23:59:59')

The composite datatypes, shown in the following table, may be declared as constants and used with the `in` *set* or `in` *range* operator. The values in these composite types are not necessarily constant, they could be determined by field values.

Composite Type	Description	Example
set	Contains any number of any simple type above.	`(1, 2, 3)('apples','oranges')`
range	Inclusive range applicable to numbers and dates, including fields that are number or date types.	`(0:12)` or `(0.0:12.34)(d'2009-01-01':d'2009-12-31')(limit_min:limit_max)`

7.3.2 Field References

DomEL expressions are stored in the Domain design and interpreted when the server prepares to run a query to retrieve data from the data source. Therefore, all references to field values in an expression are based on the IDs given in the Domain design. Field references have the following format, depending on where the expression appears:

Appears In	Field Reference	Explanation
Derived table	`table_ID.field_name`	The SQL query that defines a derived table can refer to any previously defined table or derived table in the Domain. Therefore, you must include the table ID.

Appears In	Field Reference	Explanation
Join expression	`table_alias.field_name`	Within a join expression, tables are given alias names that must be used. Even though join expressions may appear in separate `joinedDatasetRef` elements, the alias declared in each one can be used in any subsequent one.
Calculated field on a table or derived table	`field_name`	Calculated fields can only appear on a table if they refer exclusively to fields of the table, in which case no table ID is needed. However, the table ID is not forbidden, and the Domain Designer sometimes includes it.
Calculated field on a join tree	`table_ID.field_name`	Calculated fields declared in join trees refer to fields prefixed with their table ID.
Filter on a table or derived table	`field_name`	Filters that are evaluated within the table or derived table do not need the table ID.
Filter on a join tree	`table_ID.field_name`	Filters that refer to fields in separate tables of the join tree need to use the table ID on each field name.
Calculated field in an Ad Hoc view	`"Field Label"`	Calculated fields declared in Ad Hoc views can refer to fields using their Item Labels entered with double quotes (`"`). Item Labels are created on the Display tab of the Domain Designer.
	`table_ID.field_name`	Calculated fields declared in Ad Hoc views can refer to fields prefixed with their table ID.

7.3.3 Operators and Functions

DomEL provides the following operators, listed in order of precedence. Operators higher in this list are evaluated before operators lower in the list:

Operator	Syntax	Description
multiply, divide	`i * j / k`	Arithmetic operators for numeric types only.
percent	`i % j`	Calculates i as a percent of j; numeric types only.
add, subtract	`i + j - k`	Arithmetic operators for numeric types only.
equal	`i == j`	Comparison operators for string, numeric, and date types.
not equal	`i != j`	

Operator	Syntax	Description
less than	`i < j`	Comparison operators for numeric and date types only.
less than or equal	`i <= j`	
greater than	`i > j`	
greater than or equal	`i >= j`	
IN *set*	`i IN ('apples','oranges')`	Sets can be of any type.
IN *range*	`i IN (j:k)`	Ranges must be numeric or date types.
NOT	`NOT( i )`	Boolean operators. Parentheses are required for `NOT`.
AND	`i AND j AND k`	
OR	`i OR j OR k`	
parentheses	`()`	Used for grouping.

DomEL also defines the following operations as functions.

Function	Syntax	Description
startsWith	`startsWith(i, 'prefix')`	Comparison operators for strings.
endsWith	`endsWith(j, 'suffix')`	
contains	`contains(k, 'substring')`	
concat	`concat(i, ' and ', j, ...)`	Returns the string of all parameters concatenated.

Additional functions are supported in Ad Hoc views.

7.3.4 Attribute Functions

An attribute is a name-value pair associated with a user account. Attributes can also be defined on organizations or at the server level and apply to all users in the organization or server, respectively. To define or modify the attributes for a user, organization, or server, see the section on defining attributes in the *JasperReports Server Administrator Guide*.

Use the following syntax to access attribute values for the logged-in user.

```
groovy('authentication.principal.attributes.find{ it.attrName == &quot;locale&quot;
}.attrValue'))
```

An attribute value is always a string. You can use the following functions to cast it to other types, for example if you want to compare it to a field value.

Function	Syntax	Example
Integer	`Integer('string')`	Integer('345') = 345;
Decimal	`Decimal('string')`	Decimal('24.5') = 24.5;
Boolean	`Boolean('string')`	Boolean('true') = true;
Date	`Date('string')`	Date('2008-02-08') = d'2008-02-08';
Timestamp	`Timestamp('string')`	Timestamp('2008-02-08 11:30:00') = ts'2008-02-08 11:30:00';
Time	`Time('string')`	Time('11:30:00') = t'11:30:00';

For more information about using attributes in security files, see the *JasperReports Server Security Guide* and *Jaspersoft OLAP Ultimate Guide*.

7.3.5 SQL Functions

DomEL is the primary language used inside the tags in the Domain design file. You can use SQL in the following situations:

- The `query` tag in the derived tables section of the Domain design file expects an SQL function. See **Structure of the Design File** for more information.
- In addition, you may use SQL functions in a DomEL expression, but only in limited circumstances:
  - The functions must be supported by the database. See the vendor documentation for available functions and their syntax.
  - The functions must follow the convention of comma-separated parameters. For example, you can use `TRIM(person.name)`, but not `TRIM('Jr' FROM person.name)`
  - The type of the return values must be appropriate, either within the expression or for the type of the calculated field.
  - The SQL context must be appropriate for the functions. For example, you cannot use aggregation functions such as `COUNT` in a calculated field because there is no `GROUP BY` clause.

  Except for the comma-separated parameter pattern, the DomEL validation cannot enforce these criteria. You must ensure that any SQL functions meet these criteria, otherwise the expression causes errors when using the Domain to create a report.

7.3.6 The groovy() Function

Groovy is an interpreted language for the JVM. Domains and DomEL use Groovy in the following ways:

- The `<principalExpression>` tag in an access grant in the Domain Security file uses a Groovy expression to get the current authentication object and determine its access privileges, along with the user and roles associated with the object. In this case, Groovy is used directly inside the tag. See **The Domain Security File** for more information.
- You can use the DomEL `groovy()` function as part of a DomEL expression. The DomEl `groovy` function takes a single string argument that is interpreted as Groovy code. The output of the function is a string that is put into the SQL. To include Groovy, use the following syntax:

```
groovy('your groovy code here')
```

- For example, the following simple expression could be used to set the value of the calculated field `e.groovyEval` to the SQL string corresponding to the value of 5.0/6:

```
<field id="e.groovyEval" dataSetExpression="groovy('(5.0/6).toString()')"
type="java.lang.String" />
```

DomEL validation cannot check your Groovy code. You must ensure that any Groovy code meets these criteria, otherwise the expression causes errors when using the Domain to create a report.

7.3.7 Complex Expressions

Complex expressions are written by grouping any of the operators or functions above. Parentheses `()` may be used for grouping boolean operators, but arithmetic expressions that rely on parentheses are not supported. To compute complex arithmetic expressions, you may need to define several expressions as separate calculated fields, and then reference them in a simpler expression in another calculated field.

The following examples show complex expressions suitable for filters. This first one selects only stores in Western states of the US:

```
s1.store_country in ('USA') and s1.store_state in ('WA', 'OR', 'CA', 'NV')
```

The following filter expression uses a date range:

```
s1.first_opened_date in ( Date( '2000-01-01' ) : Date( '2004-12-31' )) and not(
s1.closed )
```

As shown in these examples, field values are often compared to constant values such as `'USA'`. Therefore, the author of the design file must ensure that values used in a DomEL expression exist in the data source. Otherwise, a wrong value might go undetected and impact the quality of data in reports based on the Domain. The Domain Designer determines values for comparison by accessing the data source, so if you export a design file, you can use the values it has found. Another way to reduce errors and also support future data changes is to use more general expressions such as:

```
s1.store_country in ('US', 'USA', 'United States')
```

7.3.8 Return Types

DomEL expressions are used in different contexts for different purposes. The expected return type depends on where the expression appears:

Appears In	Expected Return Type	Explanation
Derived Table	SQL query with boolean expressions	A derived table is defined by an SQL expression that contains DomEL expressions. The `join on` clause may contain boolean comparisons and the `where` clause may contain filters, both of which are described below.
Join expression	boolean	Within a join expression, the `on` clause contains a comparison of fields from each table that has a boolean result. The `on` clause may also contain other DomEL expressions logically associated with `and` or `or` to create a complex join condition.

Appears In	Expected Return Type	Explanation
Calculated field	any type	The expression must evaluate to a type that is compatible with the SQL type declared in the Domain Designer or in the design file. For example, if the declared type is `java.lang.Float`, the expression must compute a decimal value.
Filter	boolean	Filters must be true or false overall. When there are several conditions, they must be logically associated with `and` or `or` (currently, only `and` is supported in the Domain Designer).

7.4 Security and Locale Information for a Domain

You can include security and locale information by uploading the following files:

- A single security file. Defines row and column-level access to data selected by the Domain.
- Any number of locale bundles. Provides internationalized labels for the sets and items of the Domain design.

You upload and manage these files using the Edit Domain or Add New Domain pages of the Domain Designer. This section describes how to upload and replace Domain resources. For more information about the syntax of resource files, see **"The Domain Security File" on page 276** and **"Locale Bundles" on page 277**.

To upload and replace a security file and locale bundles for a Domain:

1. Log in to the server as an administrator and select **View > Search Results**.
2. Locate the Domain. Typically, you set the Domain filter in **Filters > All Types > More Choices > Domains**.
3. Right-click an existing Domain and select **Edit** from the context-menu.

   The Domain appears in the Edit Domain page of the Domain Designer.
4. Click **Add Security File** or **Add Locale Bundle** to upload a security file or locale bundle to the Domain. The respective Add Security File or Add Locale Bundle dialog box appears.

Figure 7-9 Add Security File and Add Locale Bundle Dialog Boxes

There can be only one security file but any number of locale bundles. When a security file exists, the **Add Security File** text is disabled. See **step 8** to replace an existing security file.

5. You can upload resources from local files or from objects in the repository. Select **Upload a Local File** or **Select from Repository** and then **Browse** to select a file or repository object.

   When browsing the repository, you see only XML files or locale bundle objects. To upload a security file, select an XML file created explicitly as a security file. When selected, the XML file in the repository is referenced by the Domain, unlike a local file that is uploaded directly into the Domain.

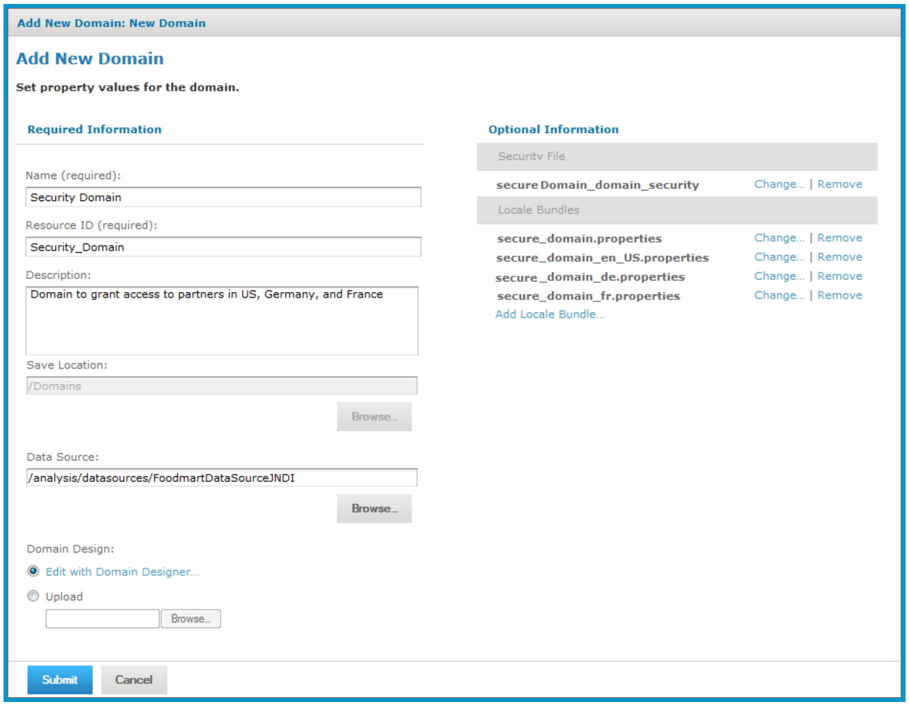
   The repository manager warns you if you attempt to delete a security or locale bundle file that is referenced by a Domain, but it does not warn you if you replace these files with a different file. When you store these files in the repository, ensure that any updates to the files are compatible with the Domains that reference them.

6. Click **Select** to upload the file and add it to the list of current resources.

   The server validates the file to make sure it matches the format of a security file or locale bundle. If the file type is not recognized or there is a syntax error, the file is not added to the list of resources and you must select another file or click **Cancel**.

7. Repeat **step 4** through **step 6** to upload one security file and any number of locale bundles.

   Optional Information in the Edit Domain page lists the referenced files.



Add New Domain: New Domain

Add New Domain

Set property values for the domain.

Required Information

Name (required): Security Domain

Resource ID (required): Security_Domain

Description: Domain to grant access to partners in US, Germany, and France

Save Location: /Domains

Data Source: /analysis/datasources/FoodmartDataSourceJNDI

Domain Design:

● Edit with Domain Designer...

○ Upload

Optional Information

Security File

secure Domain_domain_security — Change... | Remove

Locale Bundles

secure_domain.properties — Change... | Remove

secure_domain_en_US.properties — Change... | Remove

secure_domain_de.properties — Change... | Remove

secure_domain_fr.properties — Change... | Remove

Add Locale Bundle...

Figure 7-10 List of Security and Locale Files in the Edit Domain Page

8. To edit or delete the uploaded resources, click **Change** or **Remove**, respectively. To download the security file or a locale bundle file, click the name of the referenced file in the list.

   For example, you add an item to the Domain after creating the locale bundles, and you need to add its label and description keys to each bundle. Download each of the locale bundles in the Domain, edit each file to add the new keys, then upload each file to replace the corresponding locale bundle. You might also want to define access permissions for the new item by downloading, modifying, and uploading the security file as well.

9. If you modified an existing Domain, you must clear the Ad Hoc cache of all queries based on the Domain. This removes any data that was based on the old instance of the Domain and avoids inconsistencies in new reports. For instructions, see the *JasperReports Server Administrator Guide*.

7.5 The Domain Security File

The security file defines permissions to control access to Domain data on the basis of user names and roles existing in the server. When creating or running a report based on a Domain, the user name and roles are checked against the permissions in the security file. Permissions can be set separately on the data's columns and rows:

- Security on columns is defined by permissions on the sets and items of the Domain, corresponding to columns in the data source. For example, only certain users might be able to see sensitive employee information such as a Social Security Number.
- Security on rows is defined by permissions on the data values. For example a manager might be allowed to see the salary column of only employees whose manager field equals that manager’s employee number.

The IDs of tables, columns, sets and items that appear in the design are referenced by the security file. In Domains, columns display the items in the Domain; rows display the values of each item.

A user can see results only where he has both column- and row-level access.

For a given query on the data source, the security definition finds the access grants and determines access rights, determined first for item groups and items, then for resources. When the query is passed to the data source and the report is run, the grants are applied as filters on the report's columns and rows. Security defined on the physical layer applies to all content in the presentation layer. Security defined on a join applies only to the presentation layer content specific to the join.

When a user is designing a report in the Ad Hoc Editor, he sees only the columns to which he has access. When the report runs, portions to which the user has no access are blank.

All access grants for a Domain are defined in a single security file attached to the Domain as a resource, as described in **"Security and Locale Information for a Domain" on page 274**. The default access is `granted`.

When creating a security file, be sure to use the IDs of items and groups as they are defined in the Domain design file exported from the Domain Designer. For more information, see **The Domain Design File**.

If you modify the Domain, you should also export the design file and update the security file with any IDs that have changed. Update the security file using the **Change** function on the Edit Domain page of the Domain Designer.

For the structure and syntax of the security file, see "Domain and Security Recommendations" in the *Server Security Guide*.

7.6 Locale Bundles

A locale bundle is a set of properties files used in localizing software. The files contain other-language versions of the labels and messages displayed to users. The locale bundles associated with a Domain provide localized text when creating and running reports based on Domains. When a user's operating environment specifies a locale, the server looks for the corresponding locale bundle and uses the labels and messages it contains. The localized text appears in the Ad Hoc designer and in the report viewer. The server interface is already internationalized, and Domains allow reports localized so users can create and view reports in their language.

A locale bundle for a Domain consists of a single file containing all the internationalization keys for the sets and items in the Domain. Each key is associated with the text for the label or description in the language of the target locale. The name of the file includes the target locale, so the server automatically associates it with the user's locale when needed.

Language	Locale	File Name	Sample Contents

English	default	ExampleDomain.properties	`ACCOUNTS.NAME.LABEL=Customer`
			`ACCOUNTS.NAME.DESCR=Name of Customer`

French	fr	ExampleDomain_fr.properties	`ACCOUNTS.NAME.LABEL=Client`
			`ACCOUNTS.NAME.DESCR=Nom du client`

When a Domain with locale bundles is used to create or run a report, the server resolves each label to display as follows:

- The server looks for a bundle corresponding to the current locale and for the key corresponding to the label. If both exist, the server displays the text associated with the key; if the bundle or key does not exist, then
- The server looks for the same key in a default bundle among the resources. If these exist, it displays the text associated with the key in the default bundle; if the default bundle or key does not exist, then
- It displays the text of the `label` property defined in the Domain design; if no label is defined, then
- It displays the value of the `id` property.

Descriptions are localized in the same manner, except they are left blank if there is no locale bundle, default bundle, or `description` property defined.

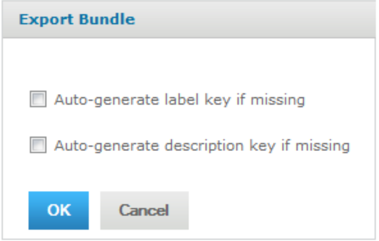
There are several strategies for managing the locale bundles. Based on the search algorithm above and your localization needs, you could have one of the following scenarios:

- You do not have any localization needs at the time you create the Domain. You give all labels and descriptions a clear and complete text within the design, and you leave the key names undefined. As your enterprise grows, users in another country need to create reports based on the Domain. To create localized text for those users, edit the Domain design to define the internationalization keys, then create the locale bundle. You can create locale bundles incrementally each time you have users who need a new language. Users in your home country who do not specify a locale still see the labels and description in the original Domain design.
- You want to create the Domain tailored to a multilingual environment where reports need to be localized for each user's specified locale. In this case, you do not specify any labels or descriptions in the Domain design, only the internationalization keys. Then you create a default locale bundle that gives the clear and complete text of every label and description in your preferred language. In this way, all user-visible text for the Domain is in the standard format of a locale bundle, and you can use specialized software or translation services to create all the locale bundles you need. In case a label or description is not translated, users see the text of the default locale bundle.

7.6.1 Defining the Internationalization Keys

In the Domain Designer, the name of the internationalization keys are defined by the Label Key and Description Key properties on each set and item on the **Display** tab. By default, these properties are blank. You may name the keys in any way you want, as long as each key is unique among all keys within the Domain. However, because keys are used in a Java properties file, they may only use characters from the ISO Latin-1 set, digits, and underscores (_).

To automate this process, use **Export Bundle** on the Display tab of the Domain Designer. The Confirm dialog box appears.



Export Bundle

□ Auto-generate label key if missing

□ Auto-generate description key if missing

Figure 7-11 Confirm Export Bundle Options Dialog Box

Exporting a bundle performs two tasks:

- Optionally generates all the key names
- Outputs a locale bundle stub

You have the option of generating only the label keys, only the description keys, both, or neither. When you click **OK**, the generated key names are added to all the blank `Label Key` and `Description Key` properties; any keys that already exist are not be modified. In order to guarantee uniqueness, the generated key names have the following format:

```
<setID>.LABEL
<setID>.DESCR
<setID>.<itemID>.LABEL
<setID>.<itemID>.DESCR
```

The locale bundle stub is a Java properties file in the proper format containing all the defined keys, ready for translation. The locale bundle stub is further explained in the next section.

In the design file, the name of the keys are given by the `labelId` and `descriptionId` attributes on each `itemGroup` and `item` element. If these attributes are missing or defined with an empty value (`""`), the keys are not defined. To define the keys, give these attributes a value that is unique among all the keys. Again, keys may only use characters from the ISO Latin-1 set, digits, and underscores (_).

If you have many sets and items, it may be easier to upload the design file in a Domain and open it in the Domain Designer. You can then generate the keys automatically, export the bundle as explained above, and export the XML design file that now contains the generated keys.

If the design file does not load properly in the Domain Designer, you are not be able to use the exported design file. In this case, do not overwrite the design file, but save the exported file to another name and attempt to merge in the generated keys.

7.6.2 Creating Locale Bundle Files

A locale bundle for a Domain is a Java properties file where each property name is a unique internationalization key from one of the labels or descriptions in the design. The value of each property is the text for the label or description in the target language. The name of the file identifies the locale in the form <any_name>_<locale>.properties, where <locale> is any Java-compliant locale identifier, for example `fr` or `fr_CA`. If the _<locale> part of the file name is omitted, the file designates the strings for the default locale.

Usually, the set of keys is identical in the properties file of each locale bundle, but this is not necessary. The text for a given key in a given locale is determined by the algorithm given in **“Locale Bundles” on page 277**.

Often, it is simplest to create the properties files from the stub exported by the Domain Designer. Whether you have created the design in the Domain Designer or in an external file uploaded to the Domain Designer, use **Export Bundle** in the menu bar to export a blank properties file. Optionally, you may generate any missing keys, as described in the previous section.

The following example shows part of the properties file output for the Domain created in **“Example of Creating a Domain” on page 204**. All of the internationalization keys were generated automatically as well. As the following example shows, sets and items in the bundle stub appear in the same order as on the Display tab of the Domain Designer.

```
ACCOUNTS.LABEL=
ACCOUNTS.DESCR=
ACCOUNTS.NAME.LABEL=
ACCOUNTS.NAME.DESCR=
ACCOUNTS.ACCOUNT_TYPE.LABEL=
ACCOUNTS.ACCOUNT_TYPE.DESCR=
ACCOUNTS.INDUSTRY.LABEL=
ACCOUNTS.INDUSTRY.DESCR=
...
ACCOUNTS.CITY_AND_STATE.LABEL=
ACCOUNTS.CITY_AND_STATE.DESCR=
USERS1.LABEL=
USERS1.DESCR=
USERS1.FIRST_NAME.LABEL=
USERS1.FIRST_NAME.DESCR=
USERS1.LAST_NAME.LABEL=
USERS1.LAST_NAME.DESCR=
...
```

The syntax for the properties file is the same as for Java properties files. Generally, everything after the = symbol is part of the translated text. Java properties files use the ISO-8859-1 (Latin-1) encoding that supports most, but not all, accented characters. For accented characters that are not in ISO-8859-1, or any other international characters, use Unicode escape sequences. For example, the œ character has the Unicode escape sequence `\u0153`.

The following example shows the French translation for the properties file above, including accented characters.

The single straight quote (') sometimes causes issues when processed in Domain properties files. To avoid this issue, use the right single quote (’) by its Unicode sequence \u2019, as shown in the following example.

```
ACCOUNTS.LABEL=Comptes
ACCOUNTS.DESCR=Détails des comptes clients.
ACCOUNTS.NAME.LABEL=Client
ACCOUNTS.NAME.DESCR=Nom du client
ACCOUNTS.ACCOUNT_TYPE.LABEL=Type
ACCOUNTS.ACCOUNT_TYPE.DESCR=Type du compte client.
ACCOUNTS.INDUSTRY.LABEL=Industrie
ACCOUNTS.INDUSTRY.DESCR=Industrie d\u2019activité primaire.
...
ACCOUNTS.CITY_AND_STATE.LABEL=Localité
ACCOUNTS.CITY_AND_STATE.DESCR=Ville et région ou état du client.
USERS1.LABEL=Responsable
USERS1.DESCR=Responsable du compte client.
USERS1.FIRST_NAME.LABEL=Prénom
USERS1.FIRST_NAME.DESCR=Prénom usuel.
USERS1.LAST_NAME.LABEL=Nom
USERS1.LAST_NAME.DESCR=Nom de famille.
...
```

After creating the properties file for the first locale, you can simply copy the file and change the `_<locale>` designator to begin translation of another locale bundle. A best practice is to save a blank properties file or the default locale bundle as a template for future translations. Also, if you change the Domain design, you can export the locale bundle again and compare it to the template to find all new keys.

You can create and edit the properties file in any text editor. Certain editors provide features for editing properties files. There are also translation products that manage all of the resource bundles together, allowing you to translate in parallel and find missing keys and missing translations.

Regardless of how you edit properties files, when all translations are finished, you need to upload them as locale bundles to the Domain. See the procedure in **"Security and Locale Information for a Domain" on page 274** for instructions about uploading locale bundles. Also, if you change a Domain design and need to modify the properties files, follow that same procedure to download, modify, and upload each locale bundle.

For additional information about locales and locale bundles in JasperReports Server, see the Localization chapter in the *JasperReports Server Administrator Guide*.

7.7 Internationalized Databases

In addition to locale bundles, JasperReports Server supports international characters in the database schema, for example in the table and column names. The server supports UTF-8 (8-bit Unicode Transformation Format) character encoding as described in the Localization chapter in the *JasperReports Server Administrator Guide*.

If your database is properly configured and the appropriate fonts are available to your browser, the Domain Designer displays the international characters on the various tabs where table and column names appear. This means that your data architects can work in their native language within the Domain Designer.

As with all values from your reporting database, table and column names are not localizable. They can only be displayed in the Domain Designer as they appear in the database. You can, however, have localized labels for these table and column names, so that they appear in the Ad Hoc designer according to the user's locale.

All UTF-8 characters are allowed in IDs except the following symbols that will cause errors: `().,"=!+-></:[]*|?$`. This limitation applies to table IDs in the database as well as user created IDs in design files and in the Domain Designer. In addition, the straight single quote (`'`, Unicode 0027) can appear in database table IDs but not in user-created IDs. In general, the best practice is to use alphabet characters, not punctuation or arithmetic symbols when creating Domains.

7.8 Switching the Data Source of a Domain

You can select a different data source for a Domain on the Edit Domain page or in the Manage Data Source dialog box. However, because the Domain design relies extensively on the data source, it is likely that some tables in your Domain won't be valid and information such as derived tables and calculated fields will be lost. Always back up a Domain before switching the data source, and make sure that the change makes sense for this Domain and data source. Examples where it may make sense to switch the data source include:

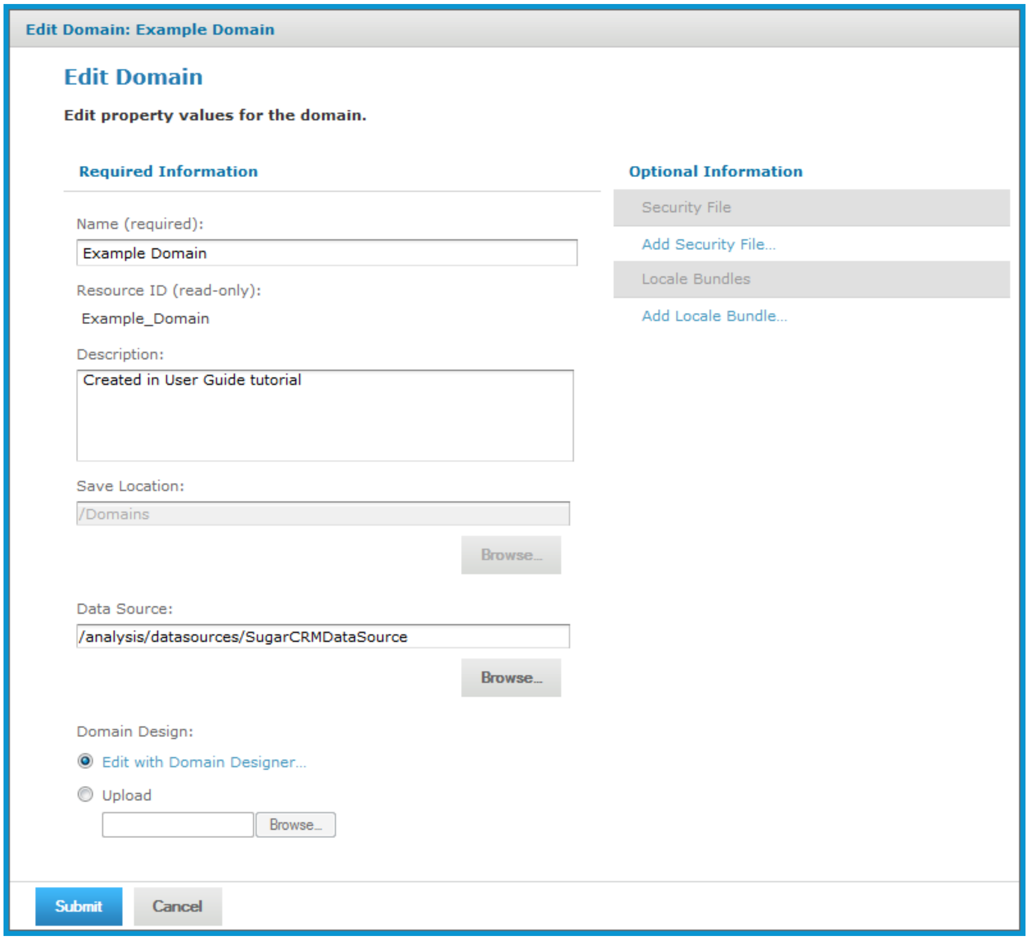
- Switching to a data source with a different database schema but with the same tables; for example, when moving from staging to production.
- Switching from a regular data source to a virtual data source that contains the original data source. In this case, JasperReports Server attempts to locate the original data source inside the virtual data source and create the correct prefix. Derived tables and calculated fields are not preserved.
- Switching from a virtual data source to one of the data sources it includes. In this case, any resources that are not in the selected data source are deleted along with any dependent information, such as joins or labels that depend on the deleted resources. Derived tables and calculated fields are not preserved.

If you change to a data source with a different schema structure, the definitions in the Domain design are no longer valid and you cannot save the Domain.

To change the data source of a Domain:

1. Log in to the server as an administrator and select **View > Search Results**.
2. Locate the Domain.
   a. Choose **View > Search Results**.
   b. In the Filters panel, under Types, click **More choices**.
   c. Click Domains.
3. Create a copy of the Domain as described in **"Copying a Domain" on page 242**.
4. Locate the Domain again, right-click it and select **Edit** from the context menu.

   The Edit Domain page appears.



Edit Domain: Example Domain

Edit Domain

Edit property values for the domain.

Required Information

Name (required):
Example Domain

Resource ID (read-only):
Example_Domain

Description:
Created in User Guide tutorial

Save Location:
/Domains

Data Source:
/analysis/datasources/SugarCRMDataSource

Domain Design:
● Edit with Domain Designer...
○ Upload

Optional Information

Figure 7-12 Edit Domain Page

5. Export the Domain design file as described in **"Exporting the Design File from a Domain" on page 246**. You can use this file to help reconstruct any derived tables or calculated fields.
6. In the **Name** field, change the display name of the Domain.

7. In the **Description** field, change the description of the Domain.
8. In the **Data Source** field, browse to locate a new data source for the Domain.
9. Click **Edit with Domain Designer**. If you are working with a virtual data source and the database schema used by the original Domain is not available in the data source, you are asked if you want to select a schema. Select **Yes** to select one or more schemas and click **OK**; the first schema you select is added to the name of the tables in the Domain. Select **No** to remove all schema prefixes from the tables in the Domain.
10. Click **Submit** to update the Domain. To close the Domain Designer without modifying the Domain stored in the repository, click **Cancel**.

As mentioned above, in some cases, derived tables and calculated fields will be missing. You may also see invalid tables. You can work with the copy of your original Domain and its exported design file to determine the modifications you need to make to fully restore your domain. For more information, you can export the design file of the modified Domain, as described in **"Exporting the Design File from a Domain" on page 246**, and compare it to the design file of the original Domain.

GLOSSARY

Ad Hoc Editor

The interactive data explorer in JasperReports Server Professional and Enterprise editions. Starting from a predefined collection of fields, the Ad Hoc Editor lets you drag and drop fields, dimensions, and measures to explore data and create tables, charts, and crosstabs. These Ad Hoc views can be saved as reports.

Ad Hoc Report

In previous versions of JasperReports Server, a report created through the Ad Hoc Editor. Such reports could be added to dashboards and be scheduled, but when edited in Jaspersoft Studio, lost their grouping and sorting. In the current version, the Ad Hoc Editor is used to explore views which in turn can be saved as reports. Such reports can be edited in Jaspersoft Studio without loss, and can be scheduled and added to dashboards.

Ad Hoc View

A view of data that is based on a Domain, Topic, or OLAP client connection. An Ad Hoc view can be a table, chart, or crosstab and is the entry point to analysis operations such as slice and dice, drill down, and drill through. **Compare OLAP View.** You can save an Ad Hoc view as a report in order to edit it in the interactive viewer, schedule it, or add it to a dashboard.

Aggregate Function

An aggregate function is one that is computed using a group of values; for example, Sum or Average. Aggregate functions can be used to create calculated fields in Ad Hoc views. Calculated fields containing aggregate functions cannot be used as fields or added to groups in an Ad Hoc view and should not be used as filters. Aggregate functions allow you to set a level, which specifies the scope of the calculation; level values include Current (not available for PercentOf), ColumnGroup, ColumnTotal, RowGroup, RowTotal, Total

Analysis View

See OLAP View.

Audit Archiving

To prevent audit logs from growing too large to be easily accessed, the installer configures JasperReports Server to move current audit logs to an archive after a certain number of days, and to delete logs in the archive after a certain age. The archive is another table in the JasperReports Server's repository database.

Audit Domains

A Domain that accesses audit data in the repository and lets administrators create Ad Hoc reports of server activity. There is one Domain for current audit logs and one for archived logs.

Audit Logging

When auditing is enabled, audit logging is the active recording of who used JasperReports Server to do what when. The system installer can configure what activities to log, the amount of detail gathered, and when to archive the data. Audit logs are stored in the same private database that JasperReports Server uses to store the repository, but the data is only accessible through the audit Domains.

Auditing

A feature of JasperReports Server Enterprise edition that records all server activity and allows administrators to view the data.

Calculated Field

In an Ad Hoc view or a Domain, a field whose value is calculated from a user-defined formula that may include any number of fields, operators, and constants. For Domains, a calculated field becomes one of the items to which the Domain's security file and locale bundles can apply. There are more functions available for Ad Hoc view calculations than for Domains.

CRM

Customer Relationship Management. The practice of managing every facet of a company's interactions with its clientele. CRM applications help businesses track and support their customers.

CrossJoin

An MDX function that combines two or more dimensions into a single axis (column or row).

Cube

The basis of most OLAP applications, a cube is a data structure that contains three or more dimensions that categorize the cube's quantitative data. When you navigate the data displayed in an OLAP view, you are exploring a cube.

Custom Field

In the Ad Hoc Editor, a field that is created through menu items as a simple function of one or two available fields, including other custom fields. When a custom field becomes too complex or needs to be used in many reports, it is best to define it as a calculated field in a Domain.

Dashboard

A collection of reports, input controls, graphics, labels, and web content displayed in a single, integrated view. Dashboards often present a high level view of your data, but input controls can parametrize the data to display. For example, you can narrow down the data to a specific date range. Embedded web content, such as other web-based applications or maps, make dashboards more interactive and functional.

Dashlet

An element in a dashboard. Dashlets are defined by editable properties that vary depending on the dashlet type. Types of dashlet include reports, text elements, filters, and external web content.

Data Island

A single join tree or a table without joins in a Domain. A Domain may contain several data islands, but when creating an Ad Hoc view from a Domain, you can only select one of them to be available in the view.

Data Policy

In JasperReports Server, a setting that determines how the server processes and caches data used by Ad Hoc reports. Select your data policies by clicking **Manage > Server > Settings Ad Hoc Settings**. By default, this setting is only available to the superuser account.

Data Source

Defines the connection properties that JasperReports Server needs to access data. The server transmits queries to data sources and obtains datasets in return for use in filling reports and previewing Ad Hoc reports. JasperReports Server supports JDBC, JNDI, and Bean data sources; custom data sources can be defined as well.

Dataset

A collection of data arranged in columns and rows. Datasets are equivalent to relational results sets and the `JRDataSource` type in the JasperReports Library.

Datatype

In JasperReports Server, a datatype is used to characterize a value entered through an input control. A datatype must be of type text, number, date, or date-time. It can include constraints on the value of the input, for example maximum and minimum values. As such, a datatype in JasperReports Server is more structured than a datatype in most programming languages.

Denormalize

A process for creating table joins that speeds up data retrieval at the cost of having duplicate row values between some columns.

Derived Table

In a Domain, a derived table is defined by an additional query whose result becomes another set of items available in the Domain. For example, with a JDBC data source, you can write an SQL query that includes complex functions for selecting data. You can use the items in a derived table for other operations on the Domain, such as joining tables, defining a calculated field, or filtering. The items in a derived table can also be referenced in the Domain's security file and locale bundles.

Dice

An OLAP operation to select columns.

Dimension

A categorization of the data in a cube. For example, a cube that stores data about sales figures might include dimensions such as time, product, region, and customer's industry.

Domain

A virtual view of a data source that presents the data in business terms, allows for localization, and provides data-level security. A Domain is not a view of the database in relational terms, but it implements the same functionality within JasperReports Server. The design of a Domain specifies tables in the database, join clauses, calculated fields, display names, and default properties, all of which define items and sets of items for creating Ad Hoc reports.

Domain Topic

A Topic that is created from a Domain by the Data Chooser. A Domain Topic is based on the data source and items in a Domain, but it allows further filtering, user input, and selection of items. Unlike a JRXML-based Topic, a Domain Topic can be edited in JasperReports Server by users with the appropriate permissions.

Drill

To click on an element of an OLAP view to change the data that is displayed:

- Drill down. An OLAP operation that exposes more detailed information down the hierarchy levels by delving deeper into the hierarchy and updating the contents of the navigation table.

- Drill through. An OLAP operation that displays detailed transactional data for a given aggregate measure. Click a fact to open a new table beneath the main navigation table; the new table displays the low-level data that constitutes the data that was clicked.
- Drill up. An OLAP operation for returning the parent hierarchy level to view to summary information.

Eclipse

An open source Integrated Development Environment (IDE) for Java and other programming languages, such as C/C++.

ETL

Extract, Transform, Load. A process that retrieves data from transactional systems, and filters and aggregates the data to create a multidimensional database. Generally, ETL prepares the database that your reports will access. The Jaspersoft ETL product lets you define and schedule ETL processes.

Fact

The specific value or aggregate value of a measure for a particular member of a dimension. Facts are typically numeric.

Field

A field is equivalent to a column in the relational database model. Fields originate in the structure of the data source, but you may define calculated fields in a Domain or custom fields in the Ad Hoc Editor. Any type of field, along with its display name and default formatting properties, is called an item and may be used in the Ad Hoc Editor.

Frame

In Jaspersoft Studio, a frame is a rectangular element that can contain other elements and optionally draw a border around them. Elements inside a frame are positioned relative to the frame, not to the band, and when you move a frame, all the elements contained in the frame move together. A frame automatically stretches to fit its contents.

Frame can also refer to an element in a legacy dashboard; it's the equivalent of a dashlet.

Group

In a report, a group is a set of data rows that have an identical value in a designated field.

- In a table, the value appears in a header and footer around the rows of the group, while the other fields appear as columns.
- In a chart, the field chosen to define the group becomes the independent variable on the X axis, while the other fields of each group are used to compute the dependent value on the Y axis.

Hierarchy Level

In an OLAP cube, a member of a dimension containing a group of members.

Input Control

A button, check box, drop-down list, text field, or calendar icon that allows users to enter a value when running a report or viewing a dashboard that accepts input parameters. For JRXML reports, input controls and their associated datatypes must be defined as repository objects and explicitly associated with the report. For Domain-based reports that prompt for filter values, the input controls are defined internally. When either type of report is used in a dashboard, its input controls are available to be added as special content.

Item

When designing a Domain or creating a Topic based on a Domain, an item is the representation of a database field or a calculated field along with its display name and formatting properties defined in the Domain. Items can be grouped in sets and are available for use in the creation of Ad Hoc reports.

JasperReport

A combination of a report template and data that produces a complex document for viewing, printing, or archiving information. In the server, a JasperReport references other resources in the repository:

- The report template (in the form of a JRXML file)
- Information about the data source that supplies data for the report
- Any additional resources, such as images, fonts, and resource bundles referenced by the report template.

The collection of all the resources that are referenced in a JasperReport is sometimes called a report unit. End users usually see and interact with a JasperReport as a single resource in the repository, but report creators must define all of the components in the report unit.

Jaspersoft Studio

A commercial open source tool for graphically designing reports that leverage all features of the JasperReports Library. Jaspersoft Studio lets you drag and drop fields, charts, and sub-reports onto a canvas, and also define parameters or expressions for each object to create pixel-perfect reports. You can generate the JRXML of the report directly in Jaspersoft Studio, or upload it to JasperReports Server. Jaspersoft Studio is implemented in Eclipse.

JasperReports Library

An embeddable, open source, Java API for generating a report, filling it with current data, drawing charts and tables, and exporting to any standard format (HTML, PDF, Excel, CSV, and others). JasperReports processes reports defined in JRXML, an open XML format that allows the report to contain expressions and logic to control report output based on run-time data.

JasperReports Server

A commercial open source, server-based application that calls the JasperReports Library to generate and share reports securely. JasperReports Server authenticates users and lets them upload, run, view, schedule, and send reports from a web browser. Commercial versions provide metadata layers, interactive report and dashboard creation, and enterprise features such as organizations and auditing.

Jaspersoft ETL

A graphical tool for designing and implementing your data extraction, transforming, and loading (ETL) tasks. It provides hundreds of data source connectors to extract data from many relational and non-relational systems. Then, it schedules and performs data aggregation and integration into data marts or data warehouses that you use for reporting.

Jaspersoft OLAP

A relational OLAP server integrated into JasperReports Server that performs data analysis with MDX queries. The product includes query builders and visualization clients that help users explore and make sense of multidimensional data. Jaspersoft OLAP also supports XML/A connections to remote servers.

Jaspersoft Studio

An open source tool for graphically designing reports that leverage all features of the JasperReports Library. Jaspersoft Studio lets you drag and drop fields, charts, and sub-reports onto a canvas, and also define parameters or expressions for each object to create pixel-perfect reports. You can generate the JRXML of the report directly in Jaspersoft Studio, or upload it to JasperReports Server. Jaspersoft Studio is implemented in Eclipse.

JavaBean

A reusable Java component that can be dropped into an application container to provide standard functionality.

JDBC

Java Database Connectivity. A standard interface that Java applications use to access databases.

JNDI

Java Naming and Directory Interface. A standard interface that Java applications use to access naming and directory services.

Join Tree

In Domains, a collection of joined tables from the actual data source. A join is the relational operation that associates the rows of one table with the rows of another table based on a common value in given field of each table. Only the fields in a same join tree or calculated from the fields in a same join tree may appear together in a report.

JPivot

An open source graphical user interface for OLAP operations. For more information, visit http://jpivot.sourceforge.net/.

JRXML

An XML file format for saving and sharing reports created for the JasperReports Library and the applications that use it, such as Jaspersoft Studio and JasperReports Server. JRXML is an open format that uses the XML standard to define precisely all the structure and configuration of a report.

Level

Specifies the scope of an aggregate function in an Ad Hoc view. Level values include Current (not available for PercentOf), ColumnGroup, ColumnTotal, RowGroup, RowTotal, Total.

MDX

Multidimensional Expression Language. A language for querying multidimensional objects, such as OLAP (On Line Analytical Processing) cubes, and returning cube data for analytical processing. An MDX query is the query that determines the data displayed in an OLAP view.

Measure

Depending on the context:

- In a report, a formula that calculates the values displayed in a table's columns, a crosstab's data values, or a chart's dependent variable (such as the slices in a pie).
- In an OLAP view, a formula that calculates the facts that constitute the quantitative data in a cube.

Mondrian

A Java-based, open source multidimensional database application.

Mondrian Connection

An OLAP client connection that consists of an OLAP schema and a data source. OLAP client connections populate OLAP views.

Mondrian Schema Editor

An open source Eclipse plug-in for creating Mondrian OLAP schemas.

Mondrian XML/A Source

A server-side XML/A source definition of a remote client-side XML/A connection used to populate an OLAP view using the XML/A standard.

MySQL

An open source relational database management system. For information, visit http://www.mysql.com/.

Navigation Table

The main table in an OLAP view that displays measures and dimensions as columns and rows.

ODBO Connect

Jaspersoft ODBO Connect enables Microsoft Excel 2003 and 2007 Pivot Tables to work with Jaspersoft OLAP and other OLAP servers that support the XML/A protocol. After setting up the Jaspersoft ODBO data source, business analysts can use Excel Pivot Tables as a front-end for OLAP analysis.

OLAP

On Line Analytical Processing. Provides multidimensional views of data that help users analyze current and past performance and model future scenarios.

OLAP Client Connection

A definition for retrieving data to populate an OLAP view. An OLAP client connection is either a direct Java connection (Mondrian connection) or an XML-based API connection (XML/A connection).

OLAP Schema

A metadata definition of a multidimensional database. In Jaspersoft OLAP, schemas are stored in the repository as XML file resources.

OLAP View

Also called an analysis view. A view of multidimensional data that is based on an OLAP client connection and an MDX query. Unlike Ad Hoc views, you can directly edit an OLAP view's MDX query to change the data and the way they are displayed. An OLAP view is the entry point for advanced analysis users who want to write their own queries. **Compare Ad Hoc View.**

Organization

A set of users that share folders and resources in the repository. An organization has its own user accounts, roles, and root folder in the repository to securely isolate it from other organizations that may be hosted on the same instance of JasperReports Server.

Organization Admin

Also called the organization administrator. A user in an organization with the privileges to manage the organization's user accounts and roles, repository permissions, and repository content. An organization admin can also create suborganizations and mange all of their accounts, roles, and repository objects. The default organization admin in each organization is the `jasperadmin` account.

Outlier

A fact that seems incongruous when compared to other member's facts. For example, a very low sales figure or a very high number of help desk tickets. Such outliers may indicate a problem (or an important achievement) in your business. The analysis features of Jaspersoft OLAP excel at revealing outliers.

Parameter

Named values that are passed to the engine at report-filling time to control the data returned or the appearance and formatting of the report. A report parameter is defined by its name and type. In JasperReports Server, parameters can be mapped to input controls that users can interact with.

Pivot

To rotate a crosstab such that its row groups become column groups and its column groups become rows. In the Ad Hoc Editor, pivot a crosstab by clicking ↻.

Pivot Table

A table with two physical dimensions (for example, X and Y axis) for organizing information containing more than two logical dimensions (for example, PRODUCT, CUSTOMER, TIME, and LOCATION), such that each physical dimension is capable of representing one or more logical dimensions, where the values described by the dimensions are aggregated using a function such as SUM. Pivot tables are used in Jaspersoft OLAP.

Properties

Settings associated with an object. The settings determine certain features of the object, such as its color and label. Properties are normally editable. In Java, properties can be set in files listing objects and their settings.

Report

In casual usage, *report* may refer to:

- A JasperReport. **See JasperReport.**
- The main JRXML in a JasperReport.
- The file generated when a JasperReport is scheduled. Such files are also called content resources or output files.
- The file generated when a JasperReport is run and then exported.
- In previous JasperReports Server versions, a report created in the Ad Hoc Editor. **See Ad Hoc Report.**

Report Run

An execution of a report, Ad Hoc view, or dashboard, or a view or dashboard designer session, it measures and limits usage of Freemium instances of JasperReports Server. The executions apply to resources no matter how they are run (either in the web interface or through the various APIs, such as REST web services). Users of our Community Project and our full-use commercial licenses are not affected by the limit. For more information, please contact sales@jaspersoft.com.

Repository

The tree structure of folders that contain all saved reports, dashboards, OLAP views, and resources. Users access the repository through the JasperReports Server web interface or through Jaspersoft Studio. Applications can access the repository through the web service API. Administrators use the import and export utilities to back up the repository contents.

Resource

In JasperReports Server, anything residing in the repository, such as an image, file, font, data source, Topic, Domain, report element, saved report, report output, dashboard, or OLAP view. Resources also include the folders in the repository. Administrators set user and role-based access permissions on repository resources to establish a security policy.

Role

A security feature of JasperReports Server. Administrators create named roles, assign them to user accounts, and then set access permissions to repository objects based on those roles. Certain roles also determine what functionality and menu options are displayed to users in the JasperReports Server interface.

Schema

A logical model that determines how data is stored. For example, the schema in a relational database is a description of the relationships between tables, views, and indexes. In Jaspersoft OLAP, an OLAP schema is the logical model of the data that appears in an OLAP view; they are uploaded to the repository as resources. For Domains, schemas are represented in XML design files.

Schema Workbench

A graphical tool for easily designing OLAP schemas, data security schemas, and MDX queries. The resulting cube and query definitions can then be used in Jaspersoft OLAP to perform simple but powerful analysis of large quantities of multi-dimensional data stored in standard RDBMS systems.

Set

In Domains and Domain Topics, a named collection of items grouped together for ease of use in the Ad Hoc Editor. A set can be based on the fields in a table or entirely defined by the Domain creator, but all items in a set must originate in the same join tree. The order of items in a set is preserved.

Slice

An OLAP operation for filtering data rows.

SQL

Structured Query Language. A standard language used to access and manipulate data and schemas in a relational database.

System Admin

Also called the system administrator. A user who has unlimited access to manage all organizations, users, roles, repository permissions, and repository objects across the entire JasperReports Server instance. The system admin can create root-level organizations and manage all server settings. The default system admin is the `superuser` account.

Topic

A JRXML file created externally and uploaded to JasperReports Server as a basis for Ad Hoc reports. Topics are created by business analysts to specify a data source and a list of fields with which business users can create reports in the Ad Hoc Editor. Topics are stored in the Ad Hoc Components folder of the repository and displayed when a user launches the Ad Hoc Editor.

Transactional Data

Data that describe measurable aspects of an event, such as a retail transaction, relevant to your business. Transactional data are often stored in relational databases, with one row for each event and a table column or field for each measure.

User

Depending on the context:

- A person who interacts with JasperReports Server through the web interface. There are generally three categories of users: administrators who install and configure JasperReports Server, database experts or business analysts who create data sources and Domains, and business users who create and view reports and dashboards.

- A user account that has an ID and password to enforce authentication. Both people and API calls accessing the server must provide the ID and password of a valid user account. Roles are assigned to user accounts to determine access to objects in the repository.

View

Several meanings pertain to JasperReports Server:

- An Ad Hoc view. **See Ad Hoc View.**
- An OLAP view. **See OLAP View.**
- A database view. See http://en.wikipedia.org/wiki/View_%28database%29.

Virtual Data Source

A virtual data source allows you to combine data residing in multiple JDBC and/or JNDI data sources into a single data source that can query the combined data. Once you have created a virtual data source, you create Domains that join tables across the data sources to define the relationships between the data sources.

WCF

Web Component Framework. A low-level GUI component of JPivot. For more information, see http://jpivot.sourceforge.net/wcf/index.html.

Web Services

A SOAP (Simple Object Access Protocol) API that enables applications to access certain features of JasperReports Server. The features include repository, scheduling and user administration tasks.

XML

eXtensible Markup language. A standard for defining, transferring, and interpreting data for use across any number of XML-enabled applications.

XML/A

XML for Analysis. An XML standard that uses Simple Object Access protocol (SOAP) to access remote data sources. For more information, see http://www.xmla.org/.

XML/A Connection

A type of OLAP client connection that consists of Simple Object Access Protocol (SOAP) definitions used to access data on a remote server. OLAP client connections populate OLAP views.

Index