TIBCO JasperReports® Server User Guide

Software Release 7.2

Important Information

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 171** 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.

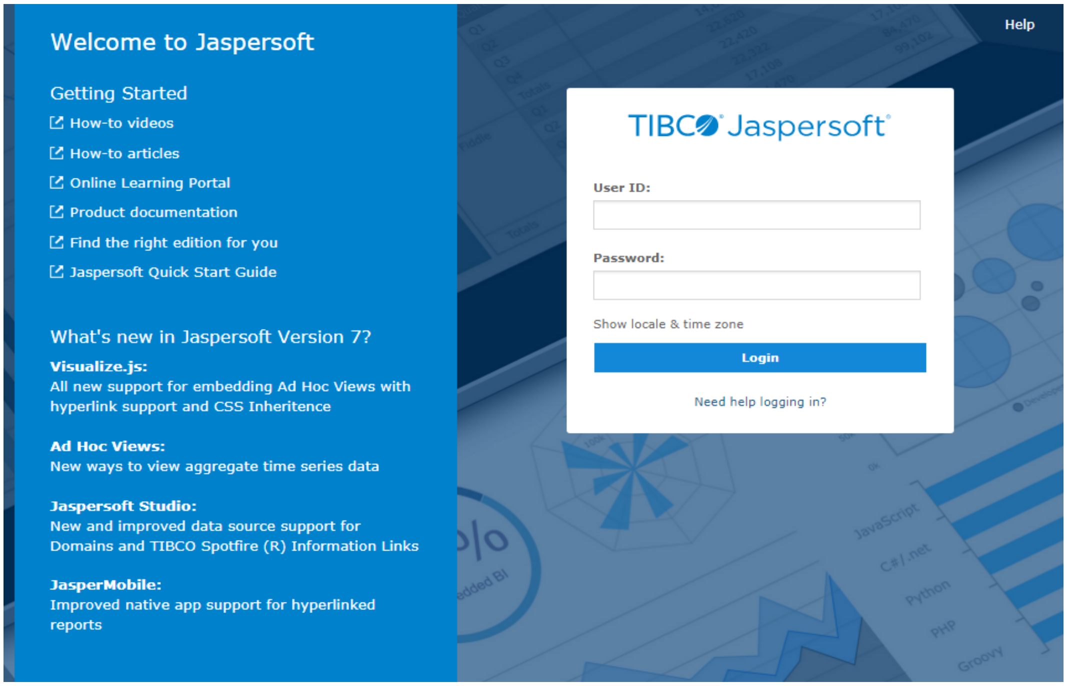
- Our Online Learning Portal lets you learn at your own pace, and covers topics for developers, system administrators, business users, and data integration users. The Portal is available online from the Professional Services section of our website.
- Our free samples, which are installed with JasperReports Library, Jaspersoft Studio, and JasperReports Server, are available and documented online. Please visit our GitHub repository. See **"Adding Reports Directly to the Repository" on page 171** and *Jaspersoft Studio User Guide* for more information.
- If you have a subscription to our professional support offerings, please contact our Technical Support team when you have questions or run into difficulties. They're available on the web at and through email at http://support.tibco.com and js-support@tibco.com.

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 views and reports, advanced charts, dashboards, Domains, auditing, and a multi-organization architecture for hosting large BI deployments.

This chapter contains the following sections:

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.



Welcome to Jaspersoft

Getting Started

- How-to videos
- How-to articles
- Online Learning Portal
- Product documentation
- Find the right edition for you
- Jaspersoft Quick Start Guide

What's new in Jaspersoft Version 7?

Visualize.js:
All new support for embedding Ad Hoc Views with hyperlink support and CSS Inheritence

Ad Hoc Views:
New ways to view aggregate time series data

Jaspersoft Studio:
New and improved data source support for Domains and TIBCO Spotfire (R) Information Links

JasperMobile:
Improved native app support for hyperlinked reports

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.

You can log in as the following users:

- **superuser/superuser** to manage configuration and organizations
- **jasperadmin/jasperadmin** to manage a single organization
- **joeuser/joeuser** to see an end user's view
- **demo/demo** to view the demo dashboard, if samples were installed

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

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 12**.

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

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.

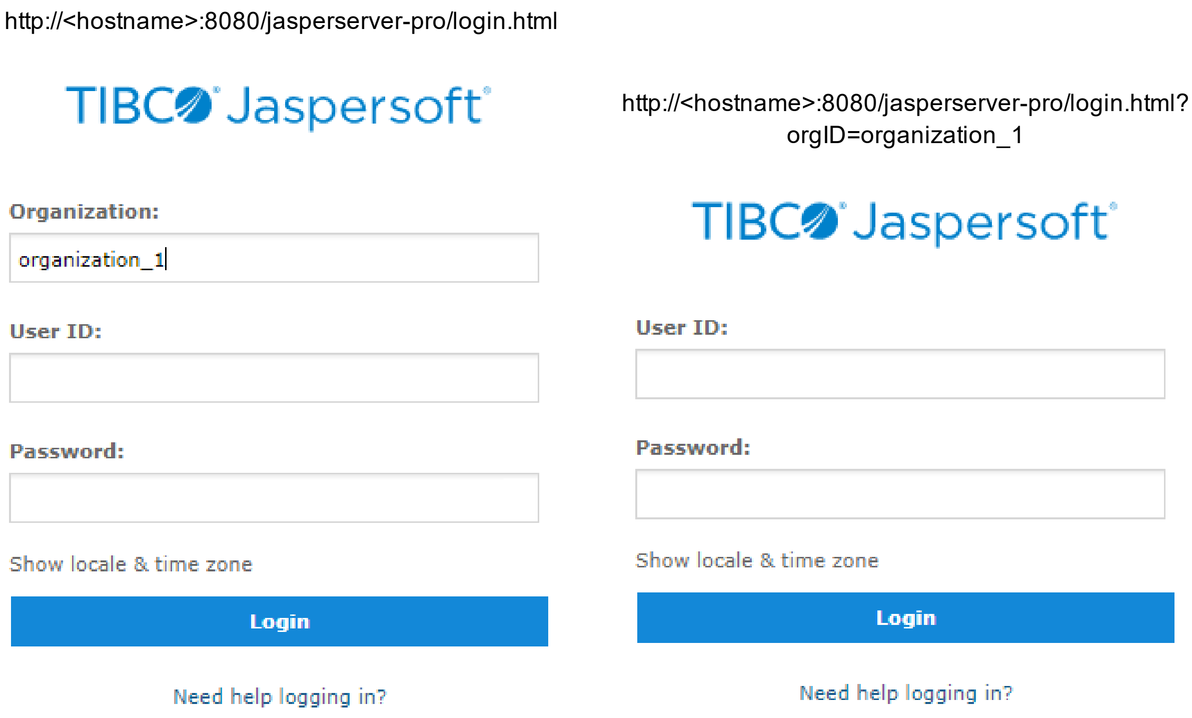
3. Click **Login**.

   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.



http://<hostname>:8080/jasperserver-pro/login.html

Organization: organization_1

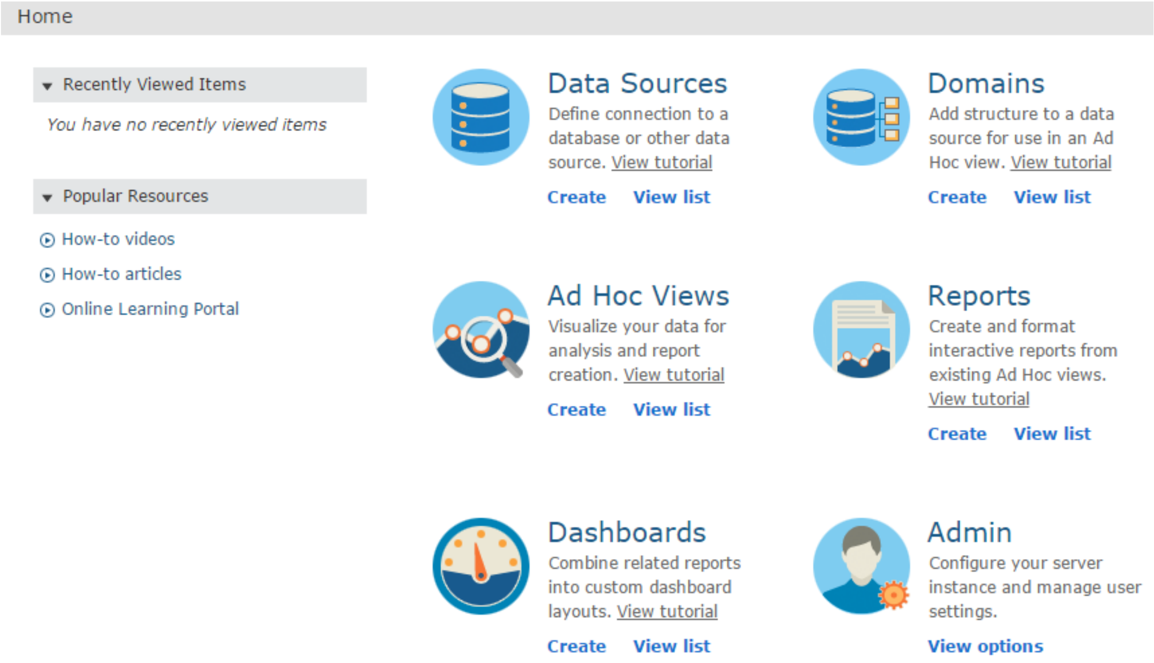
http://<hostname>:8080/jasperserver-pro/login.html?orgID=organization_1

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.
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:

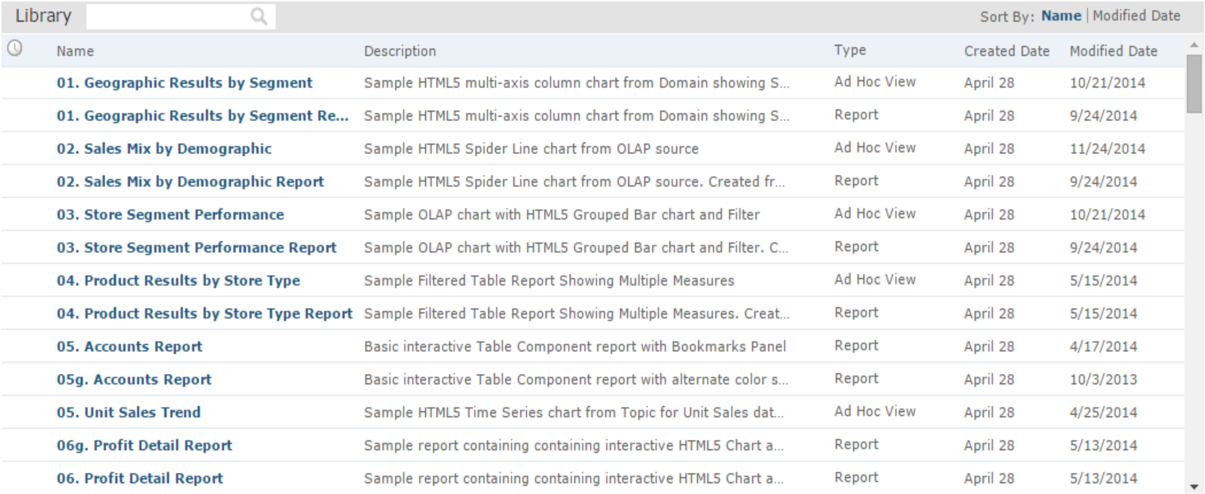
- Internet Explorer 8 with JAWS 14
- Internet Explorer 11 with JAWS 16

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.

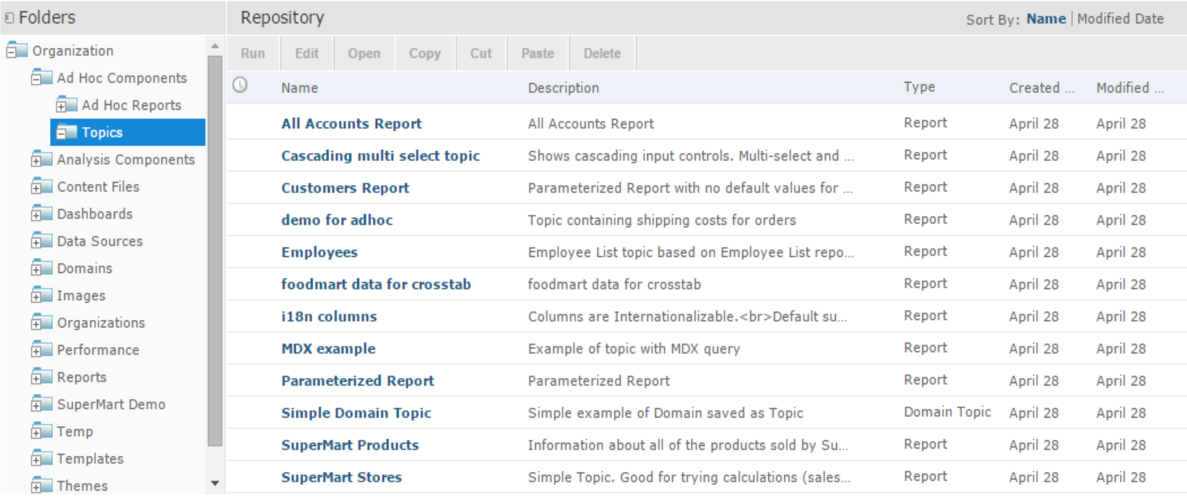
Generally, the created date will be earlier than the modified date. In some situations, however, the created date may be after the modified date. This can happen for one of two reasons:

- 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

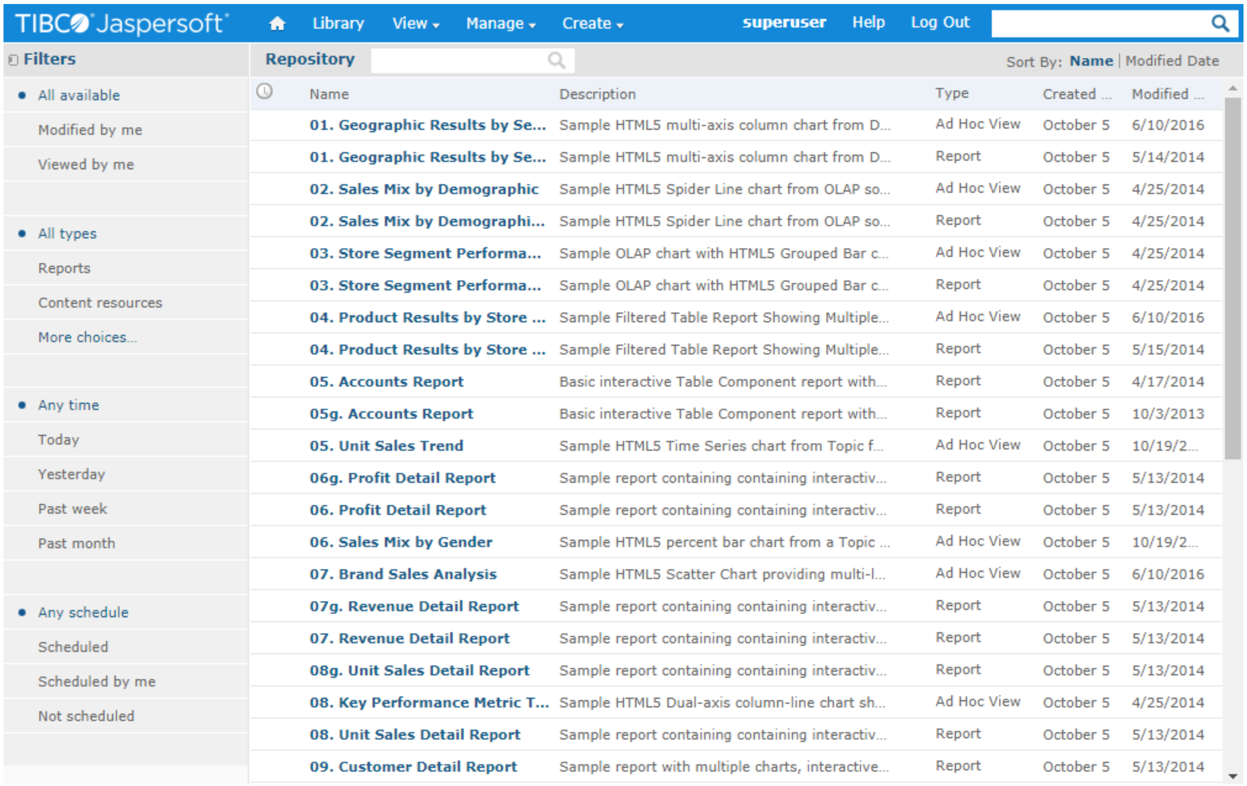
1.5 Searching the Repository

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 ...
01. Geographic Results by Se...	Sample HTML5 multi-axis column chart from D...	Ad Hoc View	October 5	6/10/2016
01. Geographic Results by Se...	Sample HTML5 multi-axis column chart from D...	Report	October 5	5/14/2014
02. Sales Mix by Demographic	Sample HTML5 Spider Line chart from OLAP so...	Ad Hoc View	October 5	4/25/2014
02. Sales Mix by Demographi...	Sample HTML5 Spider Line chart from OLAP so...	Report	October 5	4/25/2014
03. Store Segment Performa...	Sample OLAP chart with HTML5 Grouped Bar c...	Ad Hoc View	October 5	4/25/2014
03. Store Segment Performa...	Sample OLAP chart with HTML5 Grouped Bar c...	Report	October 5	4/25/2014
04. Product Results by Store ...	Sample Filtered Table Report Showing Multiple...	Ad Hoc View	October 5	6/10/2016
04. Product Results by Store ...	Sample Filtered Table Report Showing Multiple...	Report	October 5	5/15/2014
05. Accounts Report	Basic interactive Table Component report with...	Report	October 5	4/17/2014
05g. Accounts Report	Basic interactive Table Component report with...	Report	October 5	10/3/2013
05. Unit Sales Trend	Sample HTML5 Time Series chart from Topic f...	Ad Hoc View	October 5	10/19/2...
06g. Profit Detail Report	Sample report containing containing interactiv...	Report	October 5	5/13/2014
06. Profit Detail Report	Sample report containing containing interactiv...	Report	October 5	5/13/2014
06. Sales Mix by Gender	Sample HTML5 percent bar chart from a Topic ...	Ad Hoc View	October 5	10/19/2...
07. Brand Sales Analysis	Sample HTML5 Scatter Chart providing multi-l...	Ad Hoc View	October 5	6/10/2016
07g. Revenue Detail Report	Sample report containing containing interactiv...	Report	October 5	5/13/2014
07. Revenue Detail Report	Sample report containing containing interactiv...	Report	October 5	5/13/2014
08g. Unit Sales Detail Report	Sample report containing containing interactiv...	Report	October 5	5/13/2014
08. Key Performance Metric T...	Sample HTML5 Dual-axis column-line chart sh...	Ad Hoc View	October 5	4/25/2014
08. Unit Sales Detail Report	Sample report containing containing interactiv...	Report	October 5	5/13/2014
09. Customer Detail Report	Sample report with multiple charts, interactive...	Report	October 5	5/13/2014

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.

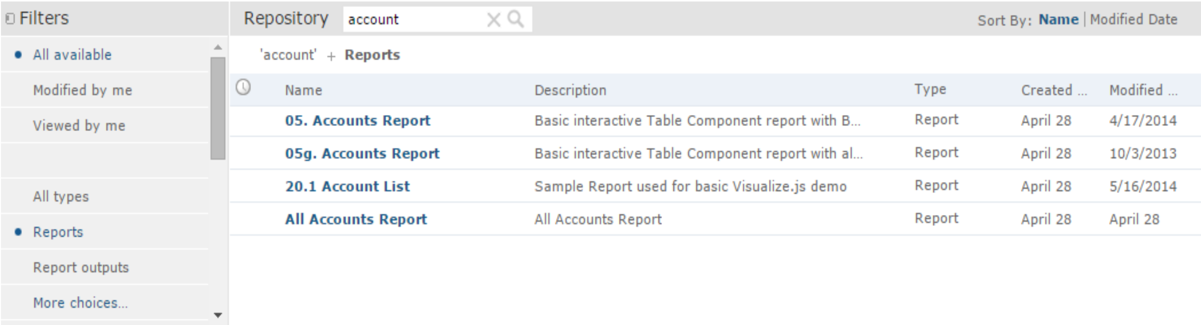
1.5.2 Filtering Search Results

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.

Filter Setting	Description
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.
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 71**.
- 🕓 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 77**.

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.

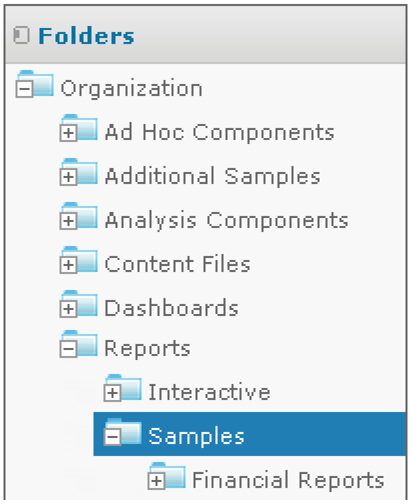


Figure 1-8 New Financial Reports Folder

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.

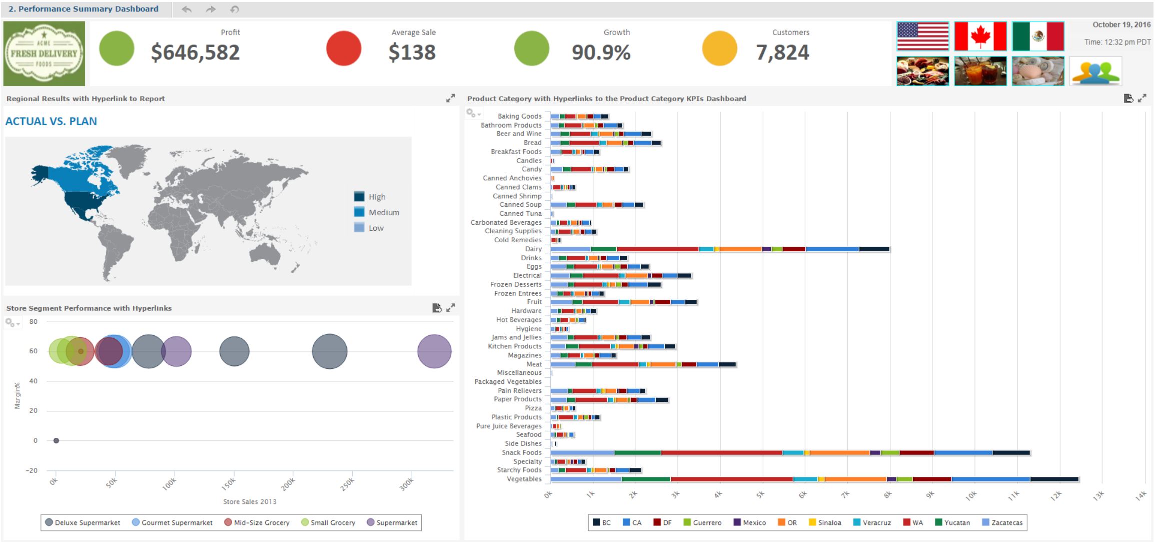


Figure 2-1 Sample dashboard

This chapter contains the following sections:

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

2.1 Viewing a Dashboard

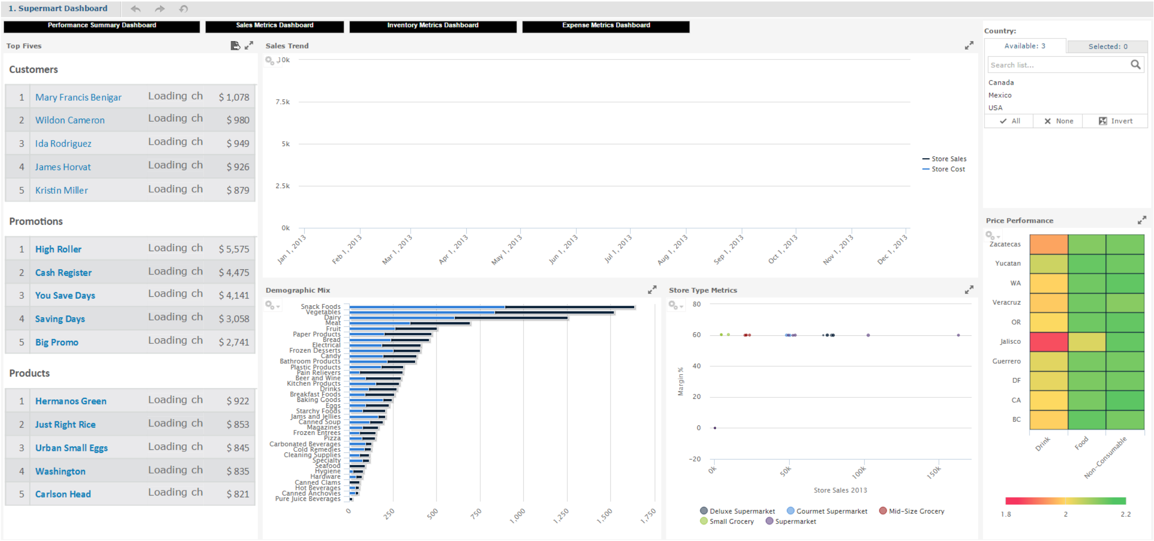
You can view a dashboard if you have the proper permissions. The following procedure walks you through opening one of JasperReports Server's samples, the SuperMart dashboard.

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.



Customers

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

Promotions

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

Products

1	Hermanos Green	Loading ch	$ 922
2	Just Right Rice	Loading ch	$ 853
3	Urban Small Eggs	Loading ch	$ 845
4	Washington	Loading ch	$ 835
5	Carlson Head	Loading ch	$ 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 dashlets that use the input control reflect the change.
- The ↶ ↷ ↺ buttons on the toolbar allow you to undo and redo recent changes made to the dashboard, including changes using an input control.
- 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.

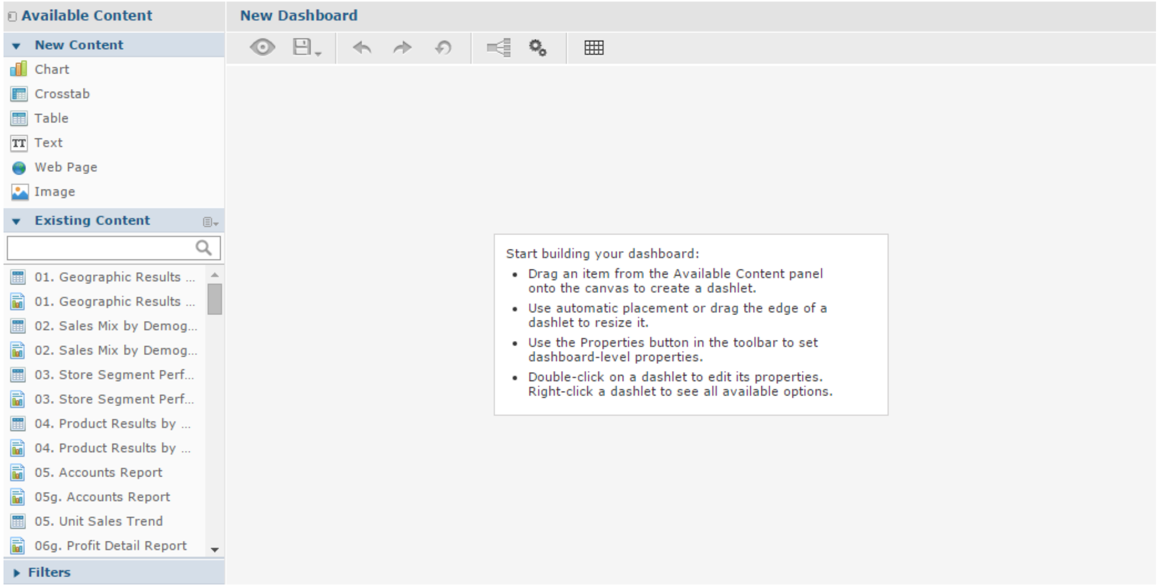
Your permissions to access the repository may limit the content you can add and the location where you can save the dashboard.

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 Dashboard
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

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.
⚙	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.
⟳	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 52** 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:

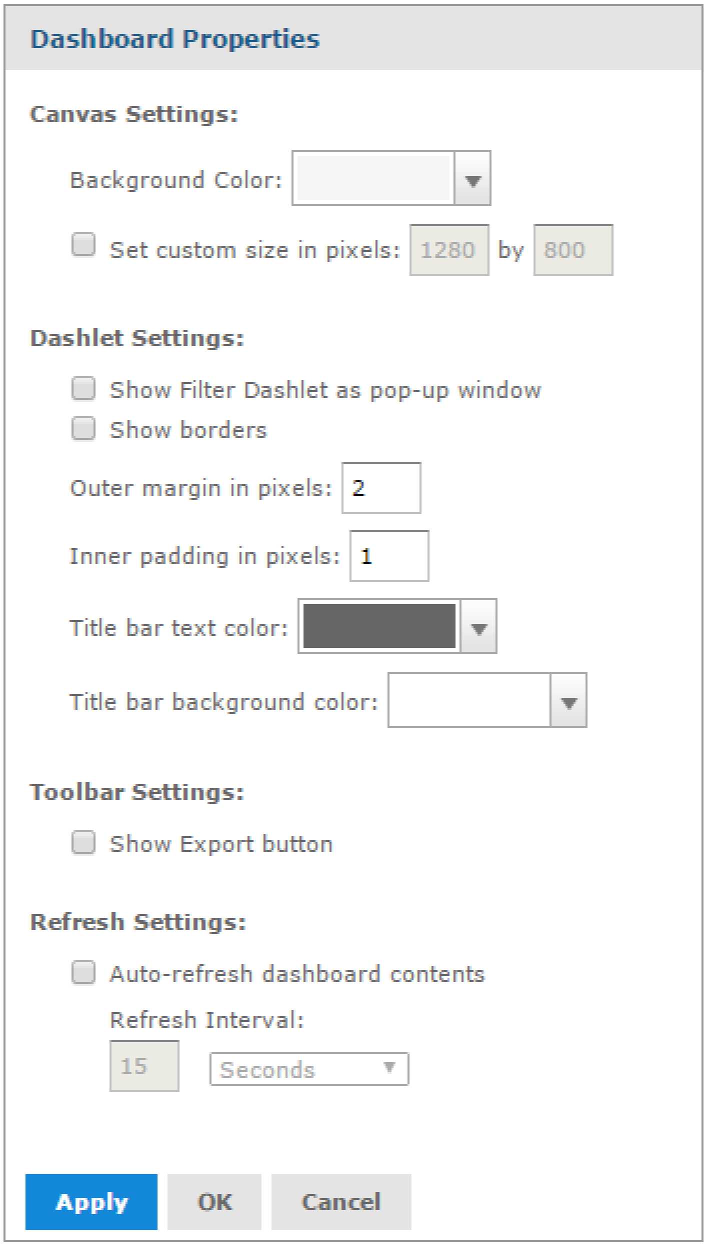
1. Click 👁. The dashboard opens in display mode.
2. To close the preview and return to the Dashboard Designer, click 👁.

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.

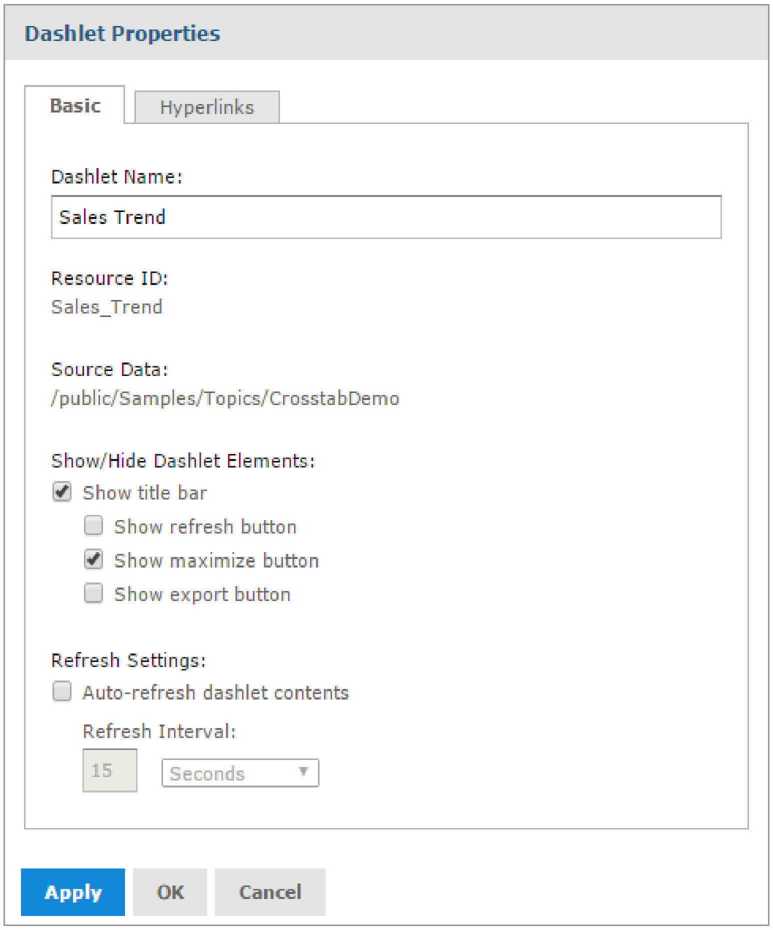
2.2.5 Dashlet Properties

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

2.2.5.1 Basic Properties for Charts, Crosstabs, Tables, Reports, and Ad Hoc Views Dashlet



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 Basic Tab Properties

- **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. Scale to Fit is available for dashlets for tables, crosstabs, reports, and existing Ad Hoc view tables and crosstabs. Scale to Fit is not available for charts or existing Ad Hoc view charts.
- **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.

2.2.5.2 Properties on the Hyperlink tab

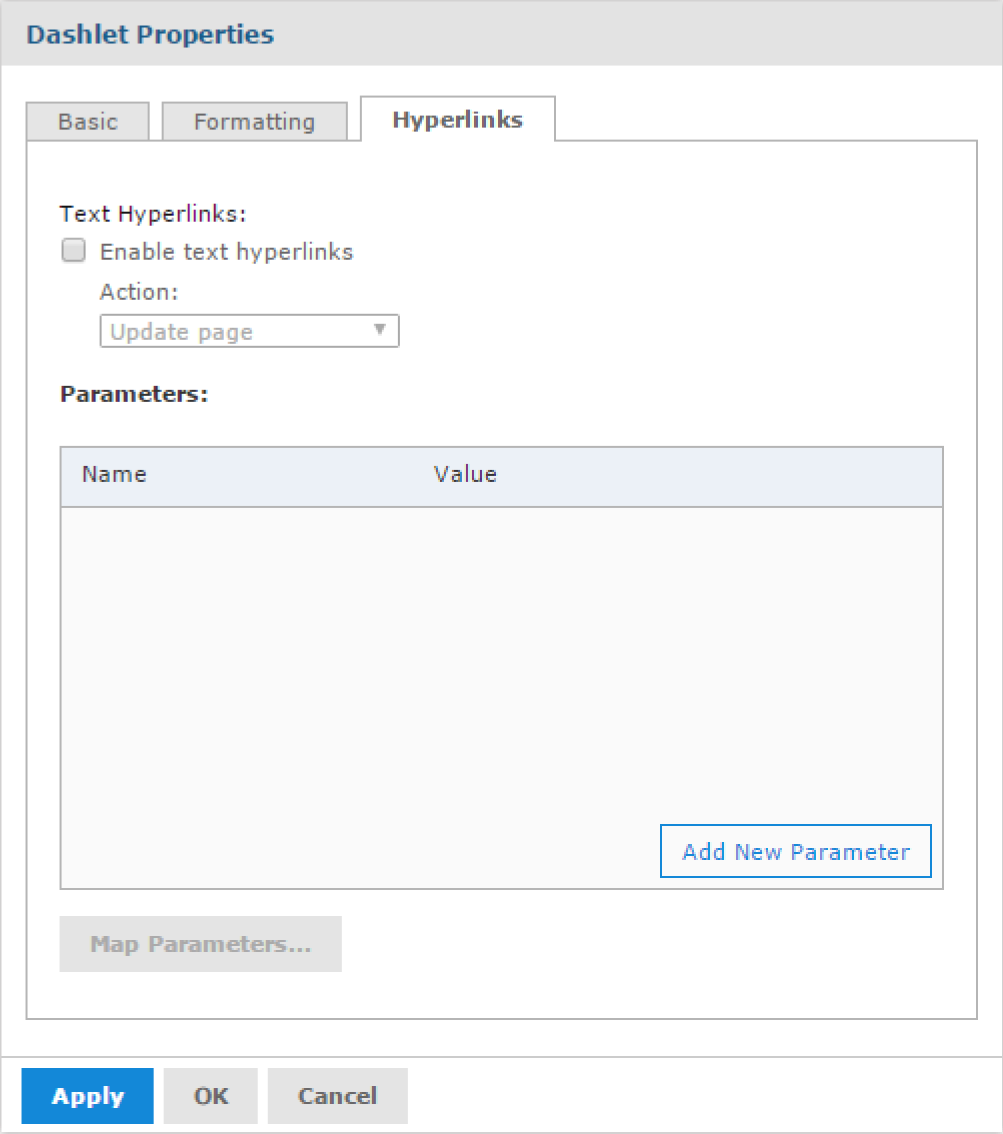


Figure 2-6 Hyperlinks Tab Properties

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 page**: Select this option to update the dashboard's contents when a user clicks on a link in the dashlet.
  - **Open new page**: Select this to have the dashlet open a web page or report, dashboard, or an Ad Hoc view in the repository in a new browser tab or window 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 44** for more information about adding parameters to hyperlinks.

- **Replace page**: Select this option to replace the current dashboard page with 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. The syntax for links and parameters is the same as the **Open new page** option.
  If the dashlet replaces the page with an Ad Hoc view, it will open in the Ad Hoc View Designer. A **Back** button appears on the toolbar after the dashlet opens a report or dashboard that will take you back to the previous dashboard.

- **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.

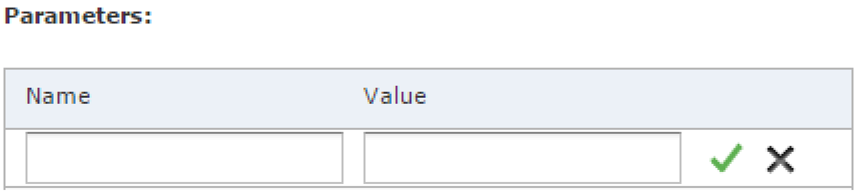
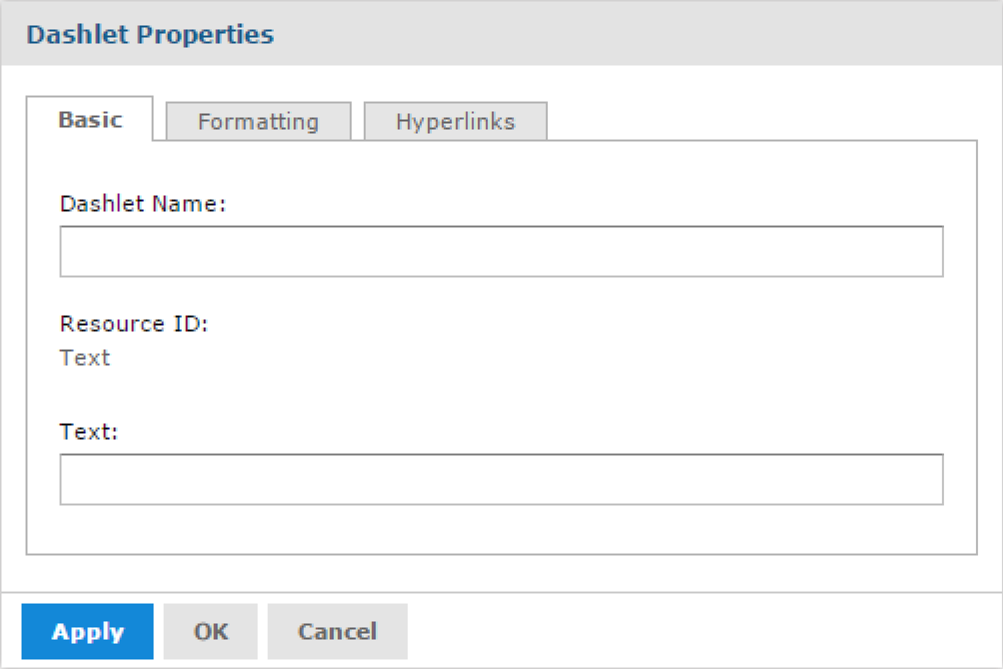


Figure 2-7 Adding a New Parameter to a Dashlet

- **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.

2.2.5.3 Basic Properties for Text Dashlets



Dashlet Properties

Dashlet Name:

Resource ID:
Text

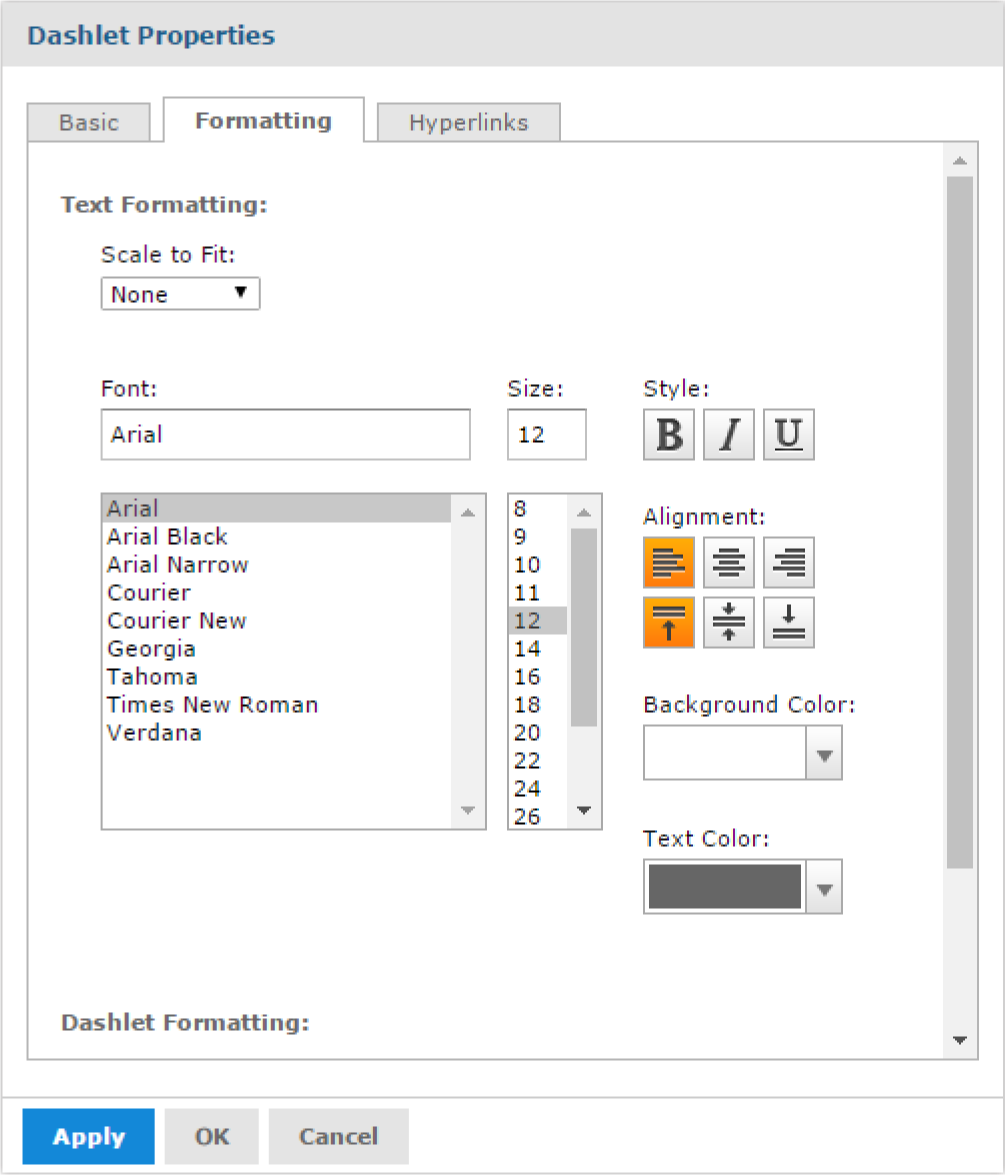
Text:

Figure 2-8 Basic Tab for Text Dashlets

- **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.

2.2.5.4 Formatting Properties for Text Dashlets

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



Dashlet Properties

Text Formatting:

Scale to Fit: None

Font: Arial

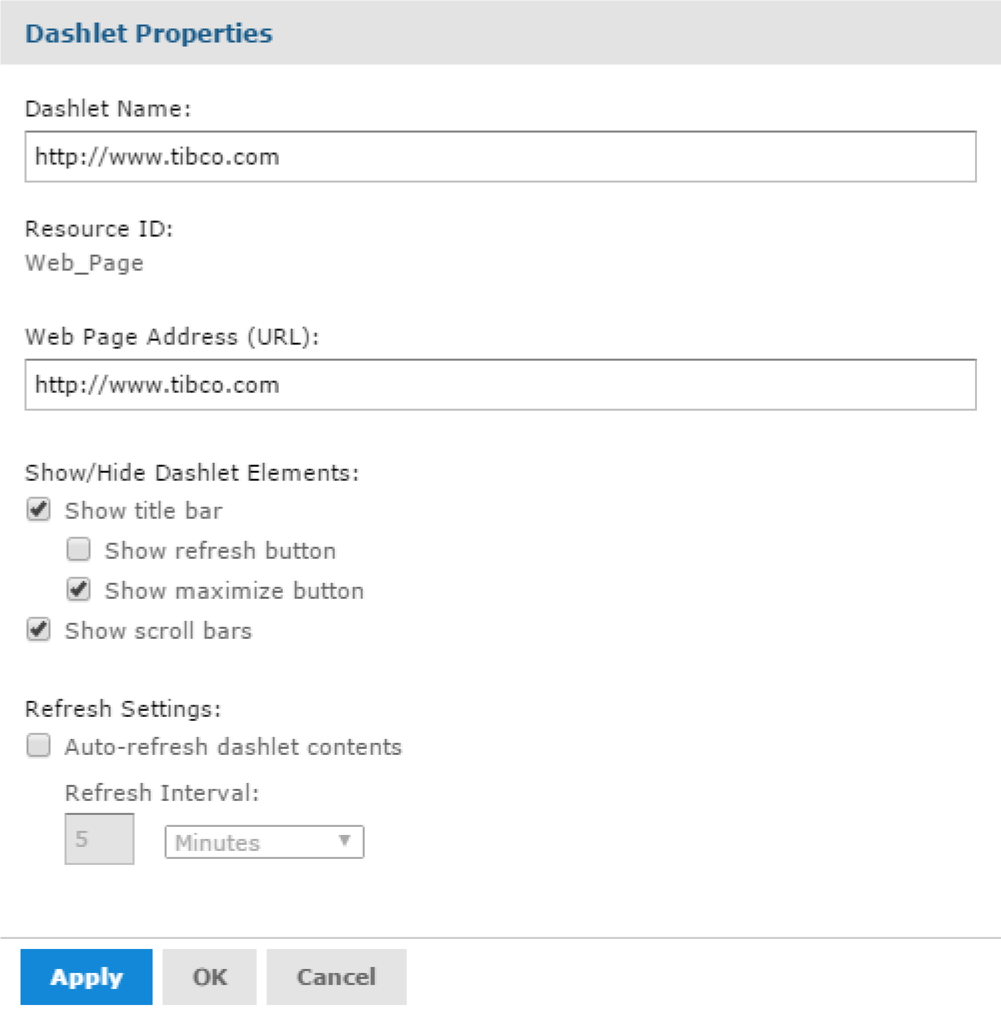
Size: 12

Dashlet Formatting:

Figure 2-9 Formatting Tab for Text Dashlets

- **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.

2.2.5.5 Basic Properties for Web Page Dashlets



Dashlet Properties

Dashlet Name:

http://www.tibco.com

Resource ID:

Web_Page

Web Page Address (URL):

http://www.tibco.com

Show/Hide Dashlet Elements:

☑ Show title bar

☐ Show refresh button

☑ Show maximize button

☑ Show scroll bars

Refresh Settings:

☐ Auto-refresh dashlet contents

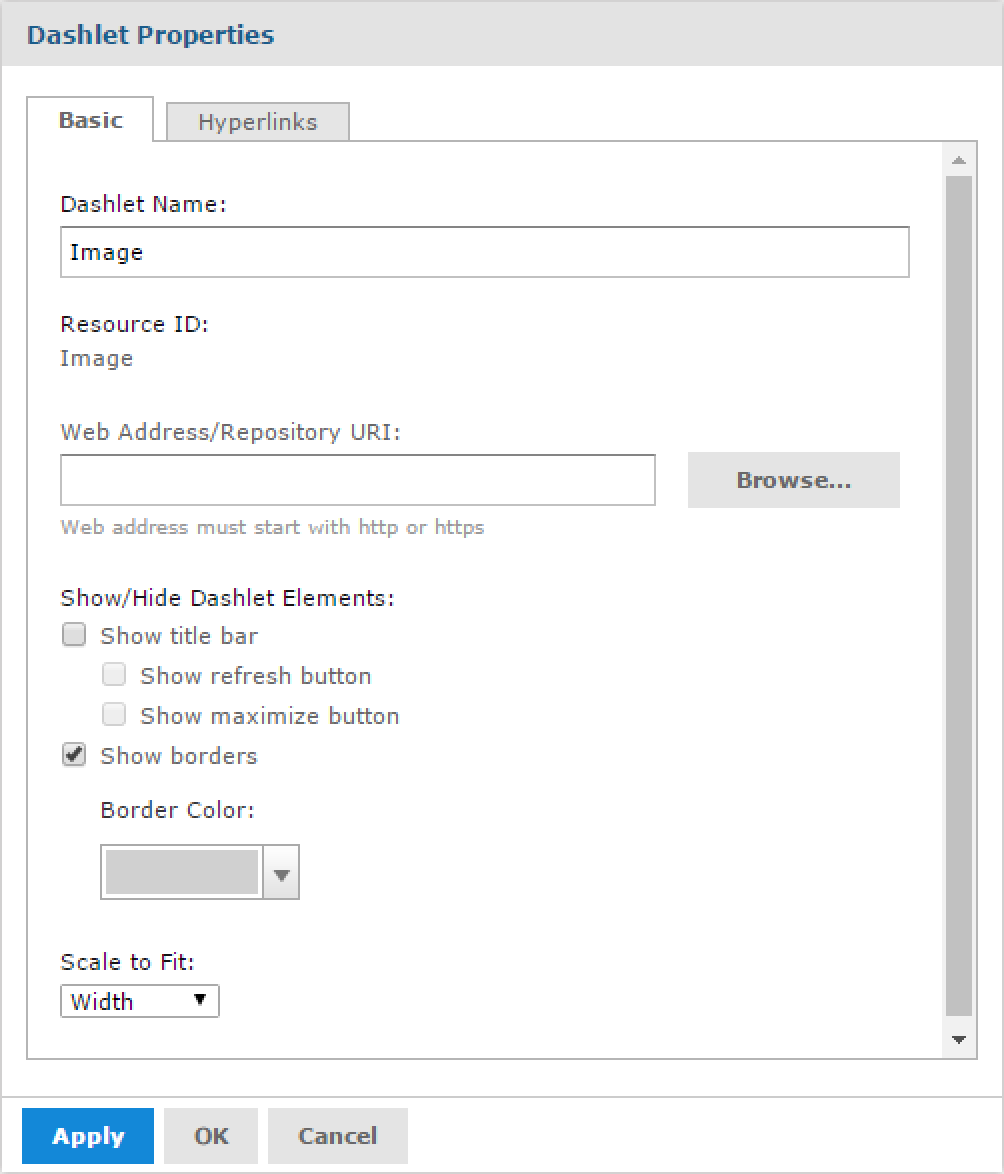
Refresh Interval:

5 Minutes

Figure 2-10 Basic Tab for Web Page Dashlets

- **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.

2.2.5.6 Basic Properties for Image Dashlets



Dashlet Name:
Image

Resource ID:
Image

Web Address/Repository URI:

Web address must start with http or https

Show/Hide Dashlet Elements:

□ Show title bar
- □ Show refresh button
- □ Show maximize button

☑ Show borders

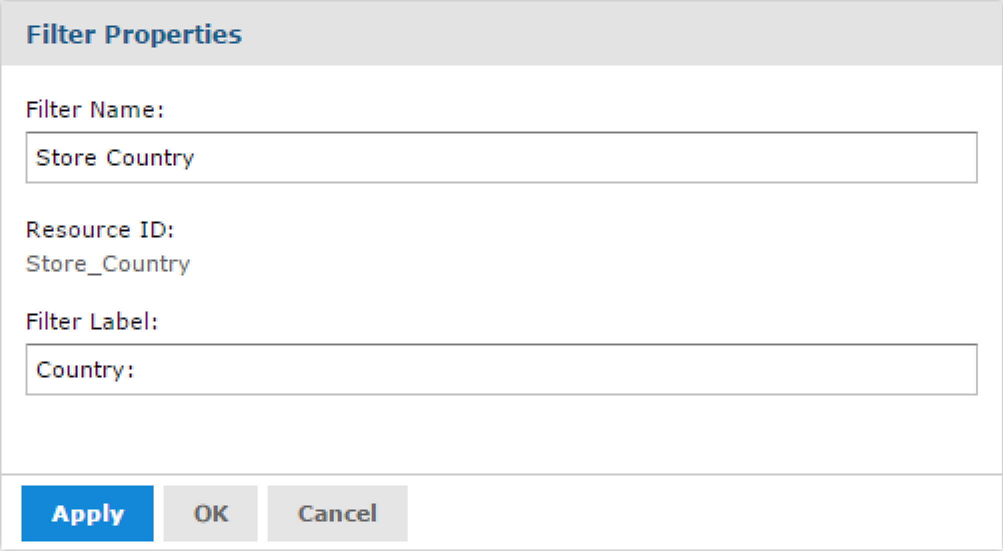
Border Color:

Scale to Fit:
Width

Figure 2-11 Basic Tab for Image Dashlets

- **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.

2.2.5.7 Properties for Input Control Dashlets



Filter Properties

Filter Name: Store Country

Resource ID: Store_Country

Filter Label: Country:

Figure 2-12 Input Control Dashlet Properties

- **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.

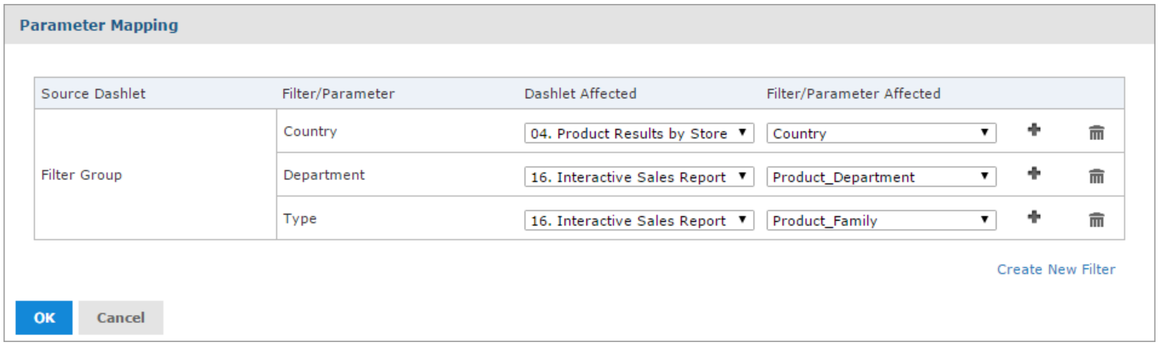
2.2.6 Parameter Mapping

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.



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 2-13 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.

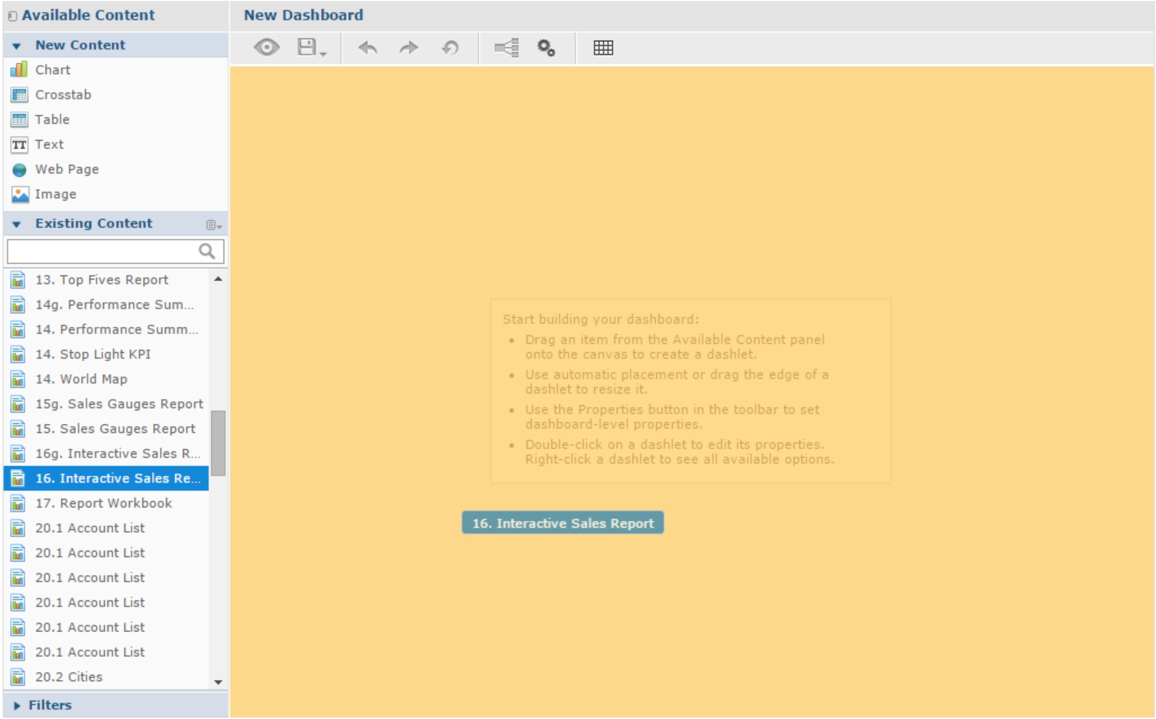
2.3 Creating a Dashboard

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 2-14 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.

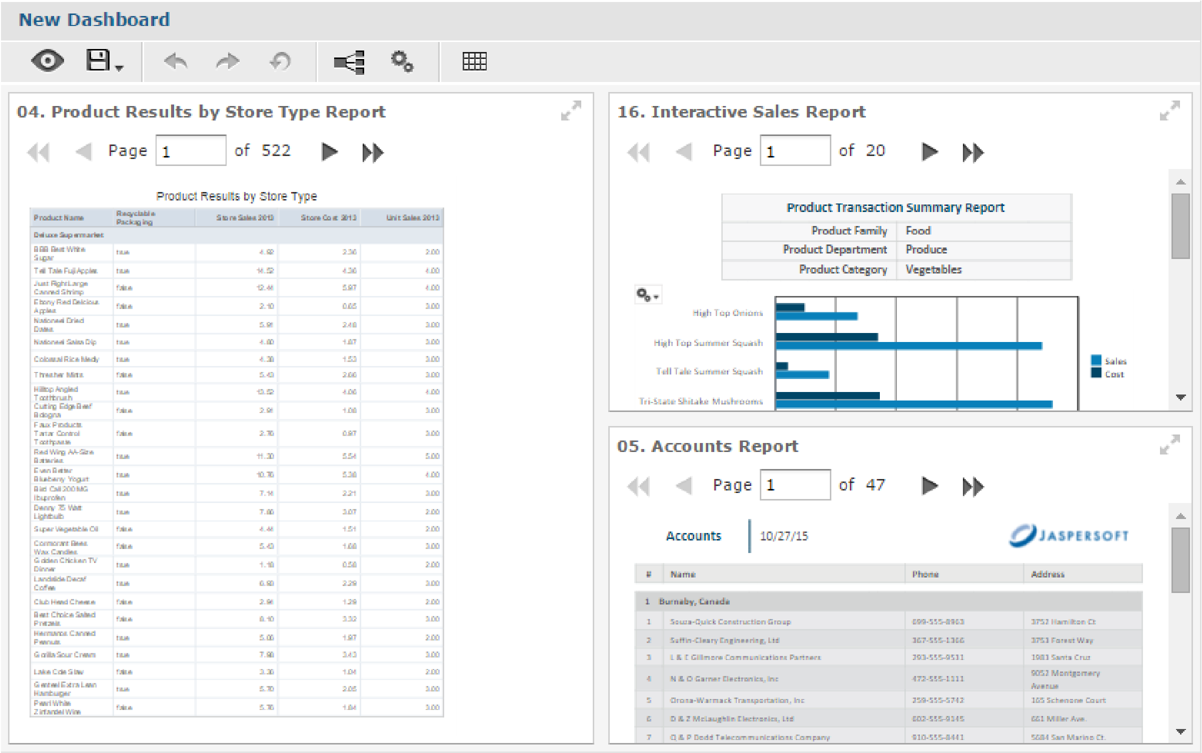
The Product Results by Store Type Report appears on the canvas, next to the Interactive Sales Report. Both report dashlets are sized to fit side-by-side on the canvas.

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 2-15** you can see how the dashboard canvas looks at this point.



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	13.52	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	7.86	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	1.16	0.58	2.00
Landslide Decaf Coffee	true	6.90	2.29	3.00
Club Head Cheese	false	2.94	1.29	2.00
Best Choice Salted Pretzels	false	8.10	3.32	3.00
Hermanos Canned Peanuts	true	5.06	1.87	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

Product Transaction Summary Report

Product Family	Food
Product Department	Produce
Product Category	Vegetables

Accounts | 10/27/15

#	Name	Phone	Address
1	Burnaby, Canada		
1	Souza-Quick Construction Group	699-555-8903	3752 Hamilton Ct
2	Suffin-Cleary Engineering, Ltd	367-555-1366	3753 Forest Way
3	L & E Gillmore Communications Partners	393-555-9531	1983 Santa Cruz
4	N & O Garner Electronics, Inc	472-555-1111	9052 Montgomery Avenue
5	Orona-Warmack Transportation, Inc	259-555-5742	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 2-15 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 **5.1.1, "Ad Hoc Sources: Topics, Domains, and OLAP Connections," on page 93**.
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 5, "Working with the Ad Hoc Editor," on page 91**.
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 29** 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 29** 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 29** 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.

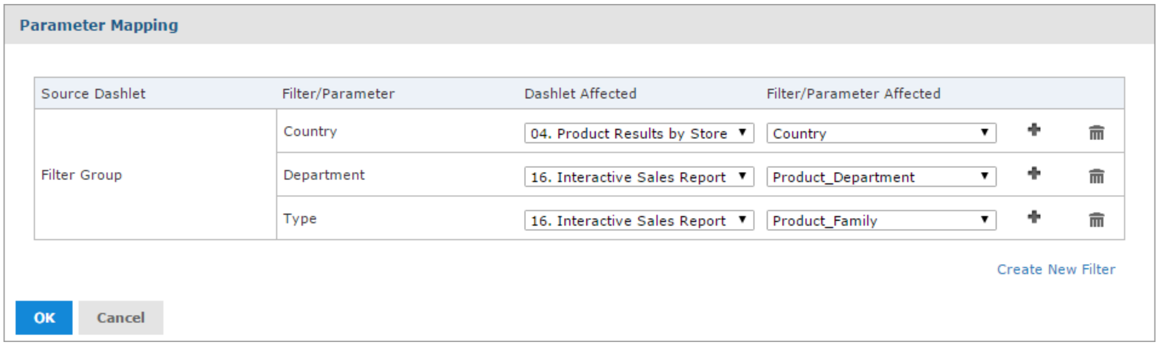
The Country input control and its label appear above the Product Results by Store type report on the canvas.

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 2-16 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{Store 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{Store 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:

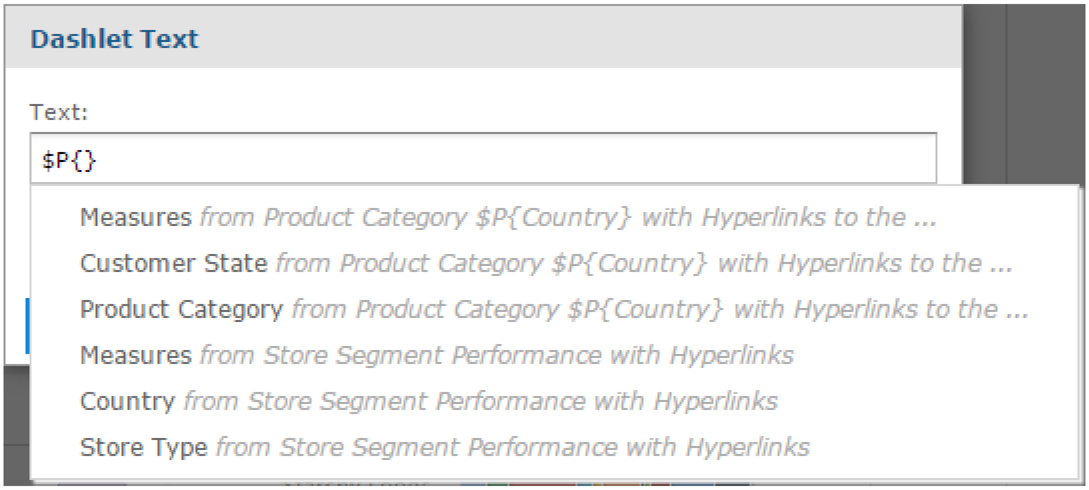
  ```
  repo:/public/Samples/Reports/05._Unit_Sales_Trend?$P{"c_country_1=", Store
  Country, "&"}
  ```

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
```

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 2-17 Text dashlet showing available parameters

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 08. Key Performance Metric Trend.
4. Click and drag the report onto the Dashboard Canvas. Note that the **Filters** section now includes the 08. Key Performance Metric Trend folder.

Next, add the filter input control to the dashboard:

1. In the **Filters** section, expand the 08. Key Performance Metric Trend folder.

   The input controls associated with the 08. Key Performance Metric Trend report appear.
2. Drag the Store 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:

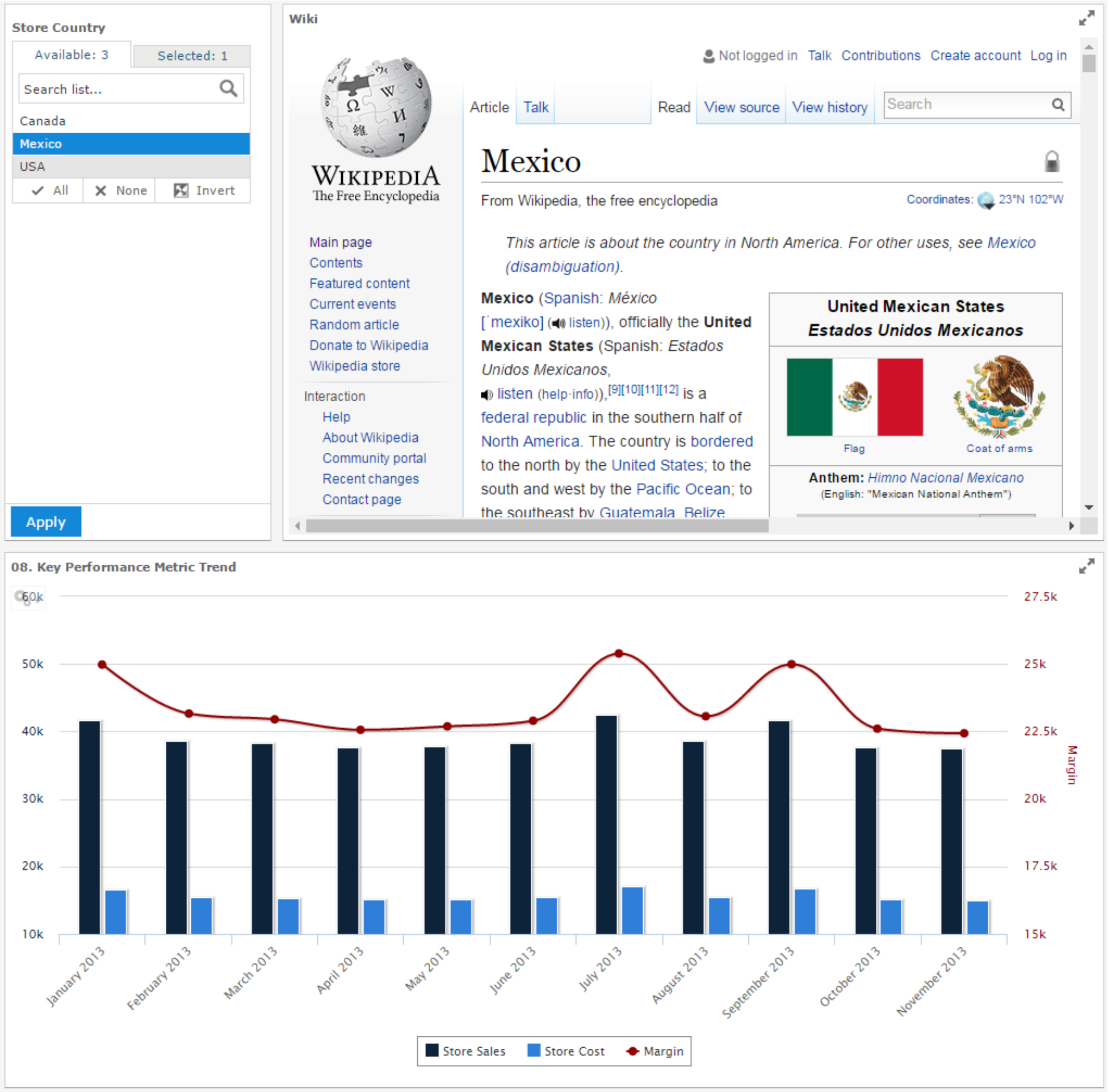
   ```
   http://en.wikipedia.org/wiki/$P{Store Country}
   ```
3. Click **OK**.
4. Right-click the dashlet and select **Properties**.
5. Change the Dashlet Name to **Wiki**.
6. Click **OK**.
7. Click ⇶ on the Ad Hoc menu to open Parameter Mapping.
8. In the **Store Country** filter group, click ✚.

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

Finally, preview the new dashboard functionality:

1. Click 👁 to preview the dashboard.
2. Click in the **Store Country** text box to display the available countries.
3. Select **Mexico** from the values list in the Store Country input controls, and click **Apply** at the bottom of the dashlet. The data in the Key Performance Metric Trend report is updated to display only information about Mexico, and the wiki dashlet displays the Wikipedia page for Mexico.

In **Figure 2-18** you can see how the dashboard preview looks at this point:



Mexico

From Wikipedia, the free encyclopedia

Coordinates: 23°N 102°W

This article is about the country in North America. For other uses, see Mexico (disambiguation).

Mexico (Spanish: *México* [ˈmexiko] (listen)), officially the **United Mexican States** (Spanish: *Estados Unidos Mexicanos*, listen (help·info)),[9][10][11][12] is a federal republic in the southern half of North America. The country is bordered to the north by the United States; to the south and west by the Pacific Ocean; to the southeast by Guatemala, Belize

United Mexican States
Estados Unidos Mexicanos

Flag
Coat of arms

Anthem: *Himno Nacional Mexicano*
(English: "Mexican National Anthem")

Figure 2-18 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 `Key Performance Metric Trend 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 46**. The example also shows how to add a parameter to a hyperlink.

First, open the dashboard:

1. If the Key Performance Metric Trend dashboard created in **2.4.1, "Creating a Web Page Dashlet," on page 46** (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 Key Performance Metric Trend dashboard appears in the designer.

Create a chart hyperlink:

1. Right-click on the 08. Key Performance Metric Trend dashlet and select **Properties...**.

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

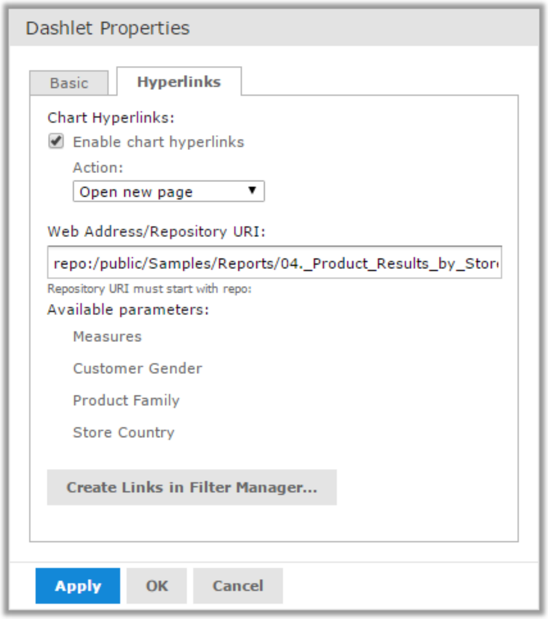


Chart Hyperlinks:

☑ 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

Figure 2-19 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. Return to the tab containing Key Performance Metric Trend dashboard and click 👁 to return to the Dashboard Designer.

Add a parameter:

1. Right-click on the Key Performance Metric Trend dashboard 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 44**.

     The link is as follows:

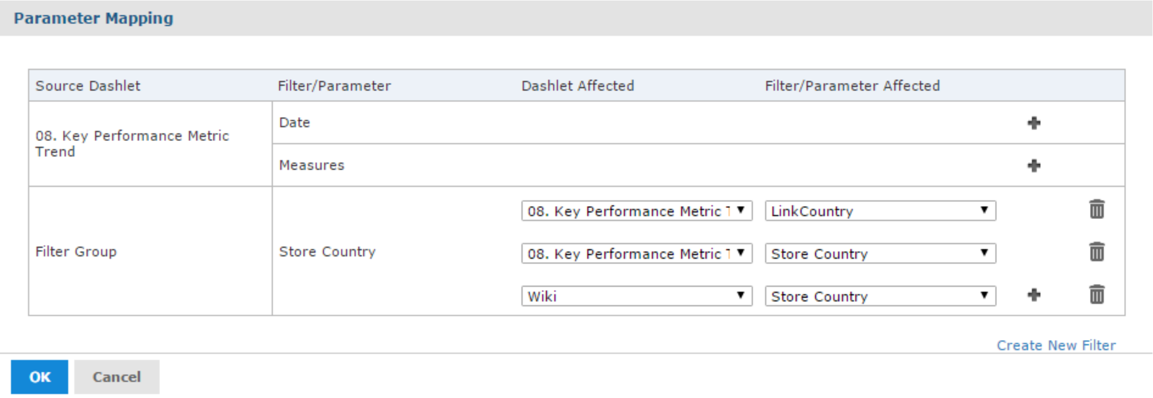
     ```
     repo:/public/Samples/Reports/04._Product_Results_by_Store_Type_Report?sales__
     store__store_contact__store_country_1=$P{LinkCountry}
     ```

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 44**.

3. Click **Map Parameters**. Click **Yes** when prompted.

   Parameter Mapping is displayed.



Source Dashlet	Filter/Parameter	Dashlet Affected	Filter/Parameter Affected	
08. Key Performance Metric Trend	Date			✚
	Measures			✚
Filter Group	Store Country	08. Key Performance Metric ▾	LinkCountry ▾	🗑
		08. Key Performance Metric ▾	Store Country ▾	🗑
		Wiki ▾	Store Country ▾	✚ 🗑

Figure 2-20 Parameter Mapping

4. In the **Store 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... dropdown 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 **Store 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 39**.
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 27** and **"Dashlet Properties" on page 29** 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 200**.

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`.
  - `$Date{DD/MM/YYYY}`. For example, January 15, 2016 is displayed as `15/01/2016`.
  - `$Date{dddd MMMM DD, YYYY}`. For example, January 15, 2016 is displayed as `Friday January 15, 2016`.
- `$Time`: Displays the current time in the 12 hour format with `am/pm`, but without the leading `0` for hours. For example, `1:25 pm`. You can customize how the time is displayed using the following parameters:
  - `$Time{HH:mm}`. Displays the time in the 24 hour time format. For example, 1:25 pm is displayed as `13:25`.
  - `$Time{hh:mm a}`. Displays the current time in the 12 hour format with `am/pm` and the leading `0` for hours. For example, `01:25 pm`.
  - `$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`.
  - `$Time{hh:mm:ss a}`. Displays the current time in the 12 hour format with seconds, `am/pm`, and the leading `0` for hours. For example, 1:25:30 pm is displayed as `01:25:30 pm`.
  - `$Time{HH:mm:ss z}`. Displays the current time in the 24 hour time format with seconds and the time zone. The time zone is selected on the login page. For example, 1:25:30 pm in the Pacific Standard time zone is displayed as `13:25:30 PST`.

- `$Time{hh:mm:ss a z}`. Displays the current time in the 12 hour format with seconds, `am/pm`, the time zone, and the leading `0` for hours. The time zone is selected on the login page. For example, 1:25:30 pm in the Pacific Standard time zone is displayed as `01:25:30 pm PST`.

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.

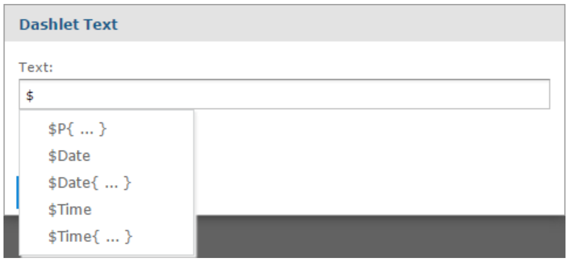


Figure 2-21 Time-Date Wildcard Auto-complete

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 Fusion 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 71**.
🔖	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

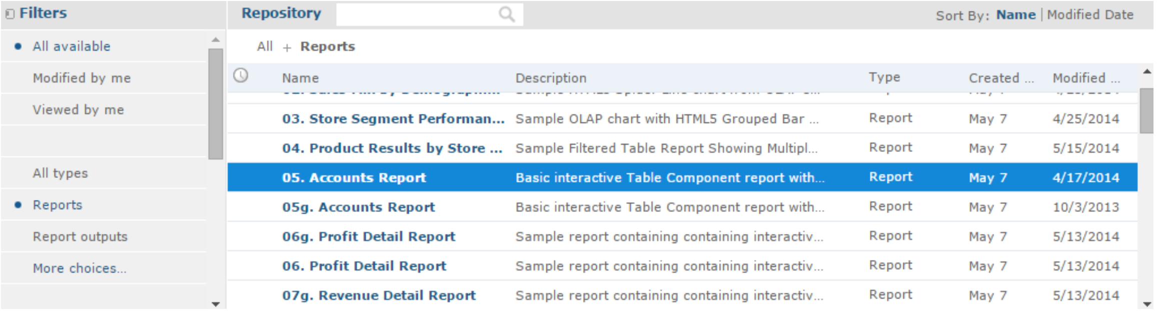
This section describes how to run a tabular report that lists account data.

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.

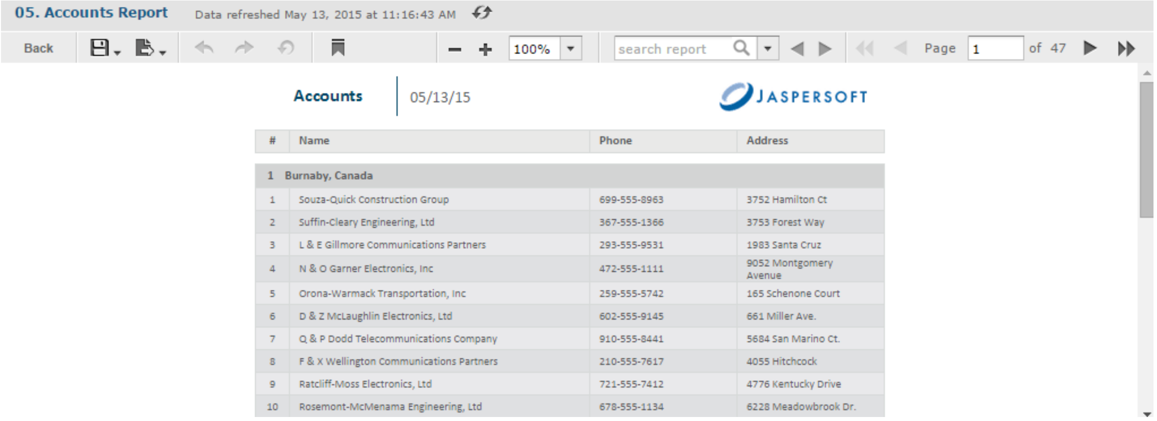
   The search results appear, listing your own files and other files that your user account has permission to view. If you log in as jasperadmin, 05. Accounts Report appears in the search results.



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**.



#	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 60** for more information.
4. Click **OK**. If asked, enter the input controls needed. See **“Using Input Controls” on page 158**.

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.

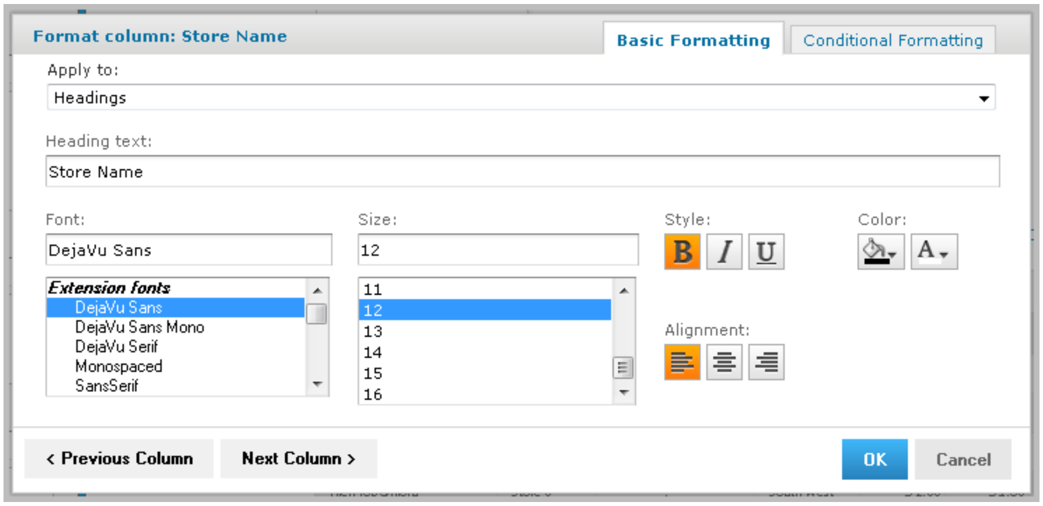
- For most PDF exports, you can use Actual Size, which supports a maximum size of 14400px by 14400px.
- For reports with an output height exceeding 14400 px, use a paginated report template that is wide enough for your report. For example, if you have a long report with width less than 842px, you can 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 62**.

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.

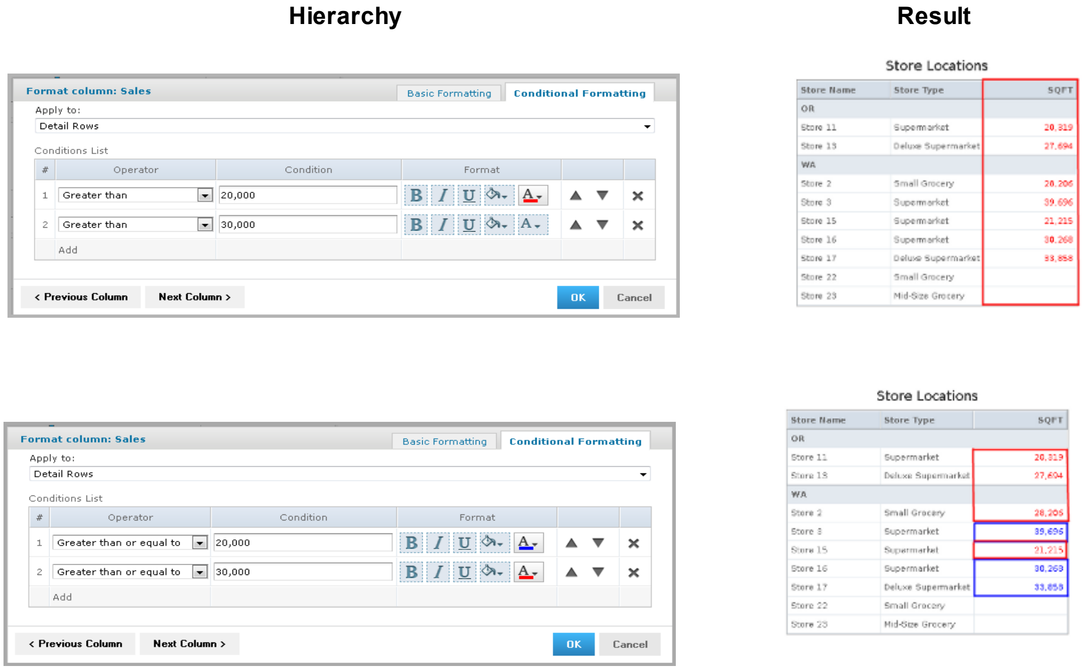
With conditional formatting, you can apply the formatting options listed in **"Column Formatting" on page 61**. However, it is a slightly more complex process than applying formatting options to entire columns. This section describes those complexities, including:

- 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 63**, 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

Result

Format column: Sales

Basic Formatting | Conditional Formatting

Apply to: Detail Rows

Conditions List

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

Add

< 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	

Format column: Sales

Basic Formatting | Conditional Formatting

Apply to: Detail Rows

Conditions List

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

Add

< 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	28,205
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.

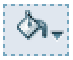
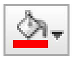
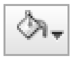
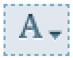
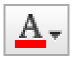
Both have two buttons at the top of the window, along with the color selection boxes.

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

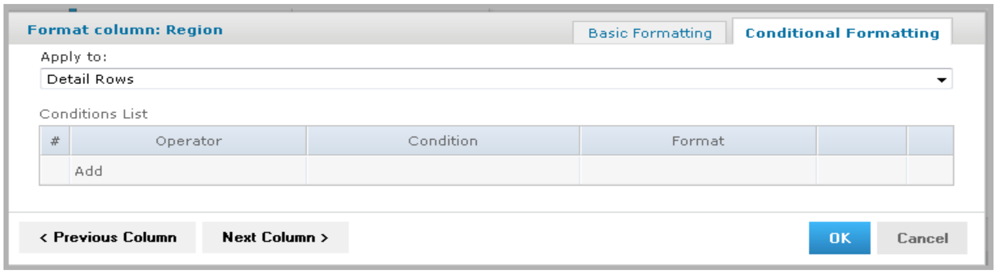
	Unchanged	Set	No Fill
Background Color			
Text Color			N/A

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 61** 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 63**.
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

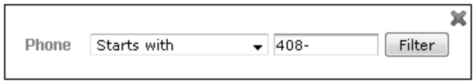
Numeric	Date	Text
Less than or equal to	Is before	Does not start with
Is between	Is on or after	Ends with
Is not between	Is after	Does not end with

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 58**, 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-`

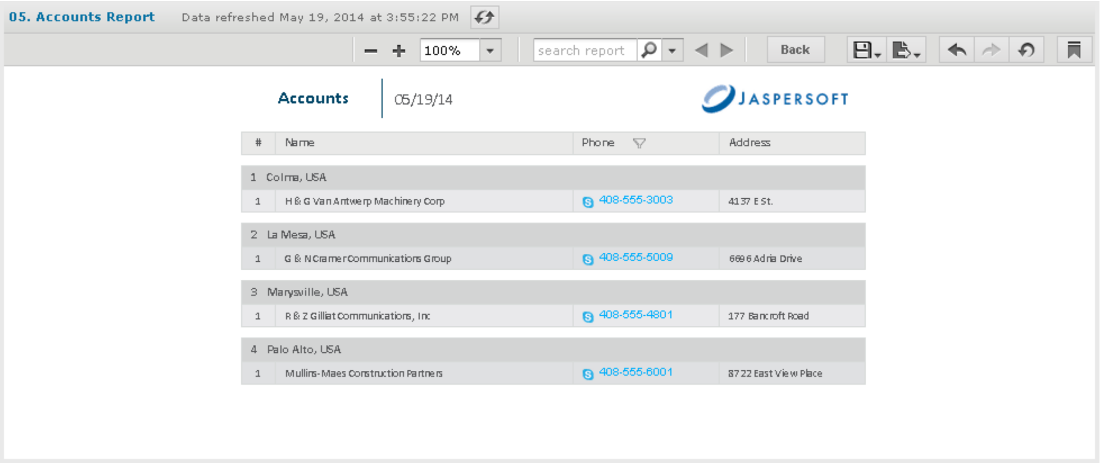


Phone | Starts with | 408- | Filter

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

100% search report Back

Accounts 05/19/14 JASPERSOFT

#	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.

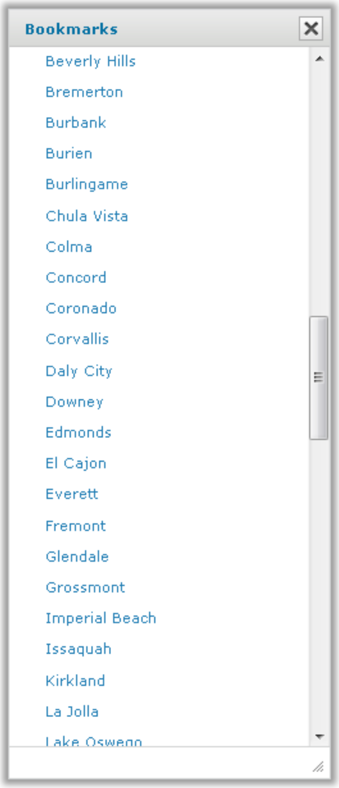


Figure 3-8 The Bookmarks Panel

- 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 60** 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 Fusion Chart

The JasperReports Server commercial editions support Fusion 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 HTML5 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.

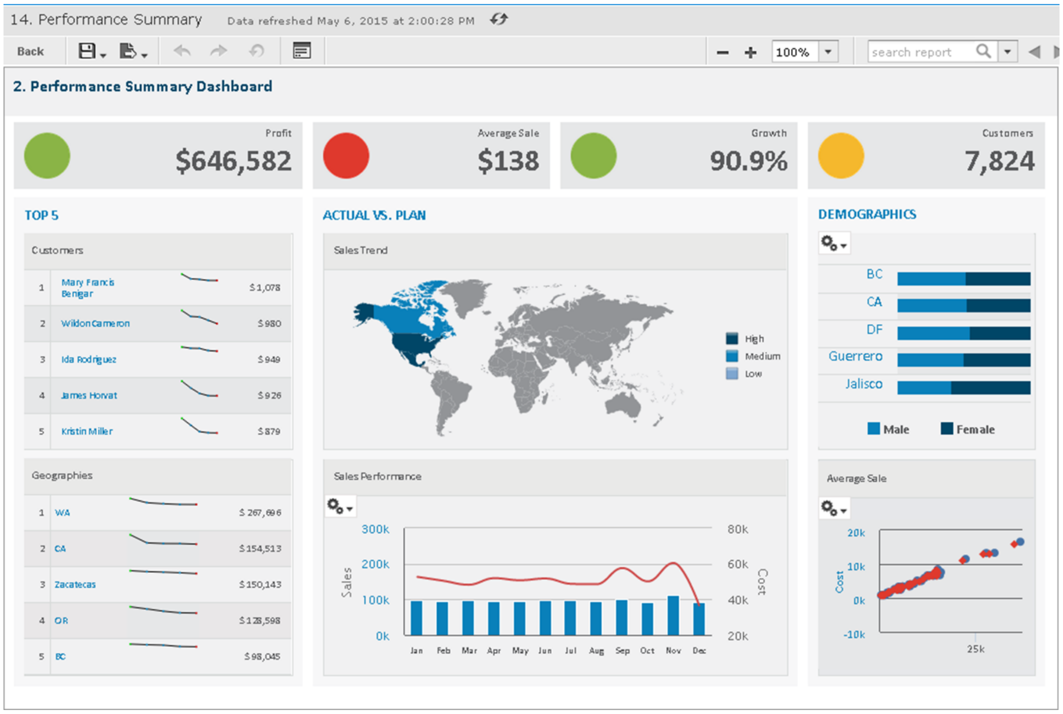
Flash is no longer recommended.

For configuration information, see the *JasperReports Server Administrator Guide* for more information.

Fusion charts are created in Jaspersoft Studio Professional as JRXML reports and uploaded to the repository as a report unit.

To find and run a 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

ACTUAL VS. PLAN

Sales Trend

Sales Performance

DEMOGRAPHICS

Average Sale

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 it has data, 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 171**.

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 58**), 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 74** for more information.

3.7.1 Simple Input Controls

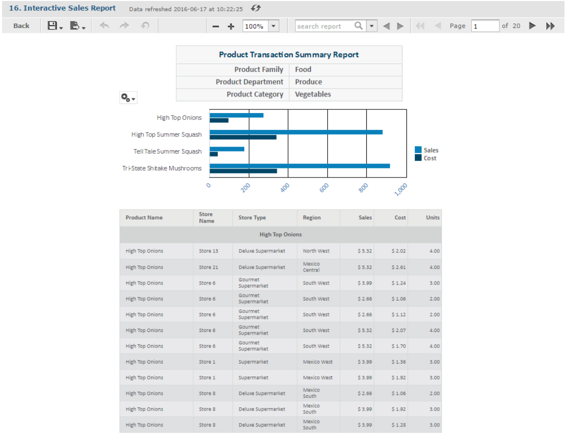
The 16. Interactive Sales Report example has several input controls:

- 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.

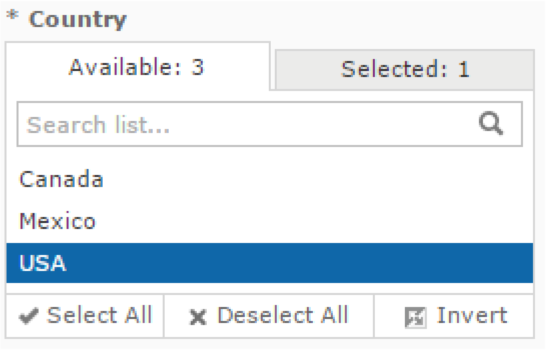


Figure 3-11 **Input Control Selection - Country**

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.

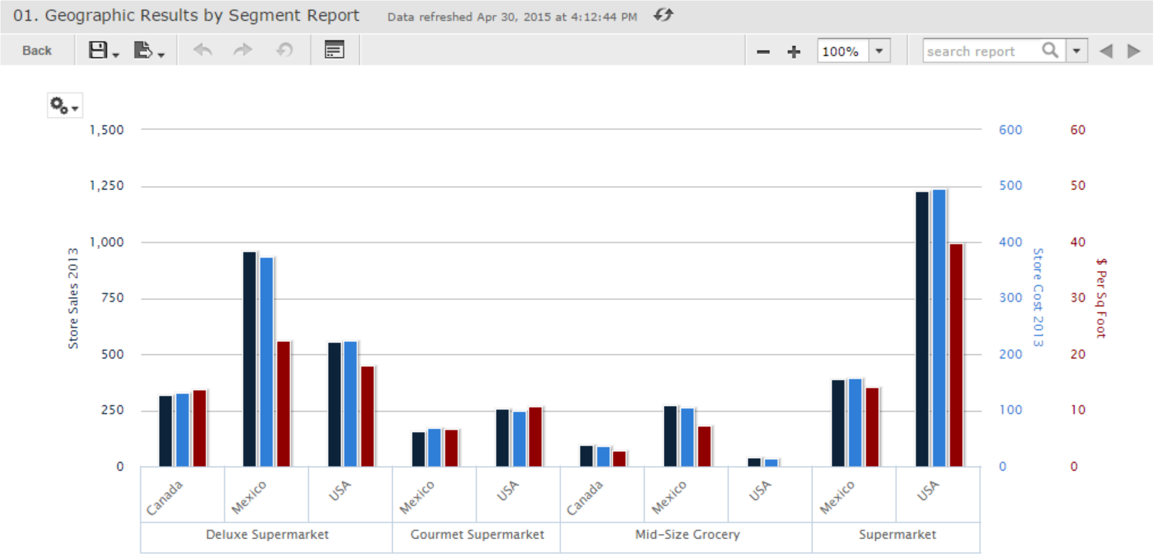


Figure 3-12 Geographic Results by Segment Report

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.

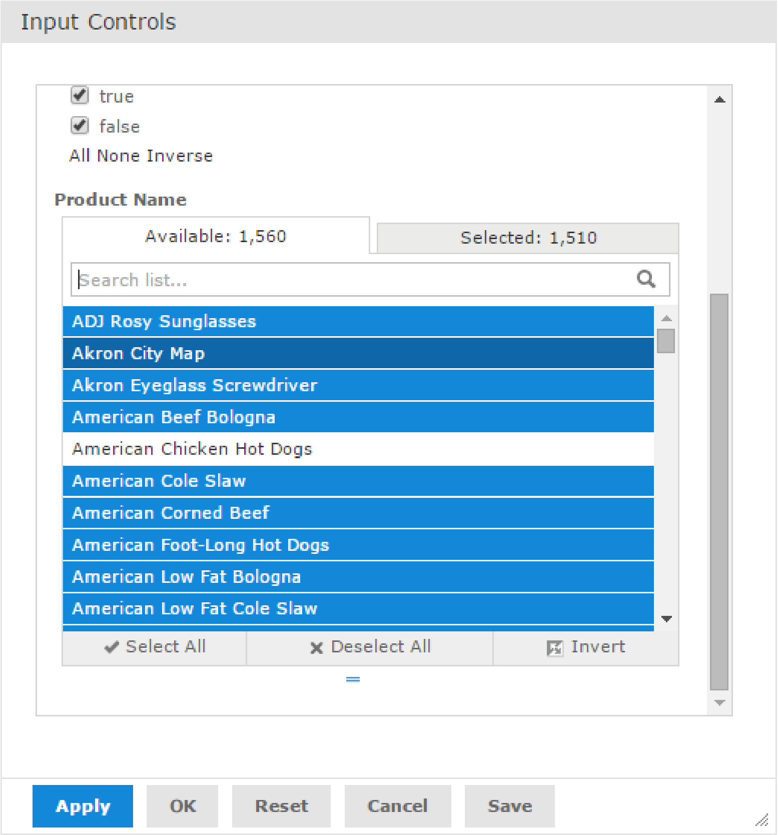
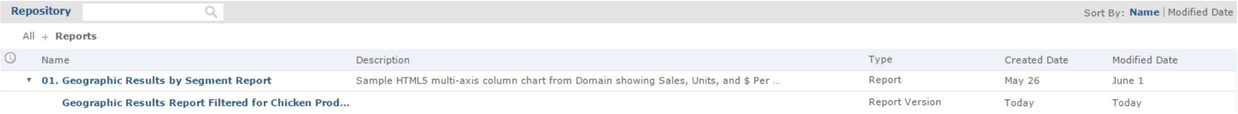


Figure 3-13 Multi-Input Control Selection

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 72**.

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.

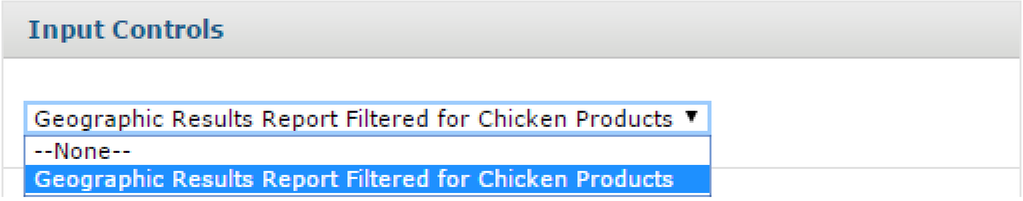

Input Controls
Geographic Results Report Filtered for Chicken Products ▼
--None--
Geographic Results Report Filtered for Chicken Products

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 196**.

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.

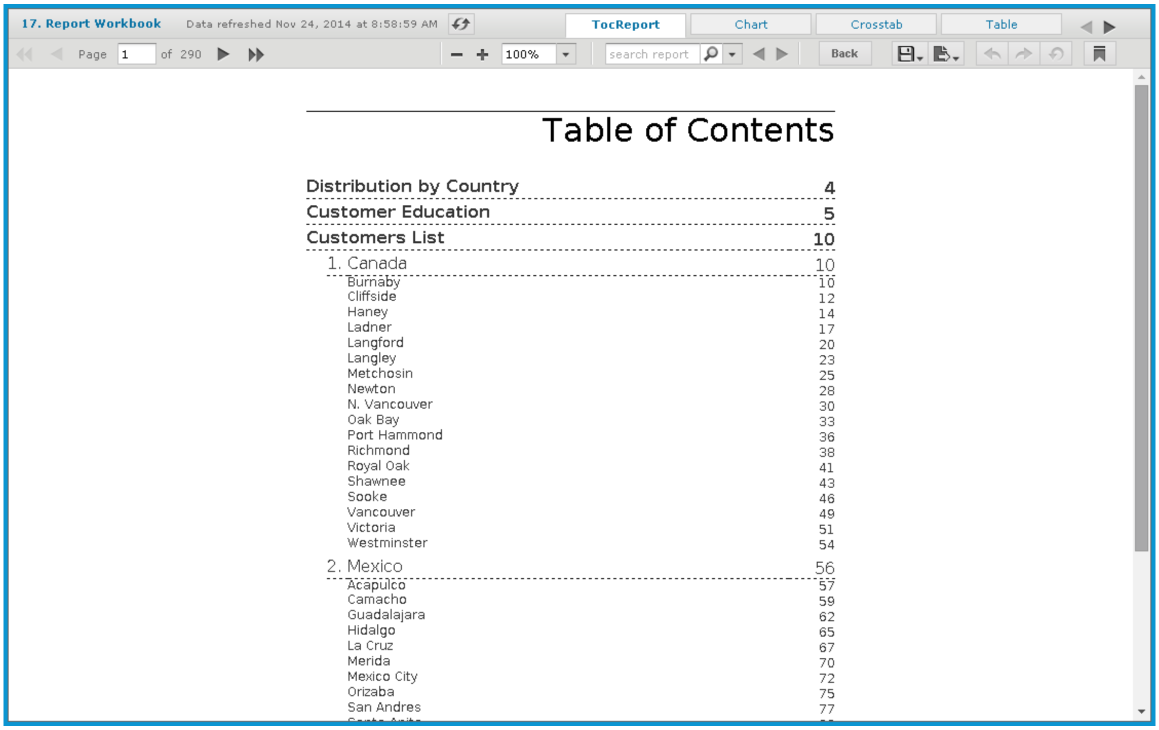


Table of Contents

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 4 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:

4.1 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.

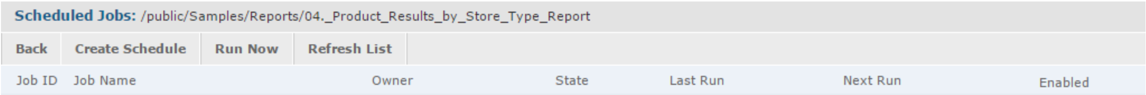
Sensitive data could be exposed to unauthorized users if you schedule a job as an administrative user with no data restrictions because the output will contain all requested data in the data source. Any user who receives the output can view all the data regardless of the user's access restrictions.

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.

4.2 Creating a Schedule

To create a schedule:

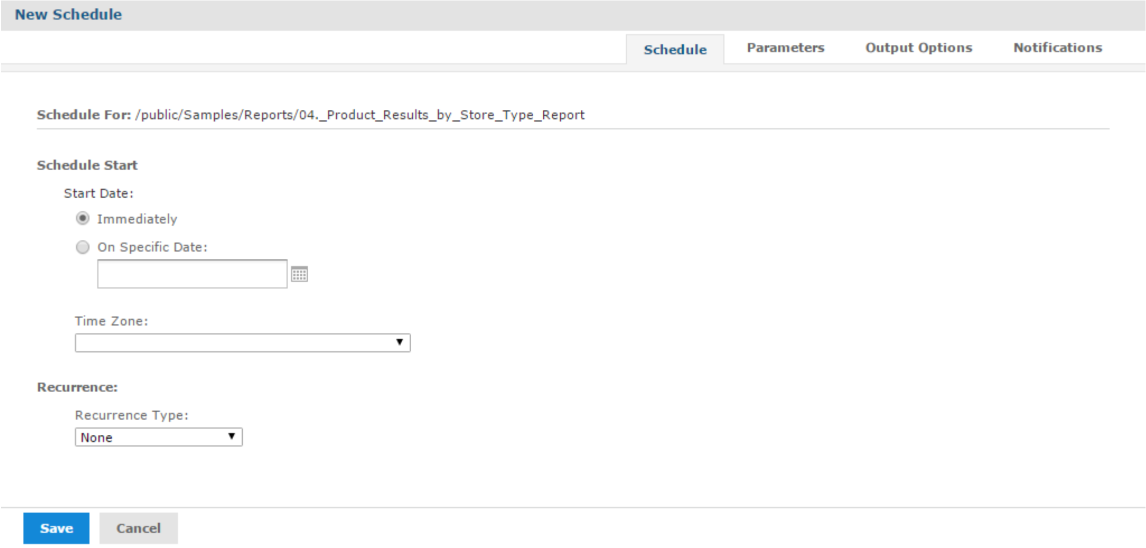
1. Locate the report or dashboard you want to schedule in the repository.
2. Right-click the report or dashboard and select **Schedule...** from the context menu, or if it already has a schedule, click the schedule icon 🕒 . The Scheduled Jobs page appears.



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

Figure 4-1 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

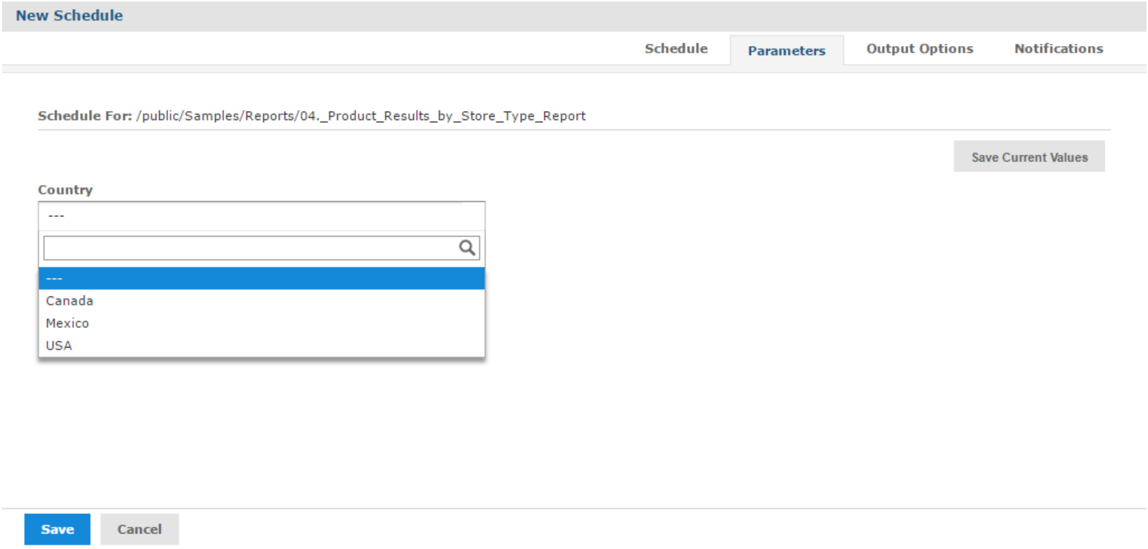
Figure 4-2 New Schedule Tab

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 84**. 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 4-3 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 4-3**. 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 71**.

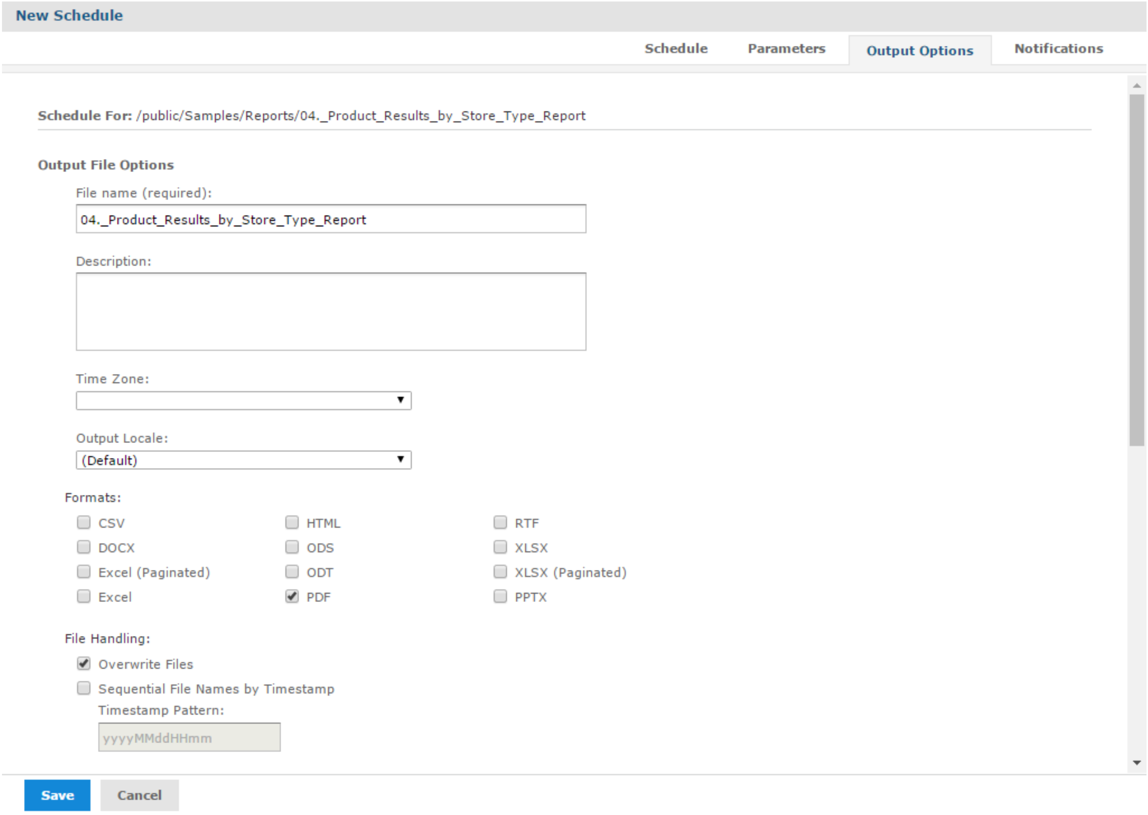
8. Choose a set of saved values, or set the input controls.
9. Click the Output Options tab and set the output format and location, as described in **Setting Output Options**.
10. Click the Notifications tab and set up email notifications, as described in **Setting Up Notifications**
11. Click **Save**. 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.

4.2.1 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 **"The Options Tab" on page 237**).
- **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

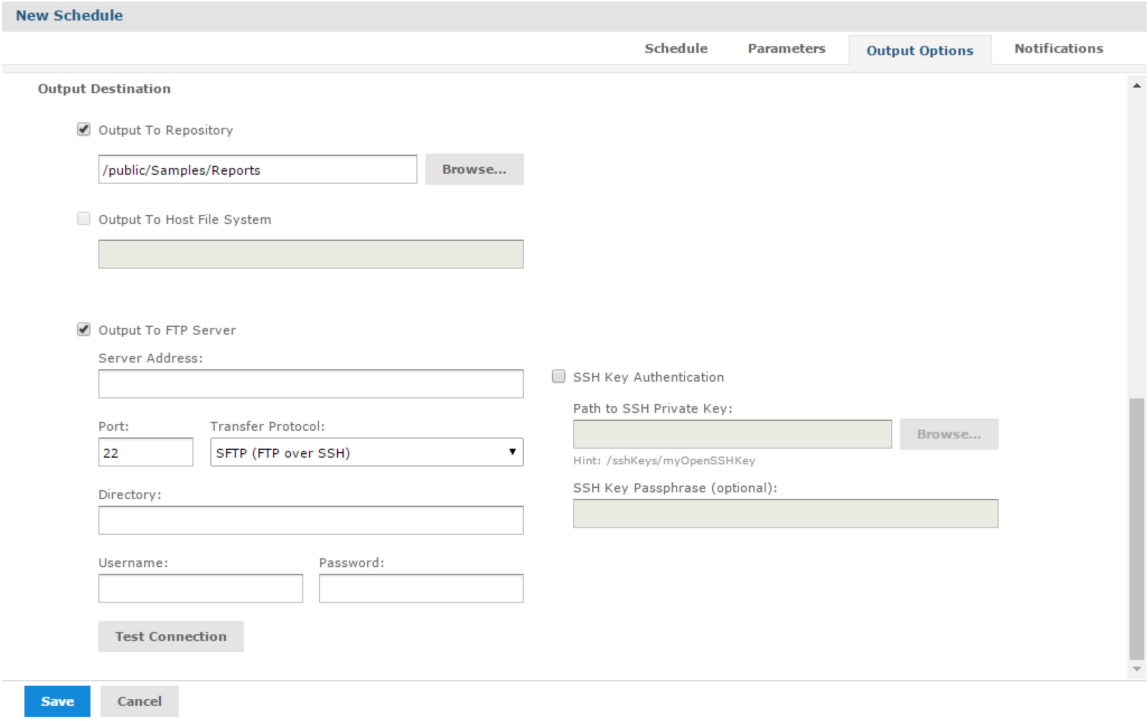
□ Sequential File Names by Timestamp

Timestamp Pattern: yyyyMMddHHmm

Figure 4-4 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 Destination

☑ Output To Repository

/public/Samples/Reports

☐ Output To Host File System

☑ Output To FTP Server

Server Address:

☐ SSH Key Authentication

Path to SSH Private Key:

Hint: /sshKeys/myOpenSSHKey

Port: 22

Transfer Protocol: SFTP (FTP over SSH)

Directory:

SSH Key Passphrase (optional):

Username:

Password:

Figure 4-5 Output Page for Scheduling a Report – Output Destination

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

4.2.2 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:

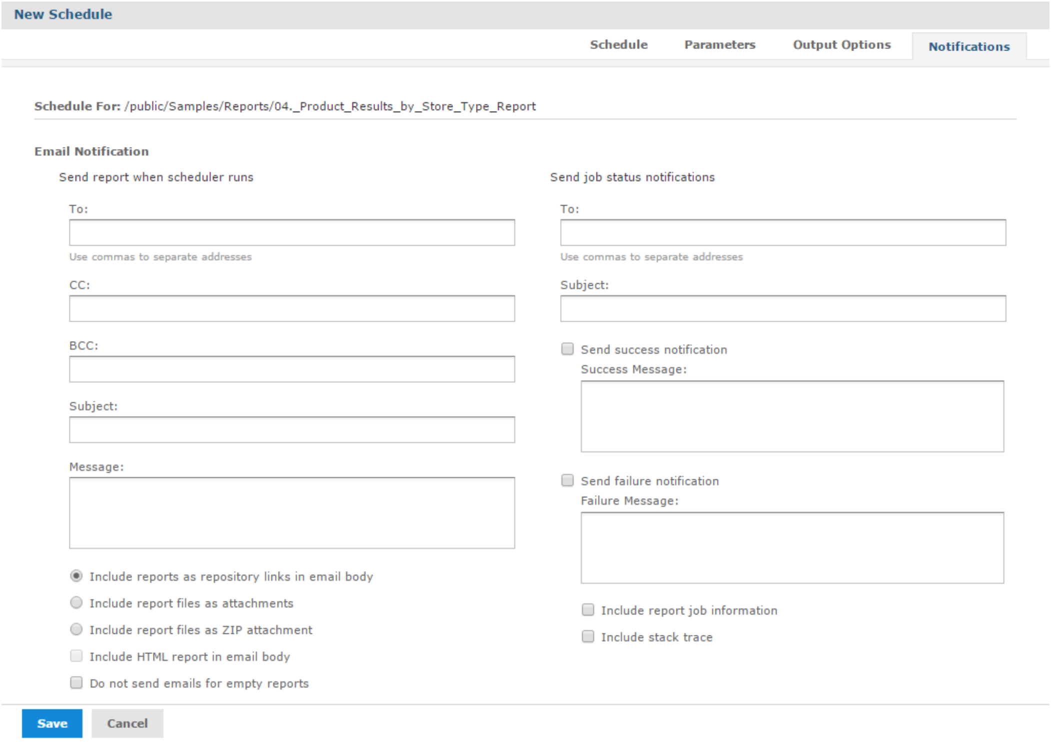
Enter one or more email addresses to send notification of job success or failure to administrators:

- **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.



New Schedule

Schedule | Parameters | Output Options | **Notifications**

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

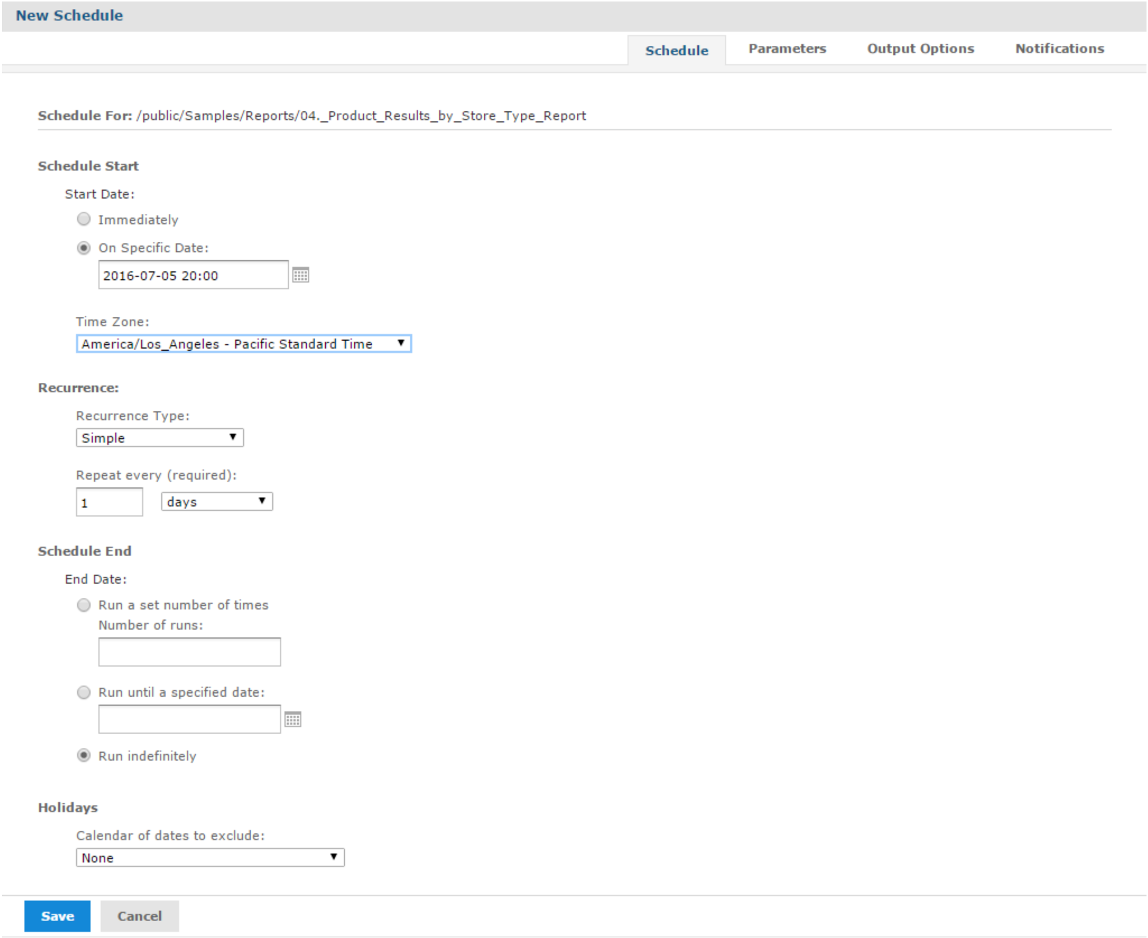
Figure 4-6 Notifications Page for Scheduling a Report

4.2.3 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 4-7** you can see an example of how to set simple recurrence.



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 4-7 Simple Recurrence Settings

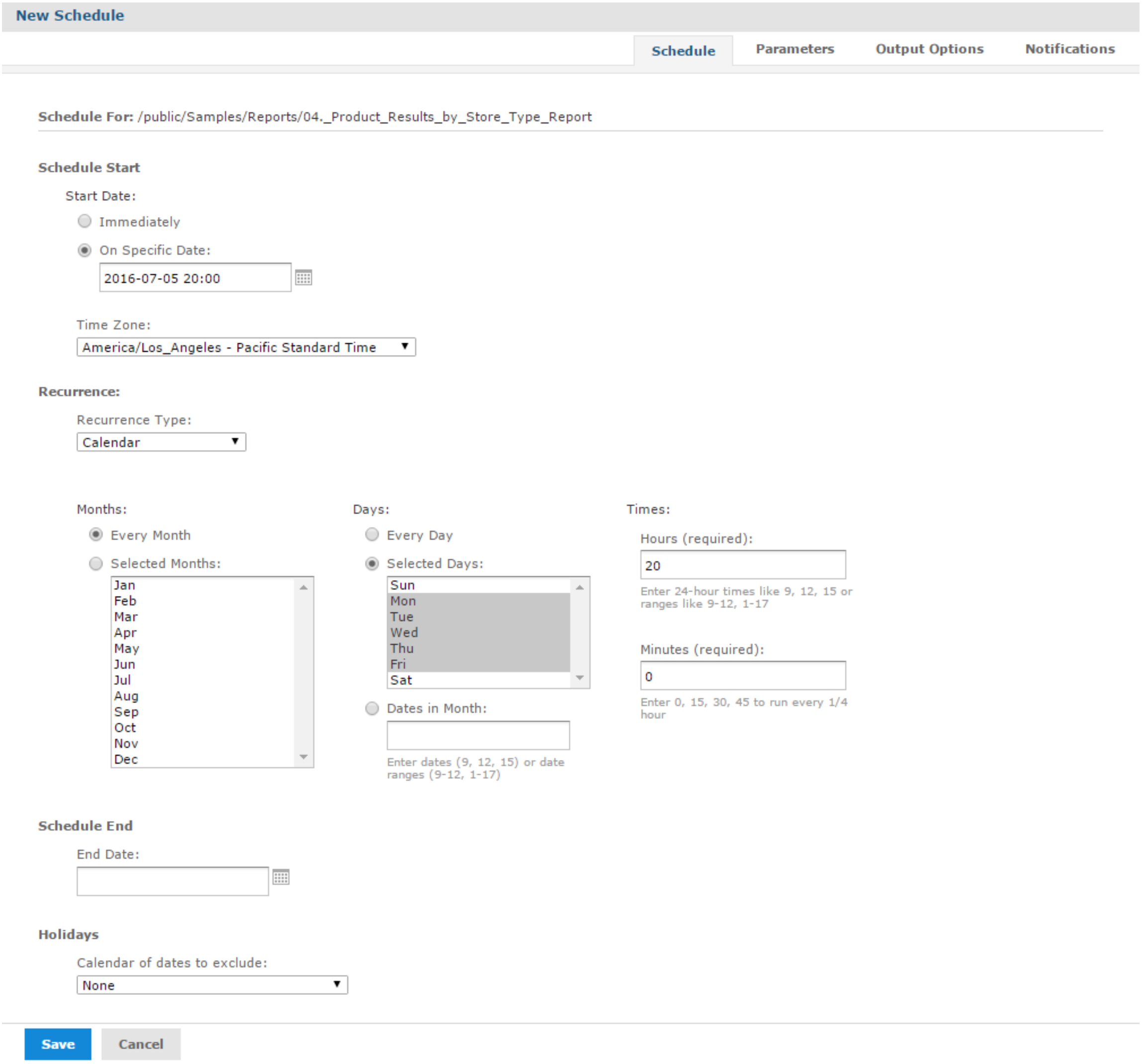
Simple recurrence options are:

- **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 4-8** you'll see an example of calendar recurrence settings.



New Schedule

Schedule | Parameters | Output Options | Notifications

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

○ 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

Save | Cancel

Figure 4-8 Calendar Recurrence Settings

Calendar recurrence options are:

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

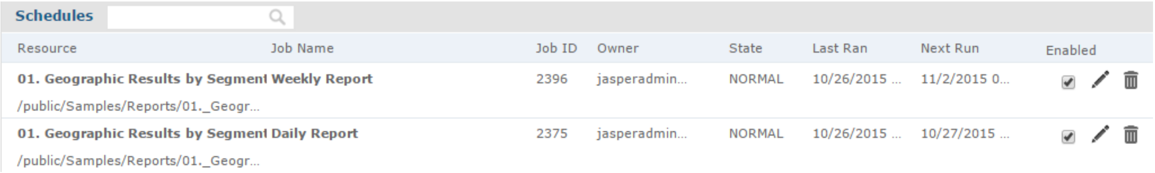
- **Days** – The days when the job runs.
  - Every Day
  - Selected Days
  - Dates in Months – Enter dates or date ranges separated by commas, for example: `1, 15`.
- **Times** – The time of day in minutes and hours when the job should run. The hours use 24-hour format. You can also enter multiple minutes 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 REST API Reference* for more information on configuring calendars.

4.3 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 4-9 The Schedules Page

Typical users see only the jobs they have defined; administrators see the jobs defined by all users. In **Figure 4-9**, 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 89**.
Refresh List	Refreshes the list of jobs, for example to see if a job has finished running.

4.4 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.

4.5 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.

4.6 Deleting a Job

To delete a scheduled job:

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

4.7 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. 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. Click the Notifications tab and set up email notifications, as described in **Setting Up Notifications**
6. Click **Save**. The job begins to run immediately.

4.8 Event Messages

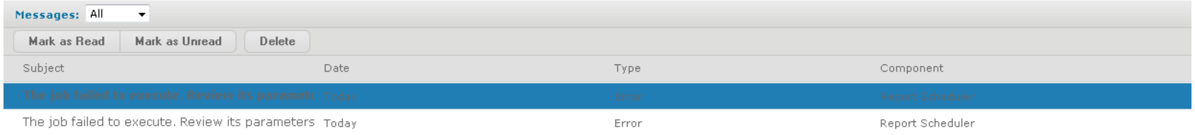
When an event occurs (for example, a scheduled report returns errors), JasperReports Server sends the owner of the job a notification message. You can browse these messages to troubleshoot report scheduling problems in the server. For example, you can determine that a job fails because its data source 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 paramete	Today	Error	Report Scheduler
The job failed to execute. Review its parameters	Today	Error	Report Scheduler

Figure 4-10 Message Management Buttons

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

CHAPTER 5 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 55**.

5.1 Overview of the Ad Hoc Editor

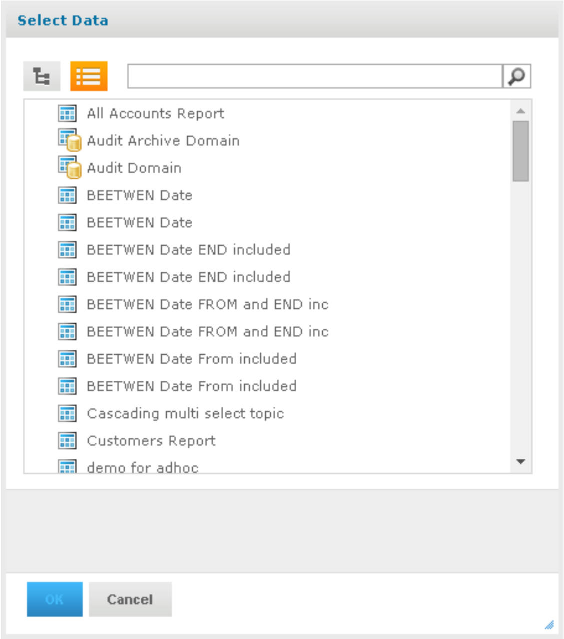
The Ad Hoc Editor is an interactive tool for creating views for various types of reports: tables, crosstabs, and charts. You create these views by simply dragging and dropping elements. You can add and summarize fields, define groups, label and title the report, and format data for each field. The effects of your changes are evident immediately, and you can adjust the display to highlight the most relevant and compelling aspects of your data.

The Ad Hoc Editor provides analysis options (such as slice, pivot, and filter) to help you recognize trends and outliers in your data. You can drill into specific details or analyze your data at a very high level.

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**.

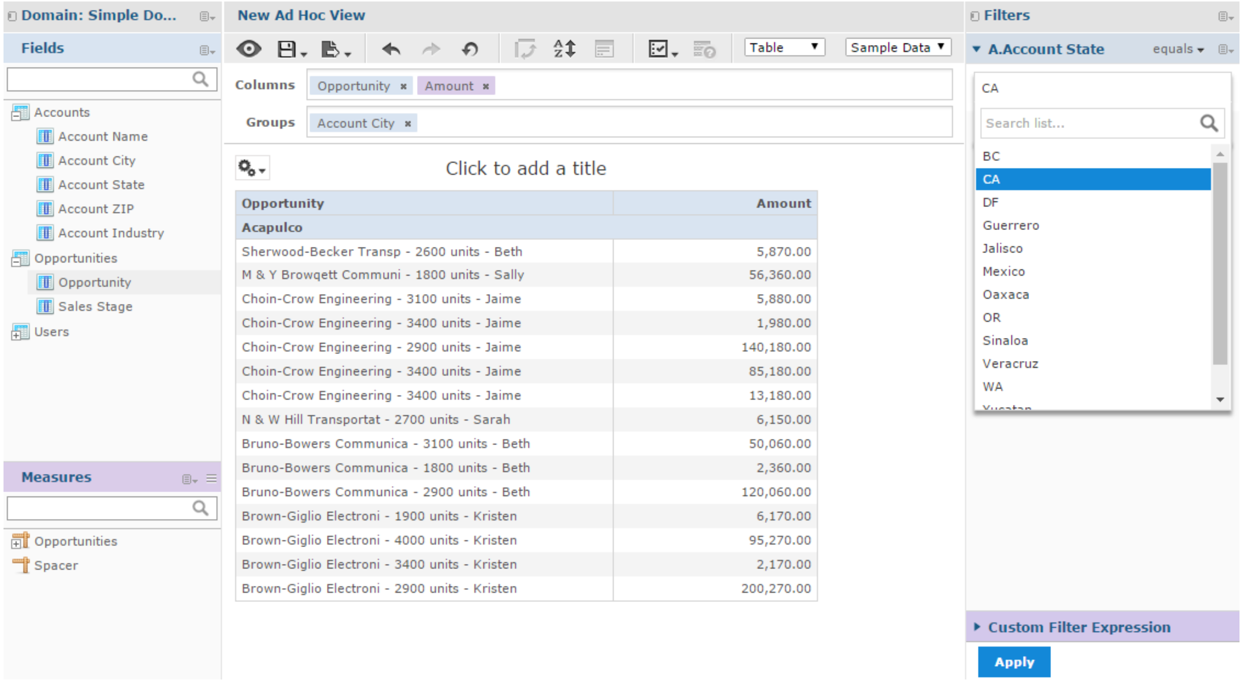


All Accounts Report
Audit Archive Domain
Audit Domain
BEETWEN Date
BEETWEN Date
BEETWEN Date END included
BEETWEN Date END included
BEETWEN Date FROM and END inc
BEETWEN Date FROM and END inc
BEETWEN Date From included
BEETWEN Date From included
Cascading multi select topic
Customers Report
demo for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

Figure 5-1 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.



Opportunity	Amount
Acapulco	
Sherwood-Becker Transp - 2600 units - Beth	5,870.00
M & Y Browqett Communi - 1800 units - Sally	56,360.00
Choin-Crow Engineering - 3100 units - Jaime	5,880.00
Choin-Crow Engineering - 3400 units - Jaime	1,980.00
Choin-Crow Engineering - 2900 units - Jaime	140,180.00
Choin-Crow Engineering - 3400 units - Jaime	85,180.00
Choin-Crow Engineering - 3400 units - Jaime	13,180.00
N & W Hill Transportat - 2700 units - Sarah	6,150.00
Bruno-Bowers Communica - 3100 units - Beth	50,060.00
Bruno-Bowers Communica - 1800 units - Beth	2,360.00
Bruno-Bowers Communica - 2900 units - Beth	120,060.00
Brown-Giglio Electroni - 1900 units - Kristen	6,170.00
Brown-Giglio Electroni - 4000 units - Kristen	95,270.00
Brown-Giglio Electroni - 3400 units - Kristen	2,170.00
Brown-Giglio Electroni - 2900 units - Kristen	200,270.00

Figure 5-2 User Interface of the Ad Hoc Editor

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

5.1.1 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.

5.1.1.1 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).

The views in the /Ad Hoc Components/Topics folder populate the Topics tab that appears when users click **Create > Ad Hoc 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 95**.

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

5.1.1.2 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.

Unlike Topics, which must be stored in a specific folder in the repository, Domains are detected regardless of their location in the repository. The /Domains folder is included for your convenience, but the Domains tab in the Source dialog displays all the Domains to which you have access in the repository.

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 95**.

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

5.1.1.3 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 128**
- **“Working with Topics” on page 165**

5.1.2 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 107**.
- For more information on crosstab views, see **“Working with Standard Crosstabs” on page 123**.

5.1.2.1 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.

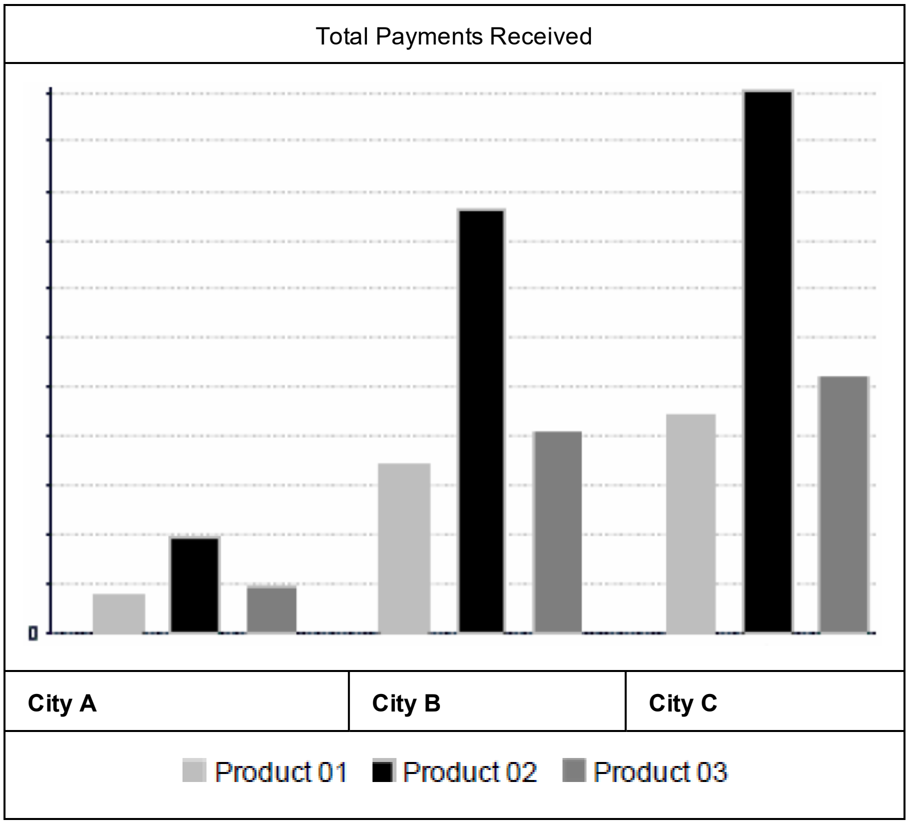
Using groups, rows can be grouped by identical values in any field with intermediate summaries for each grouped value. For example, a table view of product orders might contain columns to show the dates and amounts of each order, and its rows might be grouped by city and product.

	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 101**.

5.1.2.2 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 107**.

5.1.2.3 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 123**.

OLAP connection-based crosstabs behave differently than those created from Topics or Domains. See **"Working with Topics" on page 165**.

5.1.3 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 102**, **"Using Fields and Measures in Charts" on page 108**, and **"Using Fields in Crosstabs" on page 124**.

5.1.4 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 change 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 5-1 on page 99**.

Table 5-1 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 160**.
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 105**
☰	Input Controls	Click this icon to see the input controls applied to this view. For more information, refer to **"Using Input Controls" on page 158**.
☑	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 100** For more information viewing MDX queries, see **"Viewing the MDX Query" on page 131**

5.1.4.1 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.

5.1.4.2 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** (default) 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.

5.1.4.3 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.

5.1.5 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 153**.

5.1.6 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 58**. When you run the report, it is displayed as a JasperReport.

5.1.6.1 Dependent Reports

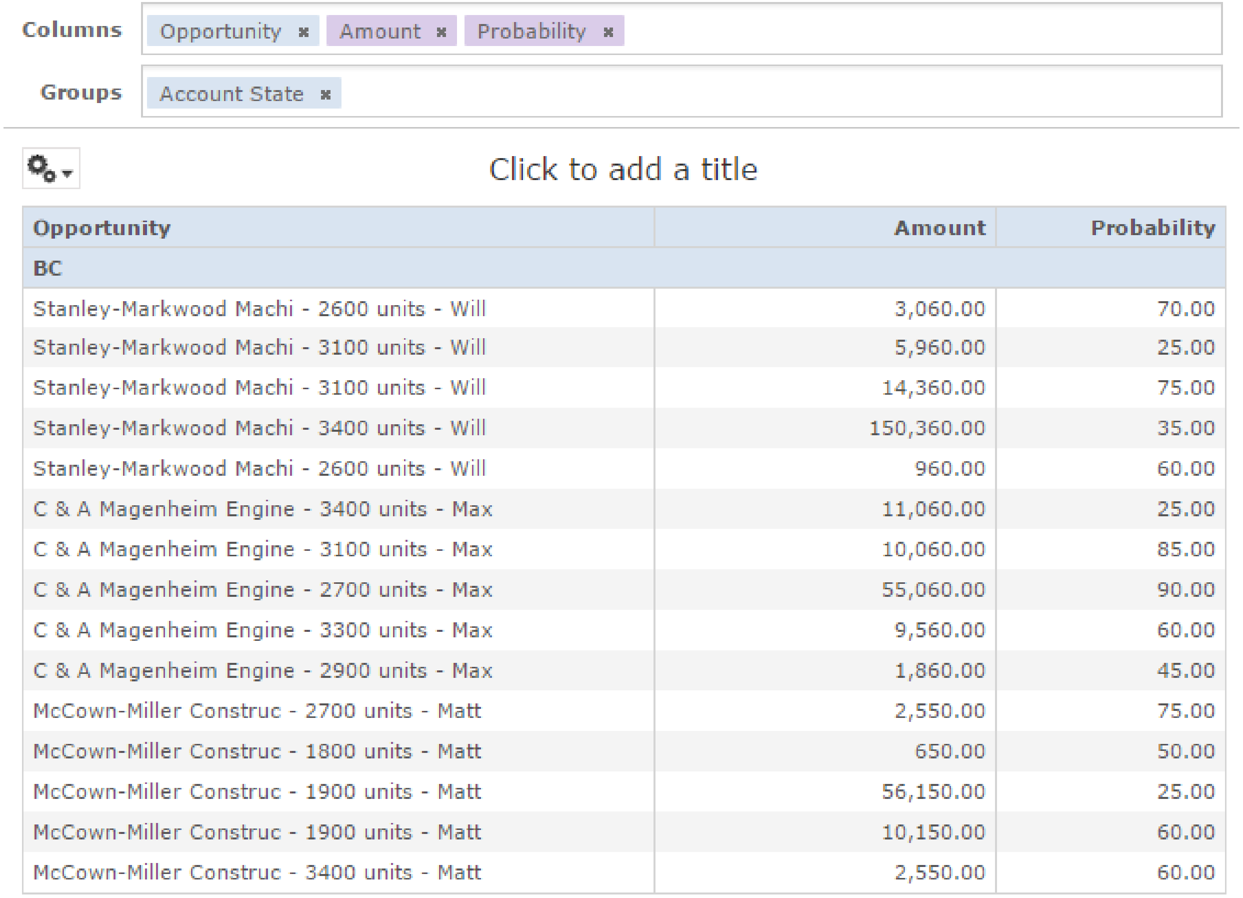
When you create a report from an Ad Hoc view, the report is considered "dependent" on that view.

When you save an Ad Hoc view, some, but not all, of its changes appear in its dependent reports. For example, if you open an Ad Hoc view with a table and adjust the data level for its columns, the column changes will show up in previous reports created from that view.

In cases where changes to an Ad Hoc view could cause errors in dependent reports, you should save the updated view with a different file name and create a new report.

5.2 Working with Tables

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



Columns: Opportunity ×, Amount ×, Probability ×

Groups: Account State ×

Click to add a title

Opportunity	Amount	Probability
BC		
Stanley-Markwood Machi - 2600 units - Will	3,060.00	70.00
Stanley-Markwood Machi - 3100 units - Will	5,960.00	25.00
Stanley-Markwood Machi - 3100 units - Will	14,360.00	75.00
Stanley-Markwood Machi - 3400 units - Will	150,360.00	35.00
Stanley-Markwood Machi - 2600 units - Will	960.00	60.00
C & A Magenheim Engine - 3400 units - Max	11,060.00	25.00
C & A Magenheim Engine - 3100 units - Max	10,060.00	85.00
C & A Magenheim Engine - 2700 units - Max	55,060.00	90.00
C & A Magenheim Engine - 3300 units - Max	9,560.00	60.00
C & A Magenheim Engine - 2900 units - Max	1,860.00	45.00
McCown-Miller Construc - 2700 units - Matt	2,550.00	75.00
McCown-Miller Construc - 1800 units - Matt	650.00	50.00
McCown-Miller Construc - 1900 units - Matt	56,150.00	25.00
McCown-Miller Construc - 1900 units - Matt	10,150.00	60.00
McCown-Miller Construc - 3400 units - Matt	2,550.00	60.00

Figure 5-3 **Ad Hoc Editor's Table View**

5.2.1 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.

5.2.1.1 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.

5.2.1.2 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**.

5.2.1.3 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**.

5.2.1.4 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 the 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.

5.2.1.5 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.

5.2.1.6 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 AZ↕ . 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.

5.2.1.7 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.

5.2.1.8 Changing the Data Format

You can change the formatting for columns containing numeric data, such as dates and monetary amounts. The format is applied to all rows as well as the group- and view-level summaries. By default, non-integer fields use the -1,234.56 data format; integers use -1234.

To change the data format for a column:

1. In the Ad Hoc view, right-click the column header.
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.

5.2.1.9 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.

5.2.1.10 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.

5.2.1.11 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.

5.3 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.

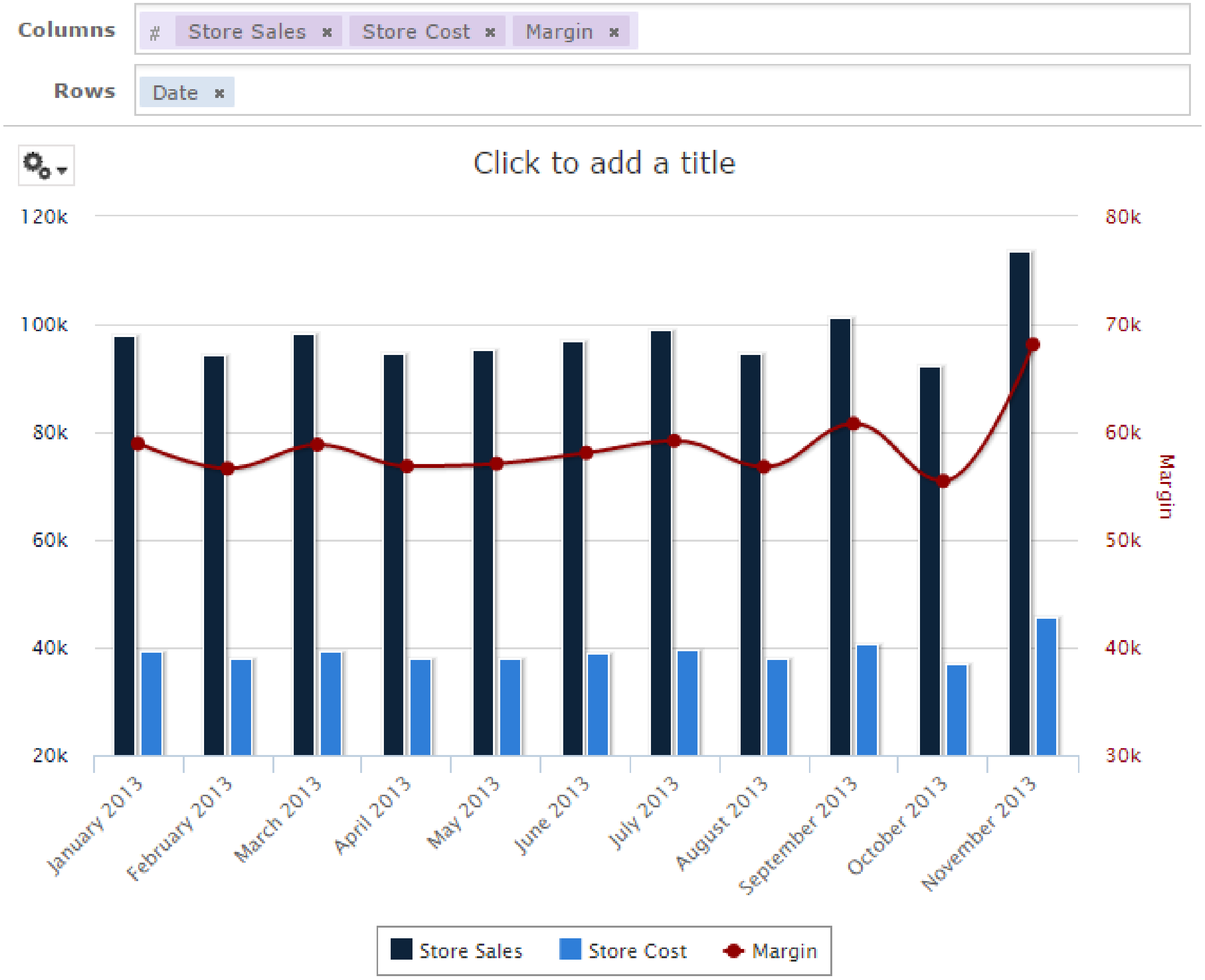


Figure 5-4 **Ad Hoc Editor's Chart View**

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 101**.

5.3.1 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 1-2** 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.

5.3.1.1 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.

Columns **Columns**

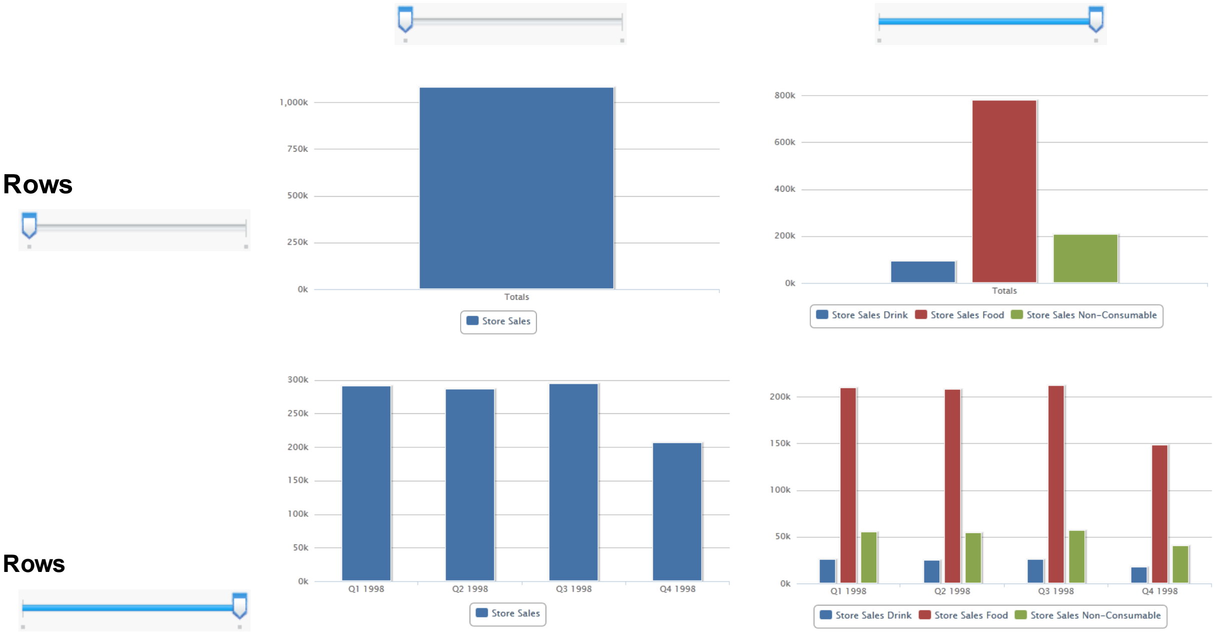


Figure 5-5 Effect of the Slider on a Chart

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 110** 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.

5.3.1.2 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.

5.3.1.3 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.

5.3.1.4 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.

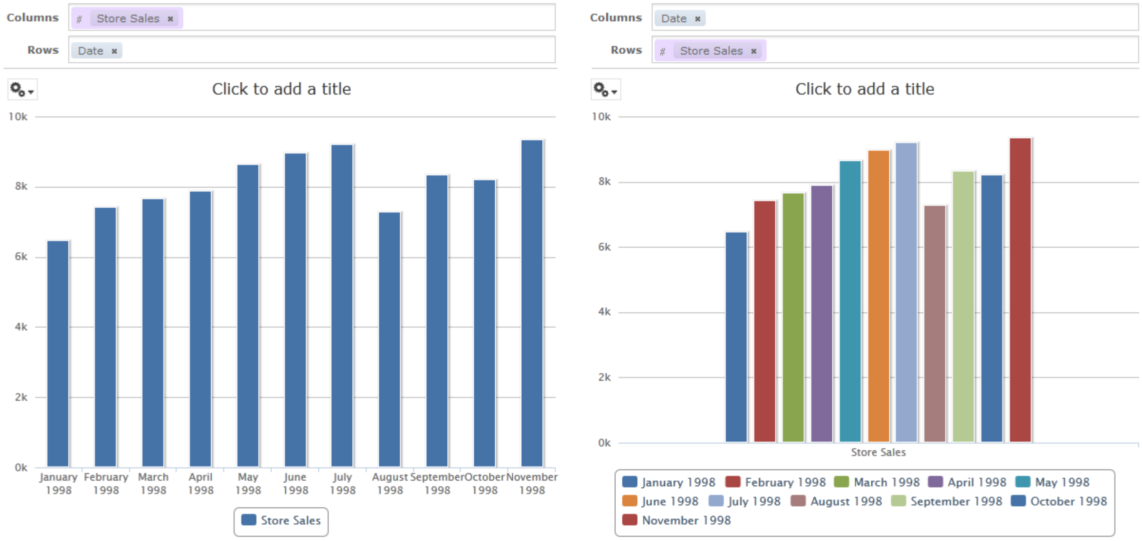


Figure 5-6 Effect of Pivoting a Chart

- 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.

5.3.2 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.

JasperReports Server has limited support for scatter charts created in some older versions. You can use the Ad Hoc views for those scatter charts to generate new reports, but you can't edit them.

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.

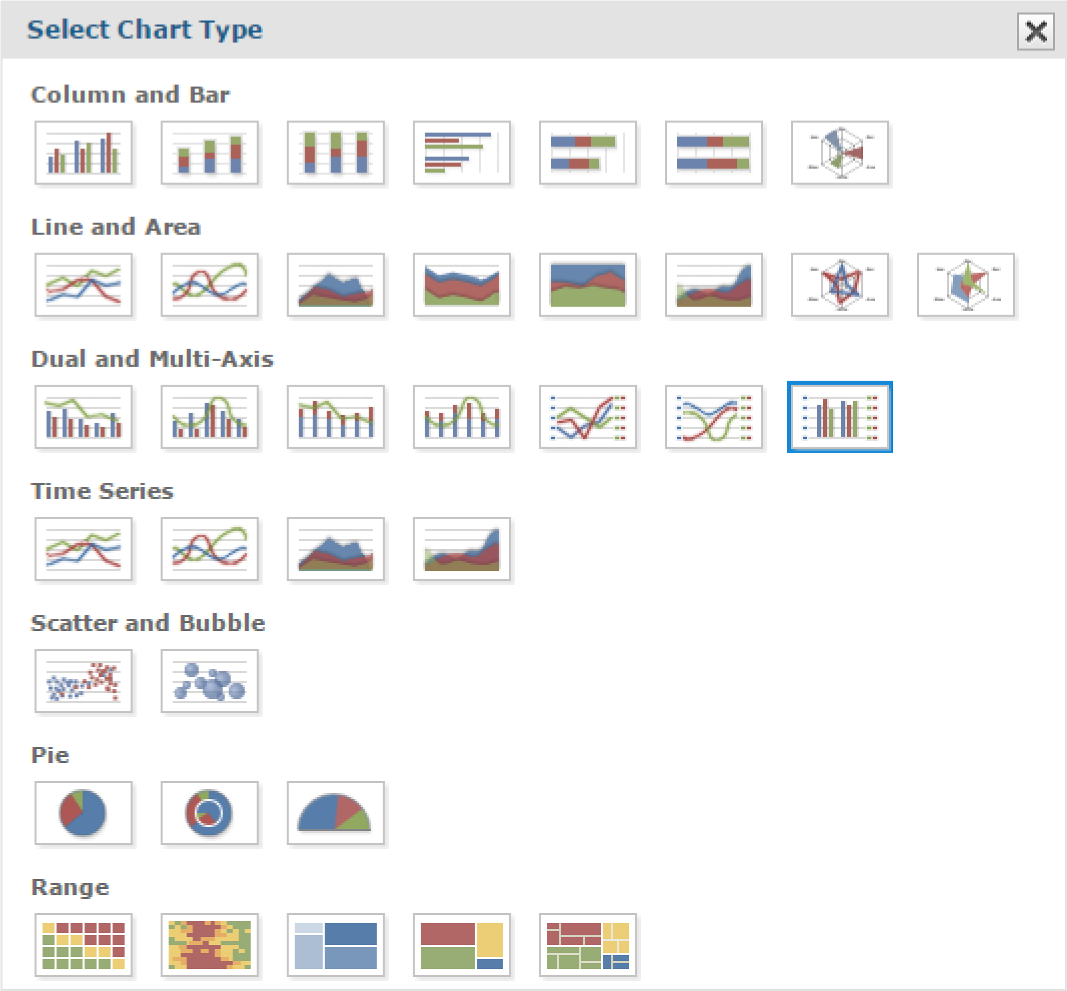


Figure 5-7 Select Chart Type Window

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 5-2 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.	

Icon	Description	Rules
	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.	
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.	

Icon	Description	Rules
	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.
	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.

Icon	Description	Rules
	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.
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.

Icon	Description	Rules
	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.	
	Semi-Pie. Multiple measures of a group are displayed as sectors of a half-circle.	

Icon	Description	Rules
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.

5.3.3 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.

5.3.3.1 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**.

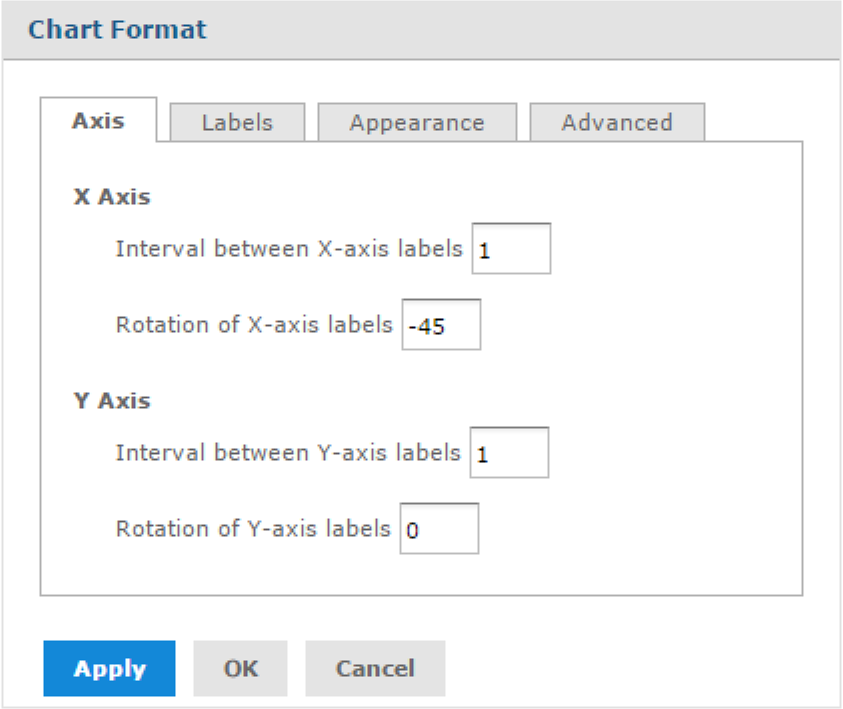


Chart Format

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 5-8 Chart Format Window

5.3.3.2 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**.

5.3.3.3 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.

5.3.3.4 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.

5.3.3.5 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.

5.3.4 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.

5.3.4.1 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:

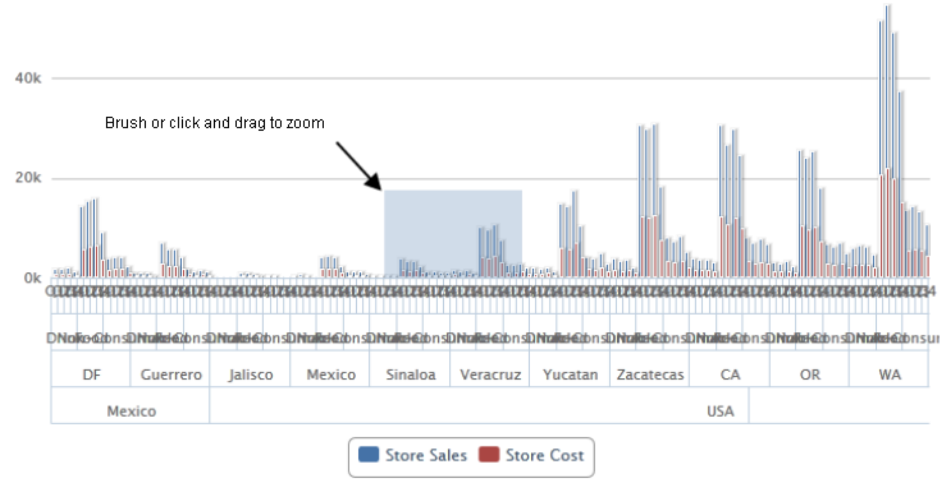


Figure 5-9 Selecting a Zoom Area

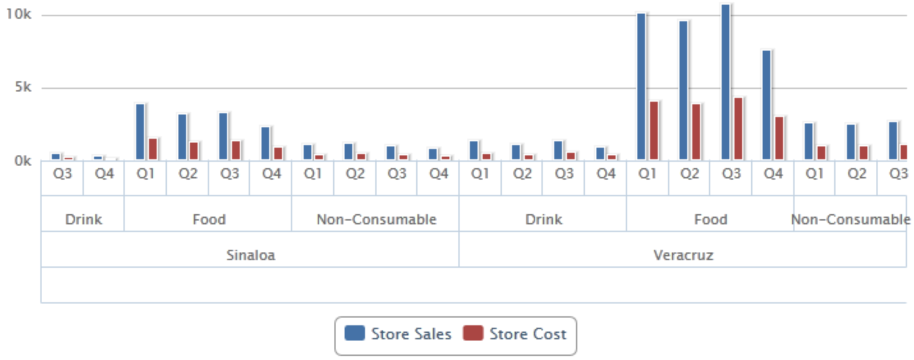


Figure 5-10 Area After Zooming

5.3.4.2 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.

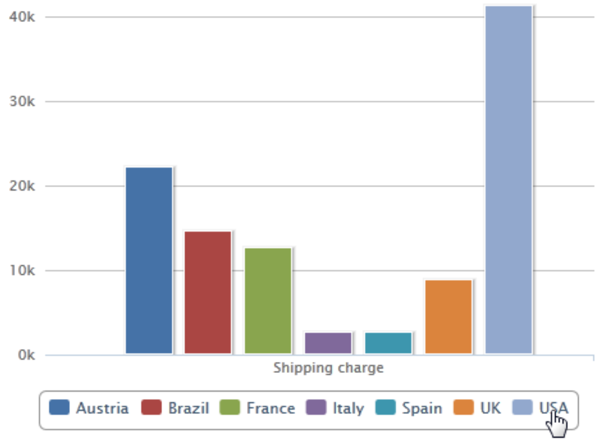


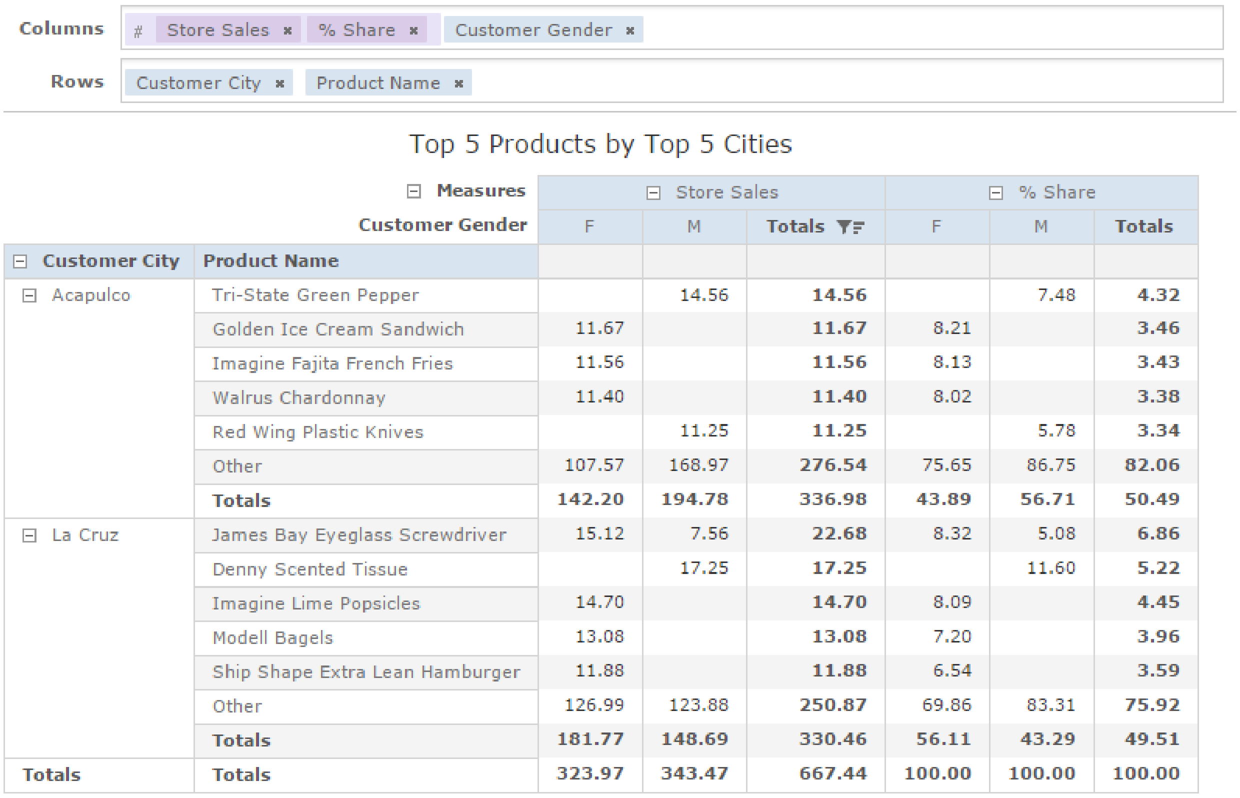
Figure 5-11 Hiding a Group Member

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.

5.4 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 128** and **"Working with Topics" on page 165**.

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



Columns # Store Sales × % Share × Customer Gender ×

Rows Customer City × Product Name ×

Top 5 Products by Top 5 Cities

	⊟ Measures	⊟ Store Sales			⊟ % Share		
	Customer Gender	F	M	Totals	F	M	Totals
⊟ Customer City	Product Name						
⊟ Acapulco	Tri-State Green Pepper		14.56	**14.56**		7.48	**4.32**
	Golden Ice Cream Sandwich	11.67		**11.67**	8.21		**3.46**
	Imagine Fajita French Fries	11.56		**11.56**	8.13		**3.43**
	Walrus Chardonnay	11.40		**11.40**	8.02		**3.38**
	Red Wing Plastic Knives		11.25	**11.25**		5.78	**3.34**
	Other	107.57	168.97	**276.54**	75.65	86.75	**82.06**
	Totals	**142.20**	**194.78**	**336.98**	**43.89**	**56.71**	**50.49**
⊟ La Cruz	James Bay Eyeglass Screwdriver	15.12	7.56	**22.68**	8.32	5.08	**6.86**
	Denny Scented Tissue		17.25	**17.25**		11.60	**5.22**
	Imagine Lime Popsicles	14.70		**14.70**	8.09		**4.45**
	Modell Bagels	13.08		**13.08**	7.20		**3.96**
	Ship Shape Extra Lean Hamburger	11.88		**11.88**	6.54		**3.59**
	Other	126.99	123.88	**250.87**	69.86	83.31	**75.92**
	Totals	**181.77**	**148.69**	**330.46**	**56.11**	**43.29**	**49.51**
Totals	**Totals**	**323.97**	**343.47**	**667.44**	**100.00**	**100.00**	**100.00**

Figure 5-12 **Ad Hoc Editor's Standard Crosstab View**

If you selected Crosstab when you created a view, as described in **"Ad Hoc View Types" on page 95**, the following sections explain tasks specific to your crosstab development.

5.4.1 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.

5.4.1.1 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.

5.4.1.2 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 Calculation**
- **Change Time Balance Calculation**
- **Change Data Format**
- **Remove From Crosstab**
- **Create Filter**
- **Move Up** or **Move Down** or **Move Left** or **Move Right**

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.

5.4.1.3 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 160**.
- 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.

5.4.1.4 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 153**.

5.4.1.5 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.

5.4.1.6 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.

5.4.1.7 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**.

5.4.1.8 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.

5.4.1.9 Changing How Totals Are Calculated Using Time Balance

By default, crosstabs show the Totals of numeric values as the sum of the values in a column. For example, if the crosstab view displays product sales grouped by month, the crosstab displays the total sum sold during those months as the Totals value.

You can change the data that the crosstab displays as the Totals value for a period of time using the Time Balance options. These options allow you to change the Totals to display the first and last numeric values for the period. For example, if you want to see the number of products left in your inventory at the end of a certain time period, use the Time Balance Last option and the Totals fields will display the last numeric value entered for the time period. If you want to see the number of products in your inventory at the beginning of the time period, use the Time Balance First option.

Changing the time balance calculation requires the crosstab to use the Date field. You cannot drill through the data when using a Time Balance option other than the default or when using Day of Week for the grouping.

To change the totals calculation using time balance:

1. In the Ad Hoc view, right-click the column or row header.
2. Select **Change Time Balance** from the context menu.
3. Select the option you want to use. You can select one of the following:
   - Time Balance Default. This option displays the sum of all the numeric data for a period of time as the Total value.
   - Time Balance First. This option displays the first numeric value for a period of time as the Total value. For example, if the period of time is a month, the crosstab will display the first numeric value entered for the month.
   - Time Balance Last. The option displays the last numeric value for a period of time as the Total value. For example, if the period of time is a month, the crosstab will display the last numeric value entered for the month.

   The crosstab is updated to show the new Total values.

5.4.1.10 Changing the Data Format

You can change the formatting for rows and columns containing numeric data, such as dates and monetary amounts. By default, non-integer fields use the -1,234.56 data format; integers use -1234.

To change the data format for a column:

1. In the Ad Hoc view, right-click the column or row header.
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 or row now appears in the new format.

5.4.1.11 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.

5.4.1.12 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:

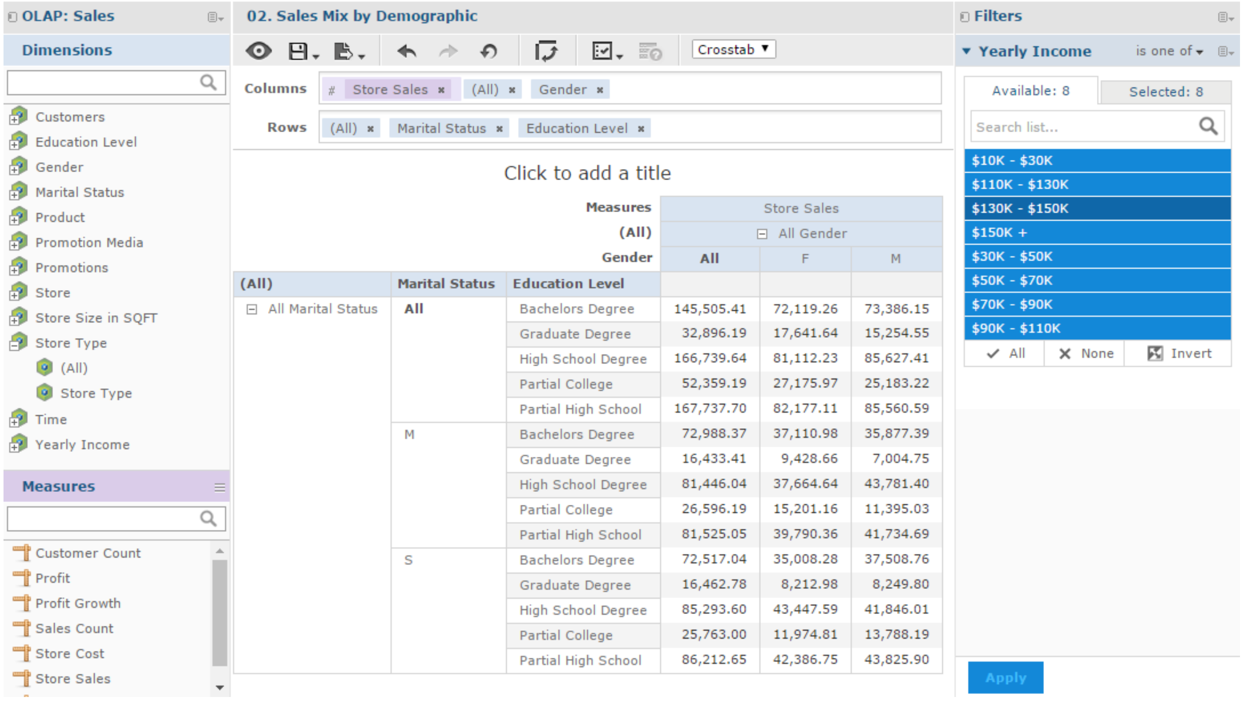
- Click hyperlinked data to display additional columns from that specific fact data.

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

5.5 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 5-1**.



Columns: Store Sales, (All), Gender

Rows: (All), Marital Status, Education Level

Click to add a title

		Measures	Store Sales		
		(All)	All Gender		
		Gender	All	F	M
(All)	Marital Status	Education Level			
All Marital Status	All	Bachelors Degree	145,505.41	72,119.26	73,386.15
		Graduate Degree	32,896.19	17,641.64	15,254.55
		High School Degree	166,739.64	81,112.23	85,627.41
		Partial College	52,359.19	27,175.97	25,183.22
		Partial High School	167,737.70	82,177.11	85,560.59
	M	Bachelors Degree	72,988.37	37,110.98	35,877.39
		Graduate Degree	16,433.41	9,428.66	7,004.75
		High School Degree	81,446.04	37,664.64	43,781.40
		Partial College	26,596.19	15,201.16	11,395.03
		Partial High School	81,525.05	39,790.36	41,734.69
	S	Bachelors Degree	72,517.04	35,008.28	37,508.76
		Graduate Degree	16,462.78	8,212.98	8,249.80
		High School Degree	85,293.60	43,447.59	41,846.01
		Partial College	25,763.00	11,974.81	13,788.19
		Partial High School	86,212.65	42,386.75	43,825.90

Yearly Income is one of
- $10K - $30K
- $110K - $130K
- $130K - $150K
- $150K +
- $30K - $50K
- $50K - $70K
- $70K - $90K
- $90K - $110K

Figure 5-13 **Ad Hoc Editor’s OLAP Connection-based Crosstab View**

5.5.1 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.

5.5.2 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.

5.5.3 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.

5.5.4 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.

5.5.5 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.

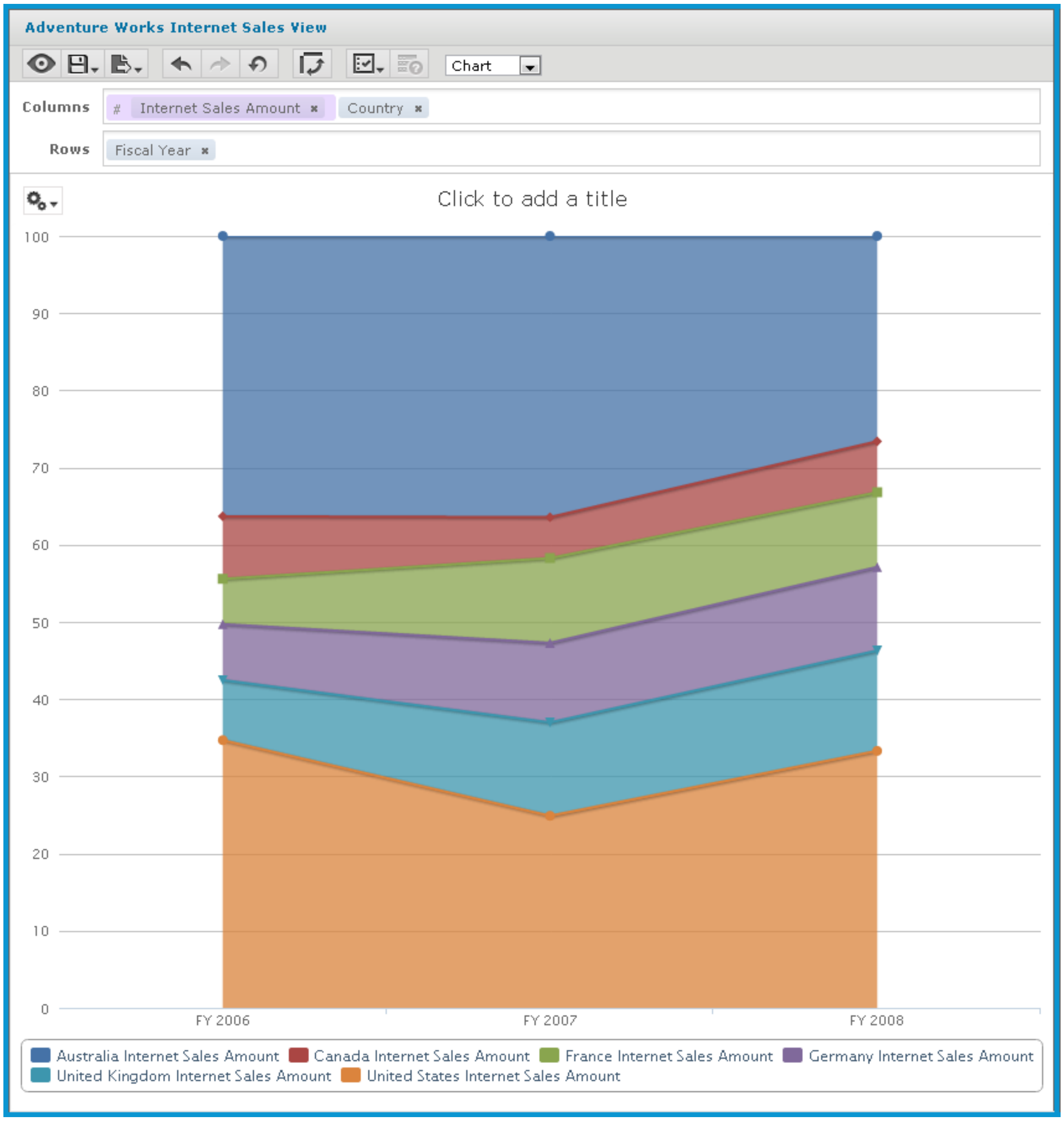

Adventure Works Internet Sales View
Chart
Columns
# Internet Sales Amount ×
Country ×
Rows
Fiscal Year ×
Click to add a title
100
90
80
70
60
50
40
30
20
10
0
FY 2006
FY 2007
FY 2008
Australia Internet Sales Amount
Canada Internet Sales Amount
France Internet Sales Amount
Germany Internet Sales Amount
United Kingdom Internet Sales Amount
United States Internet Sales Amount

Figure 5-14 Search Results Listing

For details, refer to the *Jaspersoft OLAP User Guide*.

5.6 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 fx) or calculated measures (shown by the icon 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.

5.6.1 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 fx) or calculated measure (shown by the icon fx) 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.

5.6.1.1 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.

5.6.1.2 The Summary Tab

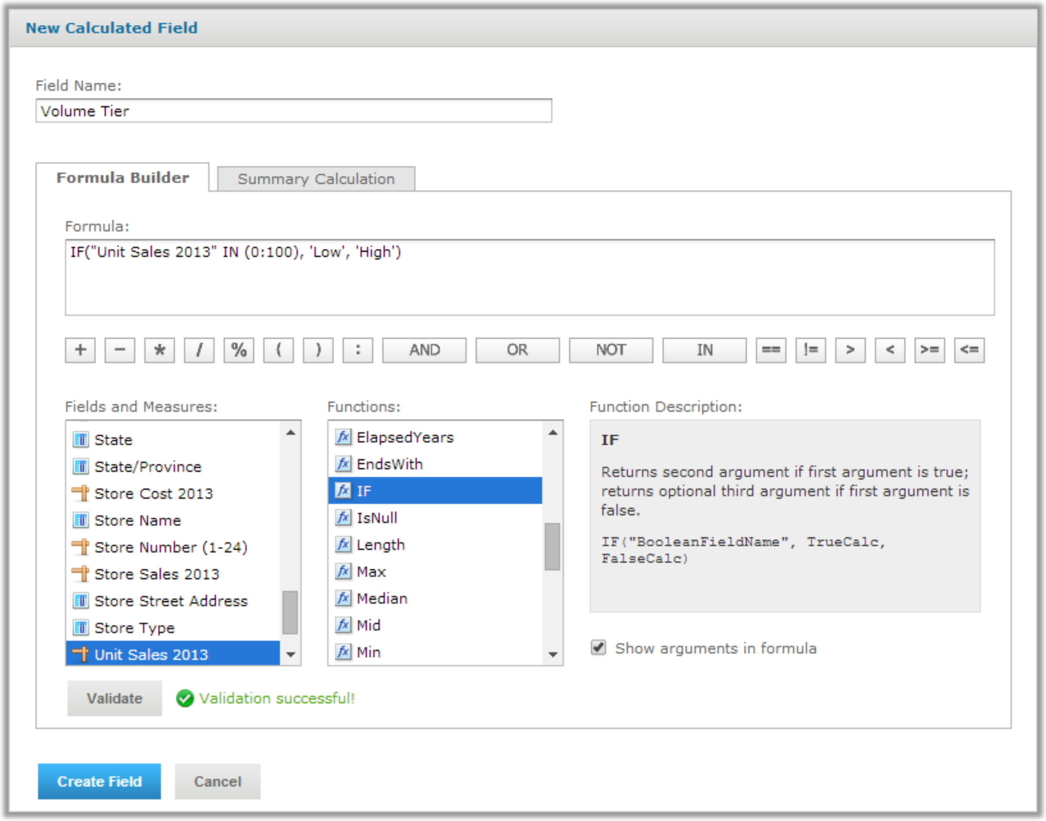
Summaries show a result applied to all data values. For example, for a numeric fields such as Cost, the summary value might be the sum of all the costs; for a text field such as Customer Name, the summary value might be the count of all customers. The Summary tab lets you set the default summary function 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.

5.6.2 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.



New Calculated Field

Field Name: Volume Tier

Formula: IF("Unit Sales 2013" IN (0:100), 'Low', 'High')

Function Description:

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 5-15 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)`.

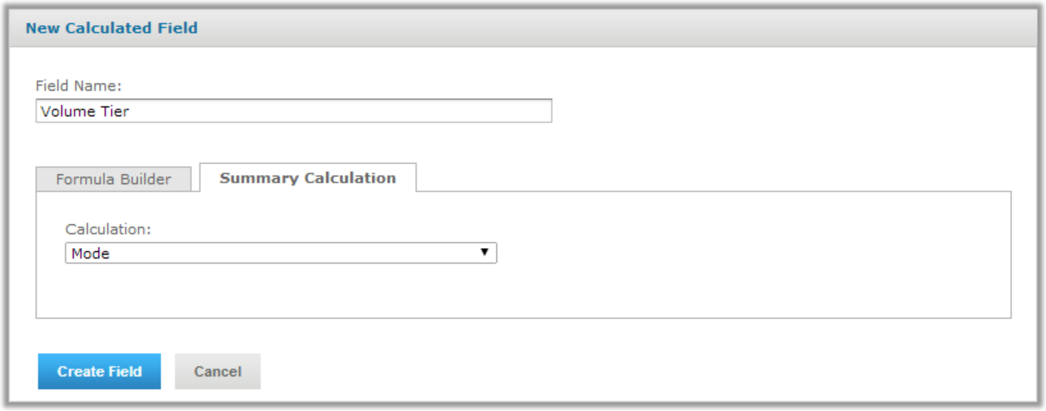
10. Edit the expression in the Formula Builder to read as follows: `IF("Unit Sales 2013" IN (0:100), 'Low', 'High')`.

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 5-16 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.

5.6.3 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.
  - If the calculated field uses an item that was removed from the data source, the Ad Hoc Editor displays an error message. You will need to either edit or delete the calculated field.

5.6.4 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.

Example:

```
Attribute('my_attribute')
```

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')
```

```
Boolean(Attribute('my_boolean_attribute'))
```

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')
```

```
Date(Attribute('my__date_attribute'))
```

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")
```

ElapsedMinutes (*DateTimeExpression1*,*DateTimeExpression2*)

Calculates the number of minutes elapsed between two date fields that contain time values.

Example:

```
ElapsedMinutes ("Date shipped","Date required")
```

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")
```

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")
```

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')`

`Integer(Attribute('my_integer_attribute'))`

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')
```

```
Time(Attribute('my_time_attribute'))
```

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')
```

```
Timestamp(Attribute('my_timestamp_attribute'))
```

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" )`

5.6.5 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.

5.6.6 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.

5.6.7 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.

5.6.7.1 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.

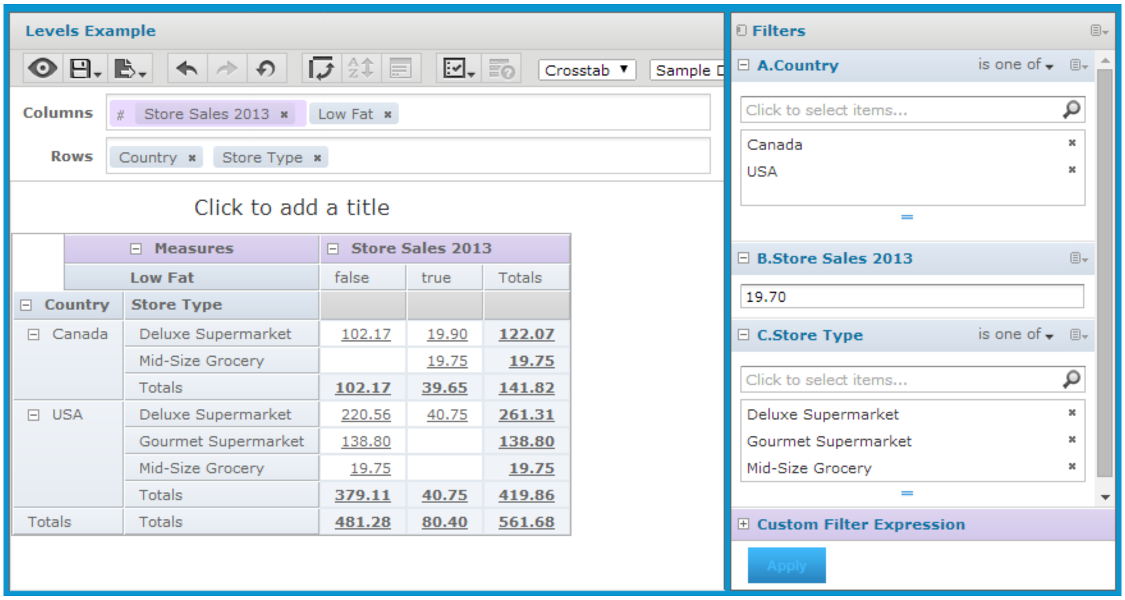
The following example shows how RowGroup works with the PercentOf() function.

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.



Levels Example

Columns: Store Sales 2013, Low Fat

Rows: Country, Store Type

Click to add a title

	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

Filters

A.Country is one of: Canada, USA

B.Store Sales 2013: 19.70

C.Store Type is one of: Deluxe Supermarket, Gourmet Supermarket, Mid-Size Grocery

Custom Filter Expression

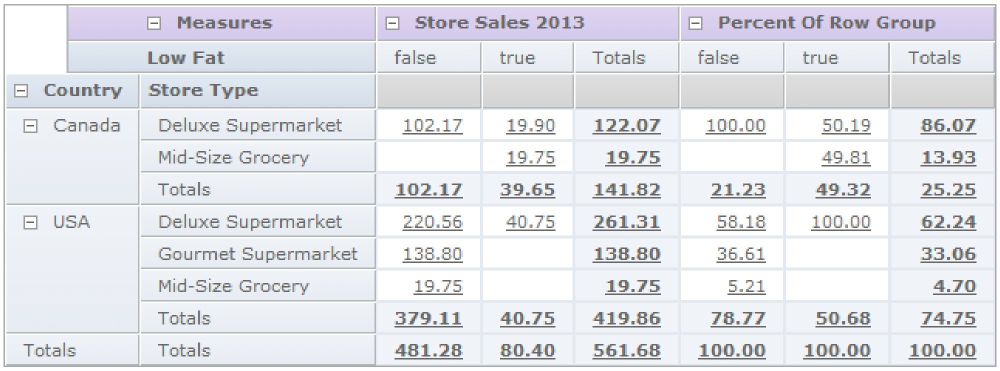
Figure 5-17 Base Example for Levels

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 5-18 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%.

5.6.8 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 5-3 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

Function	Meaning	Available for....
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
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 Calculation** 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 Calculation** 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.

5.7 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 159**. 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 155** 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 186**.
- Domain Topic filters, see **"Creating Topics from Domains" on page 167**.
- Localizing input control prompts and lists of values, see **"Localizing Reports" on page 200**.

5.7.1 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 229**. For information on defining filters in the Data Chooser, see **"The Pre-filters Page" on page 163**.

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 155**.

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.

5.7.1.1 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. An option used by itself (without `+/-<Number>`) gives the current value for that option.
- **+** 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 154**, 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.

5.7.1.2 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 154**. 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.

5.7.2 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.

5.7.3 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 186**.

> 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 163**.
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**.

5.8 Creating a View from a Domain

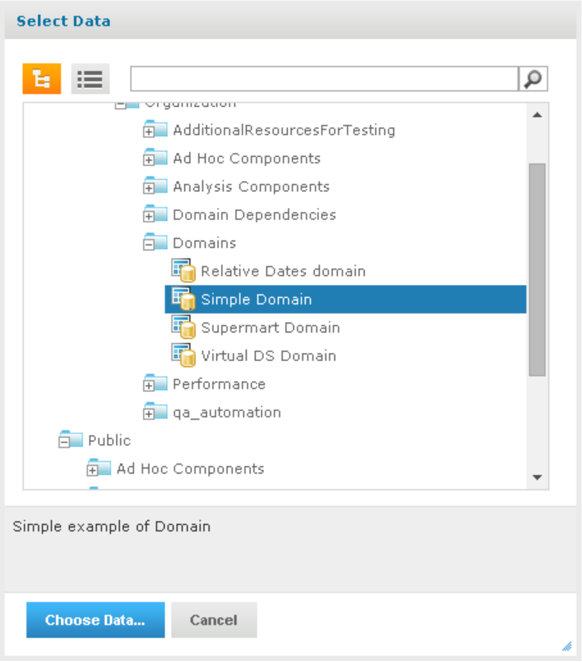
Like Topics and OLAP connections, administrators and data analysts create Domains for reuse to simplify access to data during view design. Domains give view makers more flexibility than Topics in choosing fields from the database and allow filtering of the data before it is included in a view and the 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 **"Working with the Domain Designer" on page 205**.

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.



Simple example of Domain

Figure 5-19 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.

5.8.1 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, removing the items from the Domain will result in a Missing Data error message when opening the Ad Hoc view in the editor. The user is prompted to remove the items from the Ad Hoc view. The removed items no longer appear in the Data Source Selection panel when the editor opens.
- 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.

5.8.2 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:

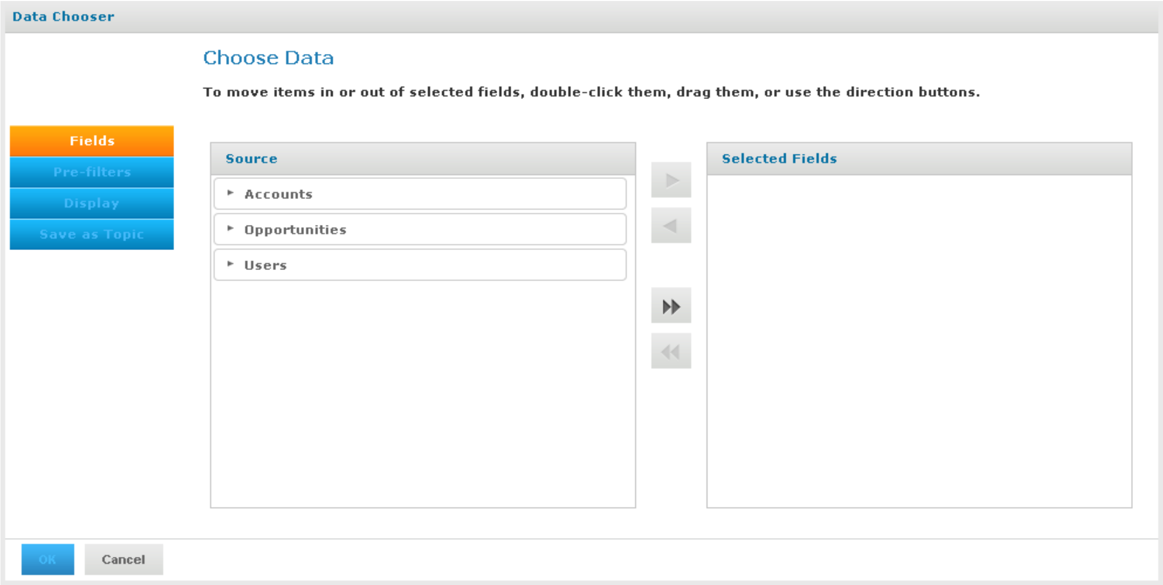
1. Click **Create** > **Ad Hoc View**.
2. Click and browse to a Domain, then click **Choose Data** to access the following pages of the Data Chooser:
   - **The Select Fields Page** – Choose the fields to make available in the Ad Hoc Editor.

- **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.

5.8.2.1 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:



To move items in or out of selected fields, double-click them, drag them, or use the direction buttons.

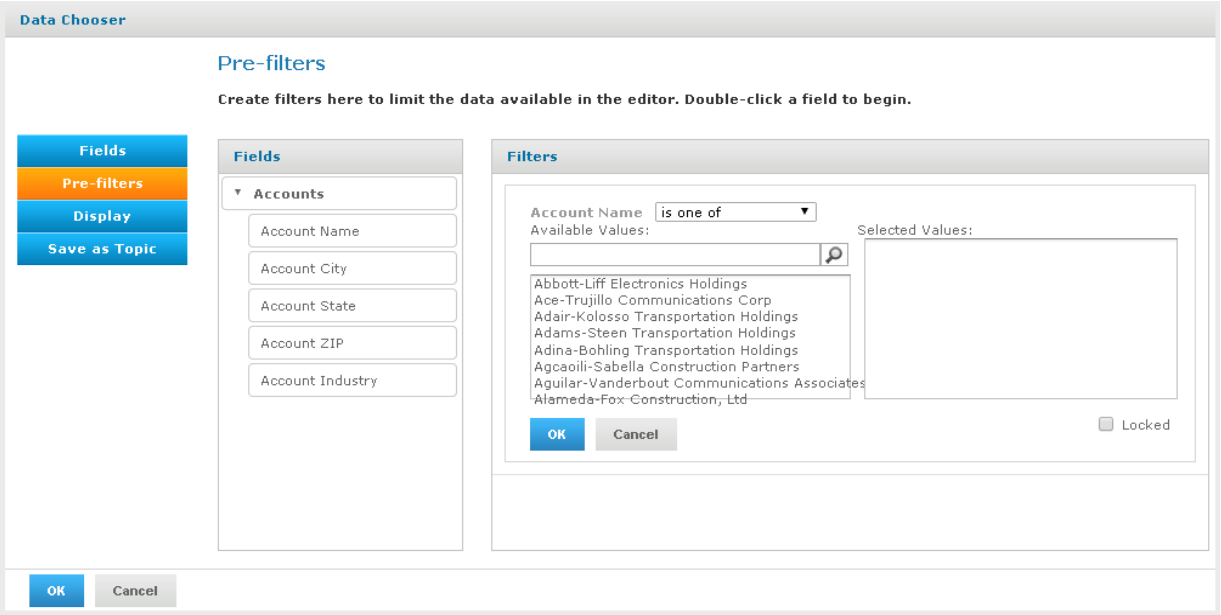
Figure 5-20 The Fields Page of the Data Chooser

- 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.

5.8.2.2 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 5-21 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 158**.

   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.

5.8.2.3 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.

5.8.2.4 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. 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.

5.9 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.

5.9.1 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 196**. 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**.

The JRXML file that the Topic is based on must contain a query and a field list.

Any report layout in the Topic's JRXML file is ignored by the Ad Hoc Editor. We recommend that a Topic's JRXML file not include anything other that the query and field list.

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 200**.

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*.

5.9.2 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 168**. 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.

Ad Hoc views based on Domain topics may return duplicate data if Data Staging is enabled.

5.9.2.1 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 the *JasperReports Server Security Guide* for a technical explanation of data security for Domains.

5.9.2.2 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:

5.9.2.3 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 161** 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 6 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. See the *Jaspersoft Studio User Guide* for information.

To add the sample report units to the server, you need access to the sample data in the server installation directory on the file system (<js-install>/samples). Contact your administrator for help in locating these files.

In most cases, it is preferable to upload the JRXML file from Jaspersoft Studio. Uploading through JasperReports Server is included for completeness.

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:

6.1 Overview of a Report Unit

In the server, a report unit is the collection of elements used for retrieving data and formatting output. **Figure 6-1 on page 172** 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*.

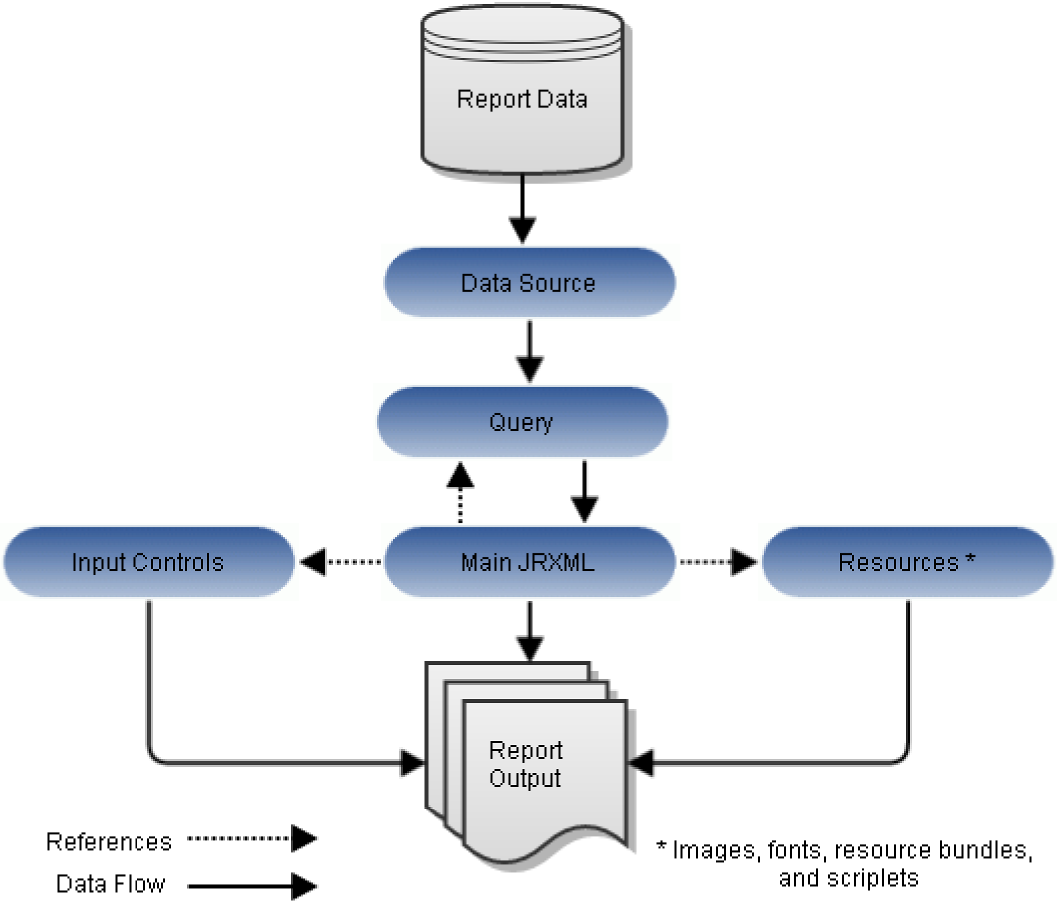


Figure 6-1 Anatomy of a Report Unit

6.2 Adding a Report Unit to the Server

This section presents an example of uploading a JRXML with resources to the server. This report has two images that need to be added.

6.2.1 Selecting the Main JRXML File

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 select the main JRXML file that references all other elements.

To upload the main JRXML for this 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.

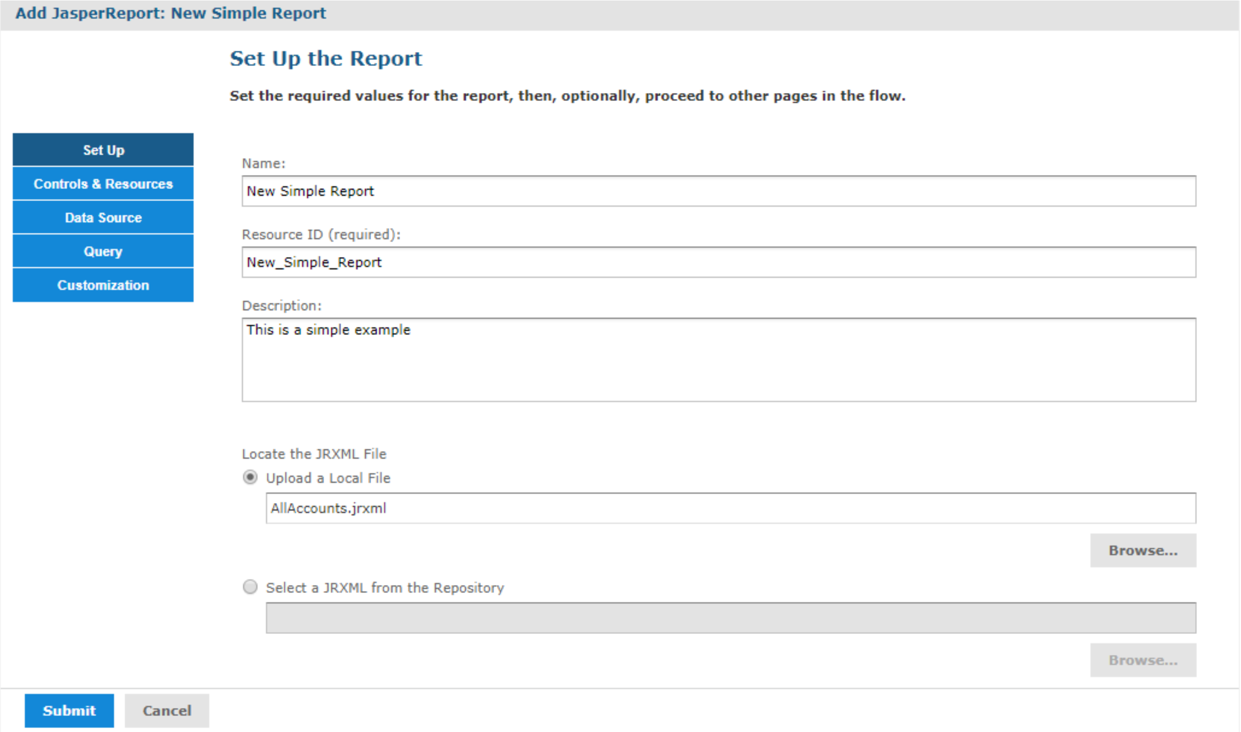
2. Locate the folder where you want to add the report. For example, go to **Public > Samples > Reports**.
3. Right-click the Reports folder and select **Add Resource > JasperReport** from the context menu. The Set Up the Report page of the JasperReport wizard appears.

   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.

   This example shows how to upload a JRXML file from the samples folder in the installation directory. You can also select a JRXML from the repository.

   In **Figure 6-2** you can see the Set Up the Report page.



Add JasperReport: New Simple Report

Set Up the Report

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

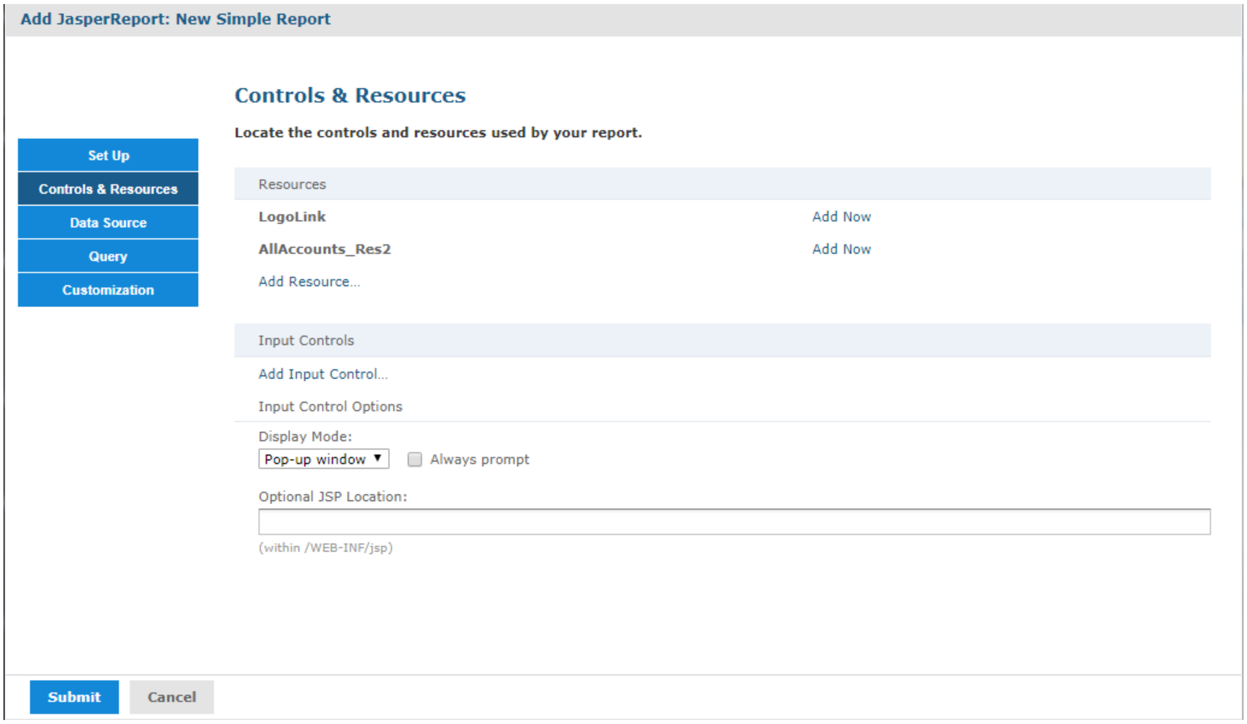
AllAccounts.jrxml

○ Select a JRXML from the Repository

Figure 6-2 Required Set Up Values

6. Click **Controls & Resources**.

   The Controls & Resources page appears. If the server detects that the report needs additional resources, you'll be prompted to locate those resources.



Add JasperReport: New Simple Report

Controls & Resources

Locate the controls and resources used by your report.

Resources

LogoLink — Add Now

AllAccounts_Res2 — Add Now

Add Resource...

Input Controls

Add Input Control...

Input Control Options

Display Mode:

Pop-up window ▼ □ Always prompt

Optional JSP Location:

(within /WEB-INF/jsp)

Figure 6-3 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 6-3, "Suggested Resources in the Resources List," on page 175** shows that two image files are missing:

- LogoLink
- AllAccounts_Res2

6.2.2 Uploading Suggested File Resources

If needed files are missing, as shown in **Figure 6-3**, you need to take one of the following actions:

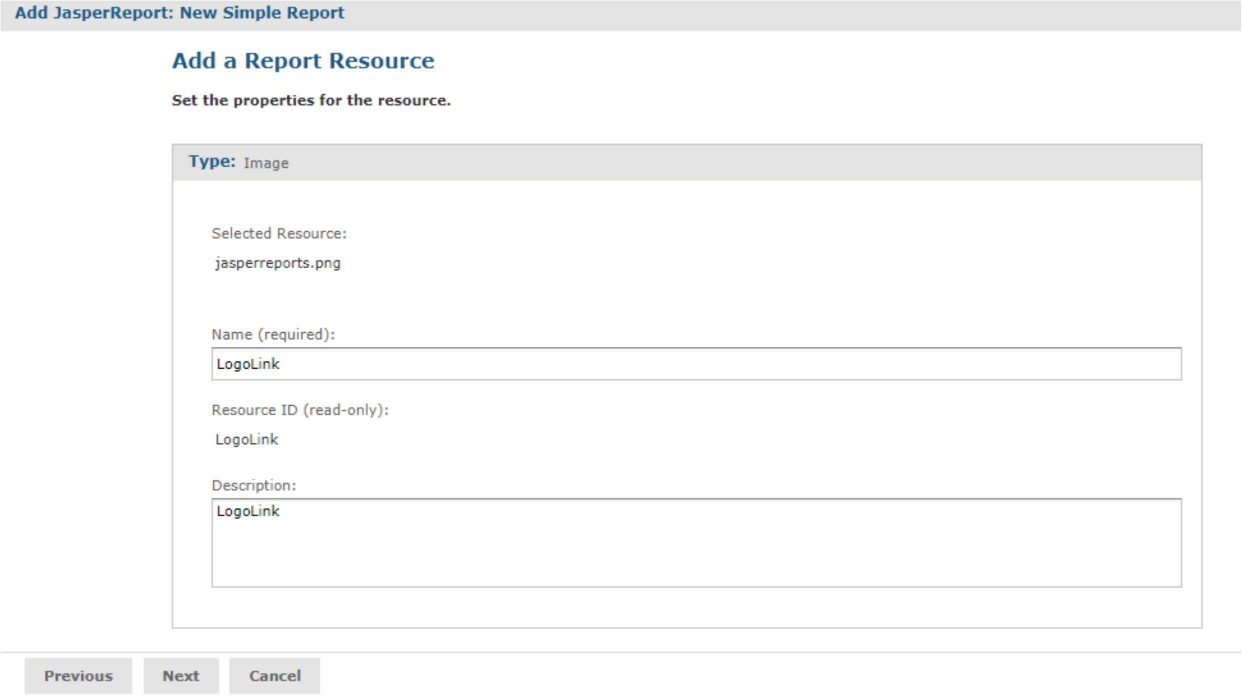
- Upload resources that the report needs
- Select a resource from the repository

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 183**.

To upload a resource from the file system:

1. On the Controls & Resources page, click **Add Now** in the row of the missing resource, for example, LogoLink. The Locate File Resource page appears.
2. Choose **Upload a Local File** then click **Choose File**.
3. **Browse** to <js-install>/samples/images. The LogoLink file is not available, but you can use an alternate image, such as <js-install>/samples/images/jasperreports.png.
4. Click **Open** to return to the Locate File Resource dialog.

5. 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: Image

Selected Resource:
jasperreports.png

Name (required):
LogoLink

Resource ID (read-only):
LogoLink

Description:
LogoLink

Figure 6-4 Properties of a Resource

The properties include the LogoLink name, resource ID, and description.

6. Click **Next** to accept the default naming of the file resource.

   The Controls & Resources page appears again, showing that the LogoLink resource was added.

To add a resource from the repository:

1. To add the second resource, click **Add Now** in the row for AllAccounts_Res2.
2. Choose **Select a resource from the Repository** and **Browse** to an image file, for example, **Public > Samples > Images > Jaspersoft_logo.png**.

   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 displays the properties of the image as set in the uploaded JRXML file.

   The properties include the LogoLink name, resource ID, and description. These properties don't redefine the properties of the Jaspersoft_logo.png file in the repository.

5. 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.

6. Click **Data Source** and define a data source as described in the next section.

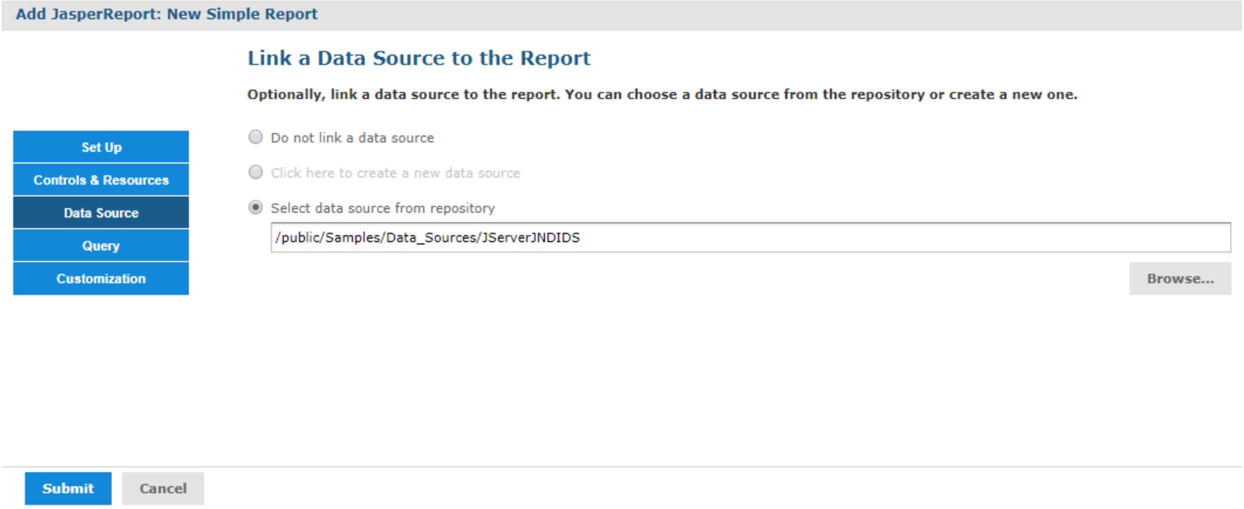
6.2.3 Defining the Data Source

Data sources are not defined directly in the report JRXML file and must be specified when you upload a file to JasperReports Server. On the Data Source page of the Add JasperReport wizard, you can select a data source in the repository or create a new data source on-the-fly.

If you want to create a data source, the application server must be able to find the driver for the database you want to use. 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 a 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 **Public > Samples > Data sources > JServer JNDI Data Source**.
3. Click **Select**. The Link a Data Source to the Report page reappears with the path to the data source.



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

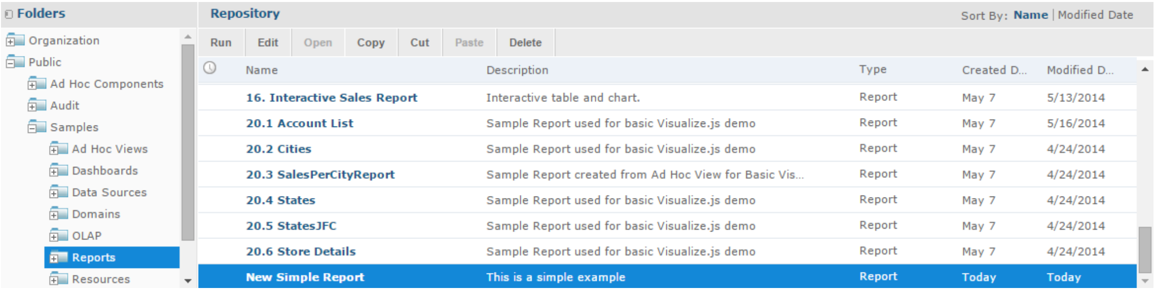
/public/Samples/Data_Sources/JServerJNDIDS

Figure 6-5 Data Source Page

4. Click **Submit** to add the new report unit to the repository.

6.2.4 Saving the New Report Unit

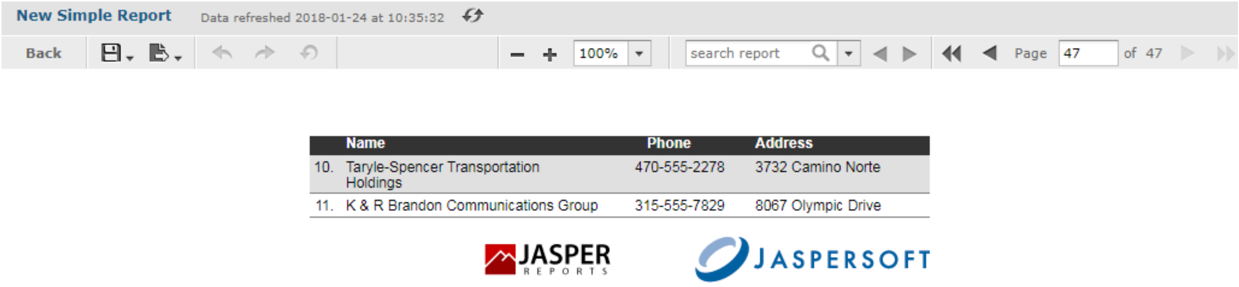
To submit a new report unit to the repository, click **Submit** on any page of the JasperReport wizard, from the Set Up page to the Customization page. You don't have to set options you don't need, such as Customization. When you click **Submit**, the server attempts to upload the report JRXML and its resources. If the upload 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 6-6 New Report Added to the Repository

In the report viewer, click ⏩ to go to the end of the report, where the logo images that you added appear. The following figure shows the output.



	Name	Phone	Address
10.	Taryle-Spencer Transportation Holdings	470-555-2278	3732 Camino Norte
11.	K & R Brandon Communications Group	315-555-7829	8067 Olympic Drive

Figure 6-7 Output of the New Simple Report

6.3 Modifying the Query When Uploading a Report

The query in the report unit determines the data that the server retrieves from a data source. You can create multiple reports that look the same but contain different data by defining different queries for the same JRXML file each time you upload it. You can choose use a stored query in the repository or define a new one. This example uses a query that returns accounts from a single country.

You can only override the main query for a report (specifically, the query defined directly within the JRXML root element `<jasperReport>`). If the report has elements that use a sub-dataset (including all table elements and all elements that use a different dataset), the queries for these elements will not be affected.

First you need to upload the report and select a data source, as in the previous example. Then you can create a query which replaces the original query in the report.

To locate the sample report for this example:

1. Log into the server as administrator and select **View > Repository**.
2. Locate the folder where you want to add the report. For example, go to **Public > Samples > 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: `Sample Query Report`
   - Resource ID – Permanent designation of the report object in the repository: `Sample_Query_Report`
   - Description – Optional description displayed in the repository: `Example of changing a query in a report`
5. Select **Upload a Local File** and **Browse** to <js-install>/samples/reports/SimpleReport.jrxml.
6. Click **Open** to upload the file.

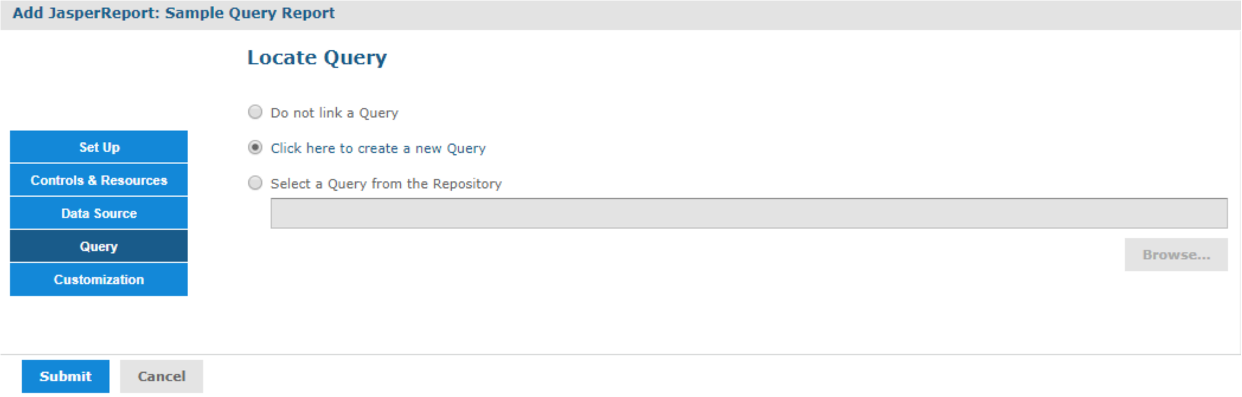
To select a data source for the report:

1. In the Add JasperReport wizard, click **Data Source**. The Link a Data Source to the Report page appears.
2. Choose **Select data source from the Repository** and **Browse** to **Public > Samples > Data sources > JServer JNDI Data Source**.
3. Click **Select**. The path to the data source appears on the page.

To define a custom query for the simple report example:

1. In the Add JasperReport wizard, click **Query**. The Locate Query page presents the following choices:
   - **Do not link a Query** – Select this option to use the existing 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 this option to use a saved query from the repository.

The SimpleReport.jrxml file already contains a query. Choosing the second or third option overrides the existing query by defining a new one.



Add JasperReport: Sample Query Report

Locate Query

○ Do not link a Query

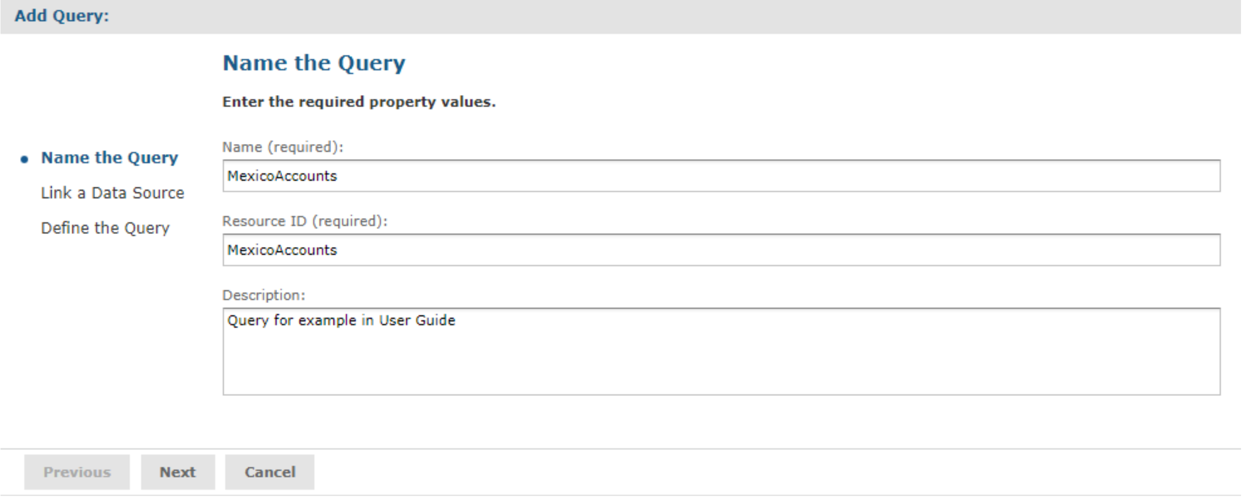
● Click here to create a new Query

○ Select a Query from the Repository

Figure 6-8 Query Page

2. Select **Click here to create a new Query**. The link becomes active.
3. Click the link, **Click here to create a new Query**. The Add Query wizard appears and displays the Name the Query page.
4. Enter the name, resource ID, and description of the query. The query in this example retrieves only Mexican accounts. Enter the following values:
   - Name – `MexicoAccounts`
   - Resource ID – `MexicoAccounts`
   - Description – `Query for example 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): MexicoAccounts

Resource ID (required): MexicoAccounts

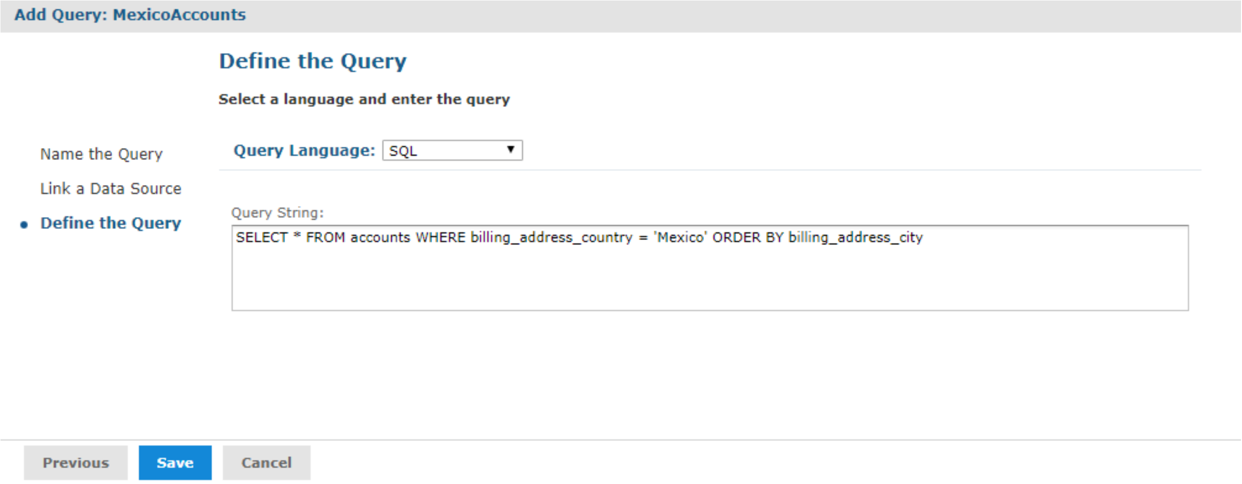
Description: Query for example in User Guide

Figure 6-9 Name the Query Page

5. Click **Next**. The Link a Data Source to the Query page appears. Here you have the option to select a data source to use only with this query. This can be different from the data source you selected for uploading the report. You can choose an existing data source 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 same data source you already selected.

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 Mexican accounts:

```
SELECT * FROM accounts WHERE billing_address_country = 'Mexico' ORDER BY billing_
address_city
```



Define the Query

Select a language and enter the query

Query Language: SQL

Query String:

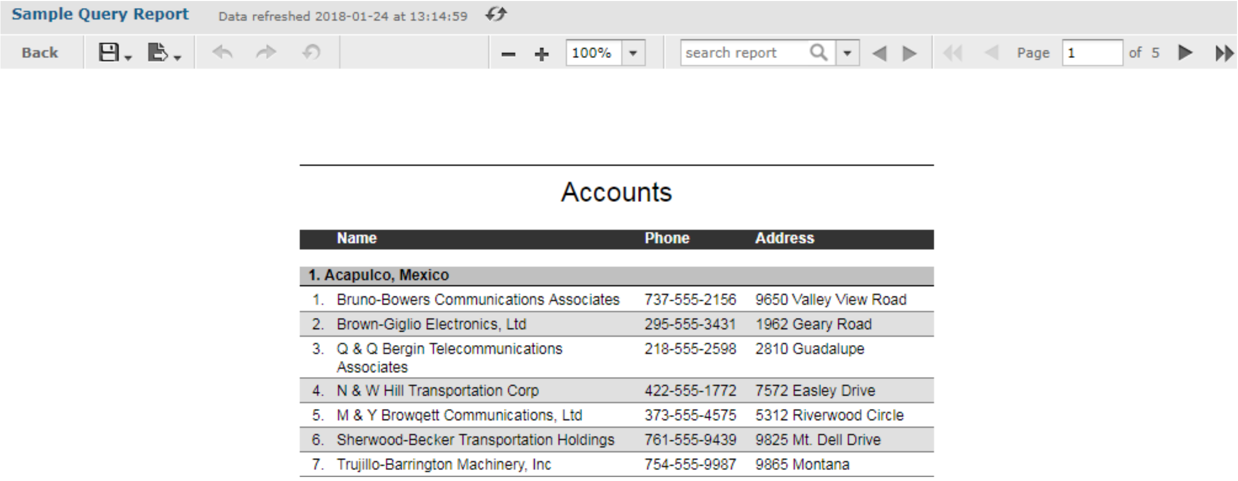
```
SELECT * FROM accounts WHERE billing_address_country = 'Mexico' ORDER BY billing_address_city
```

Figure 6-10 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.

To run the report:

1. Locate the report in the repository and click to run it. Only accounts in Mexico are shown in the report.



Accounts

	Name	Phone	Address
1. Acapulco, Mexico			
1.	Bruno-Bowers Communications Associates	737-555-2156	9650 Valley View Road
2.	Brown-Giglio Electronics, Ltd	295-555-3431	1962 Geary Road
3.	Q & Q Bergin Telecommunications Associates	218-555-2598	2810 Guadalupe
4.	N & W Hill Transportation Corp	422-555-1772	7572 Easley Drive
5.	M & Y Browgett Communications, Ltd	373-555-4575	5312 Riverwood Circle
6.	Sherwood-Becker Transportation Holdings	761-555-9439	9825 Mt. Dell Drive
7.	Trujillo-Barrington Machinery, Inc	754-555-9987	9865 Montana

Figure 6-11 Output of the Report

6.4 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.

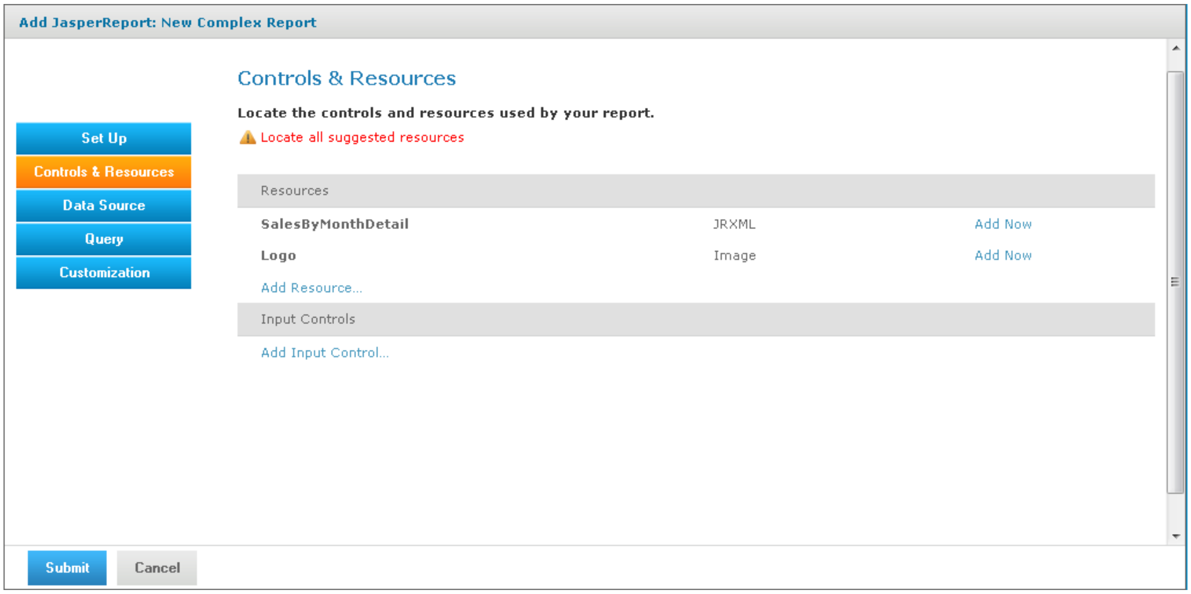
2. Navigate to the folder containing your report. For example, navigate to **Organization > Reports**.
3. Right-click the Reports folder and select **Add Resource > JasperReport** from the context menu. The Set Up the Report page appears.

   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** and **Browse** to <js-install>/samples/reports/SalesByMonth.jrxml.
6. Click **Controls & Resources**.

   The Controls & Resources page, **Figure 6-12**, 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 6-12 Suggested Resources for the Complex Report

7. 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.
8. 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.

6.4.1 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 before the server can upload 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:

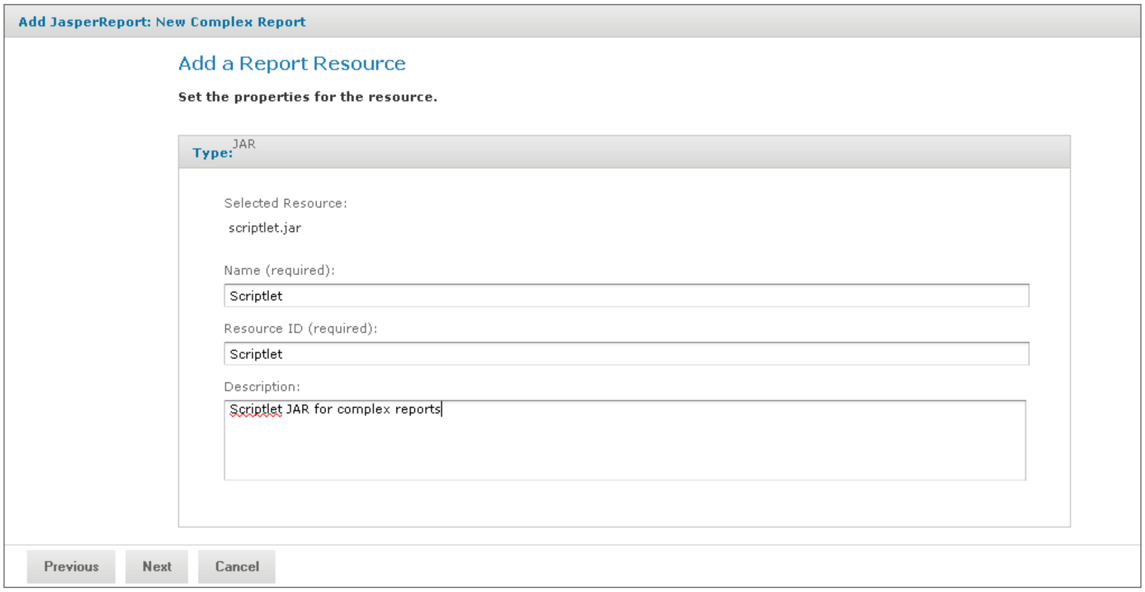
- A scriptlet JAR – The scriptlet writes the message, "I'm a scriptlet in a jar," to the last page of the report output.
- An English language resource bundle.
- The optional Romanian language resource bundle.
  If you're interested in working with a multi-lingual report, add the Romanian resource bundle. 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 **Browse** to 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 6-13 on page 184** 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 6-13 on page 184 shows these values entered on the Add a Report Resource page.



Add JasperReport: New Complex Report

Add a Report Resource

Set the properties for the resource.

Type: JAR

Selected Resource:
scriptlet.jar

Name (required):
Scriptlet

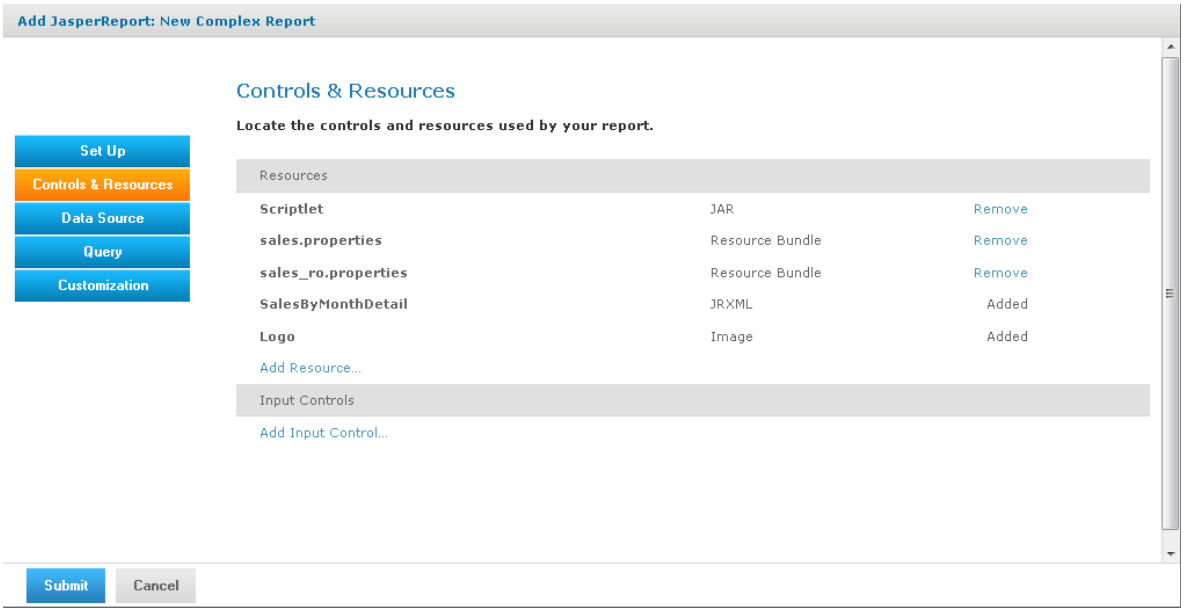
Resource ID (required):
Scriptlet

Description:
Scriptlet JAR for complex reports

Figure 6-13 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**, **Browse** to <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**, **Browse** to 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 6-14 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.

6.4.2 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.

6.4.2.1 Adding a Text Input Control

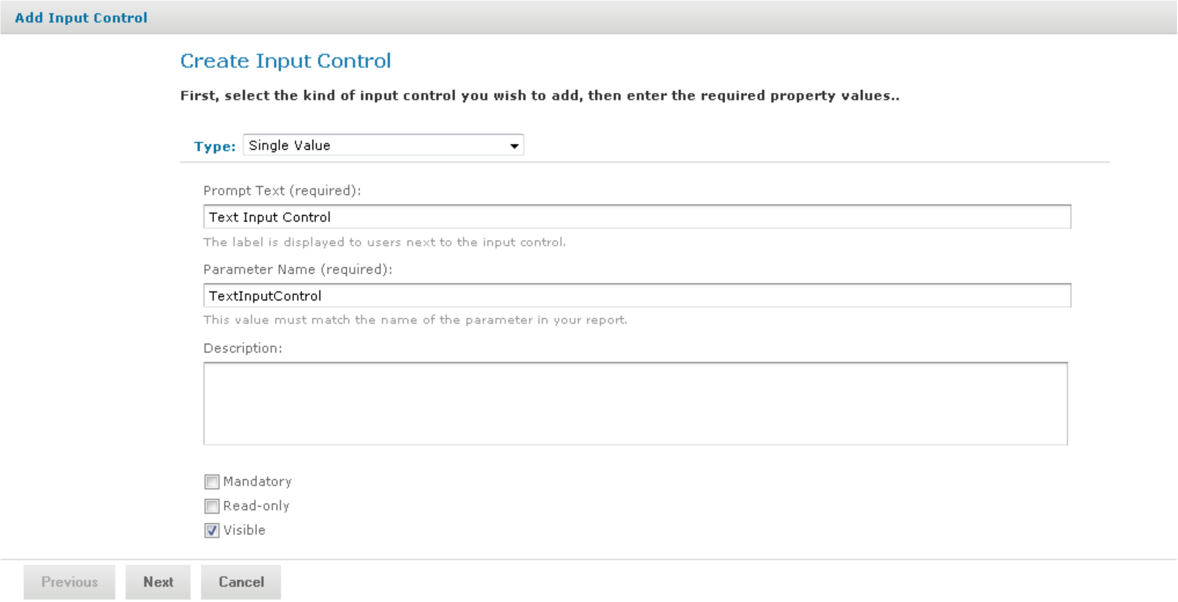
The simplest input control is a text box. In this example, the datatype for the input value is a number; the server verifies that the user enters a number into the text box.

To add a text input control to the complex report example:

1. After completing steps in **"Uploading Undetected File Resources" on page 183**, 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 appears: 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 6-15 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**:

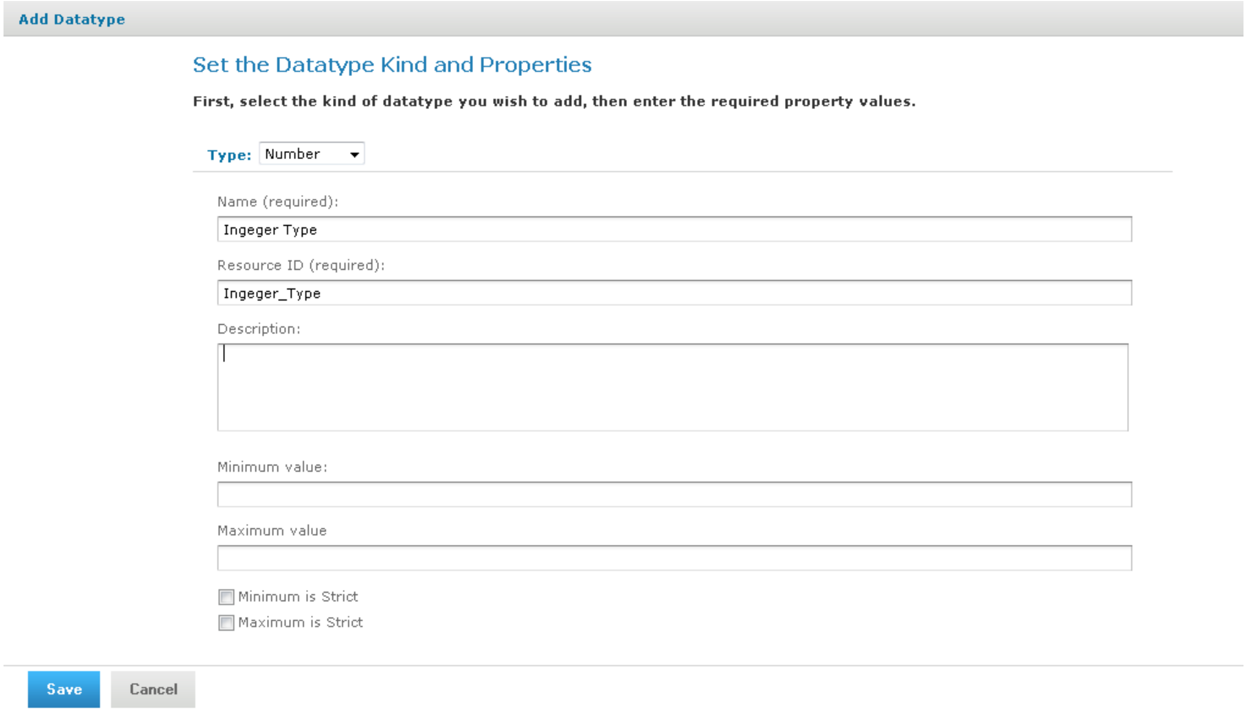
Instead of defining a datatype, you can use one in the repository if its type and range are compatible with your input control.

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.



Add Datatype

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

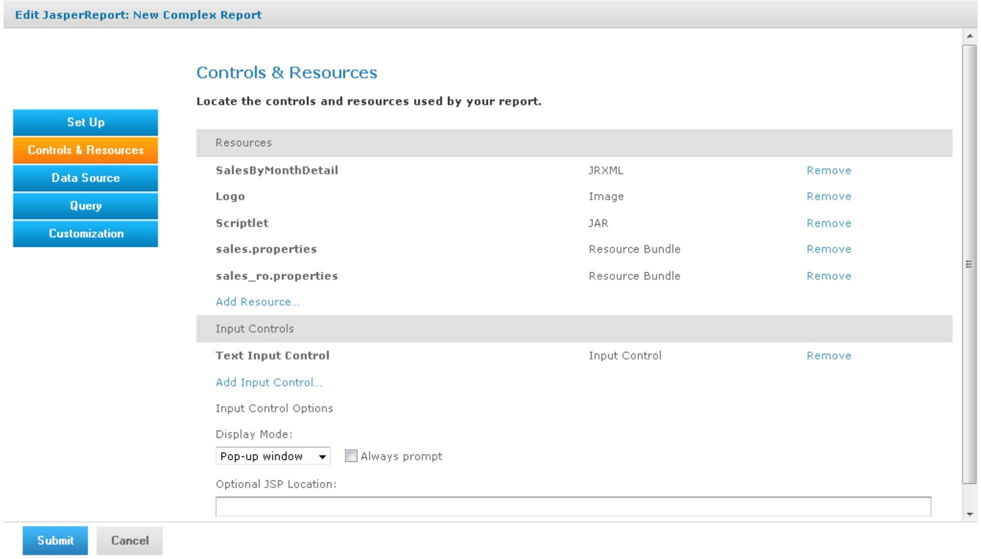
□ Maximum is Strict

Figure 6-16 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
Add Resource...		

Input Controls		
Text Input Control	Input Control	Remove
Add Input Control...		

Input Control Options

Display Mode: Pop-up window □ Always prompt

Optional JSP Location:

Figure 6-17 **Text Input Control in Input Controls List**

6.4.2.2 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.

6.4.2.3 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**.

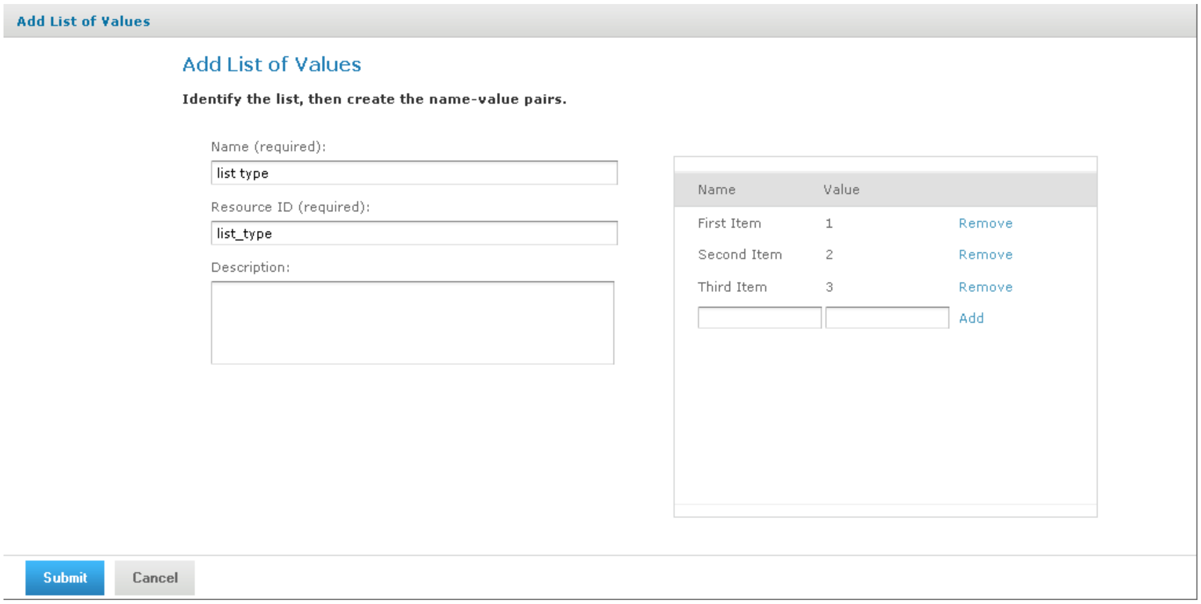
   Instead of defining a list of values, you can use one in the repository if its values are compatible with the parameter defined in the JRXML report.

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

Identify the list, then create the name-value pairs.

Name (required): list type

Resource ID (required): list_type

Description:

Name	Value
First Item	1
Second Item	2
Third Item	3

Figure 6-18 **Definition of the List of Values**

11. Click **Submit**. The Controls & Resources page appears with the new List Input control.

6.4.2.4 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.

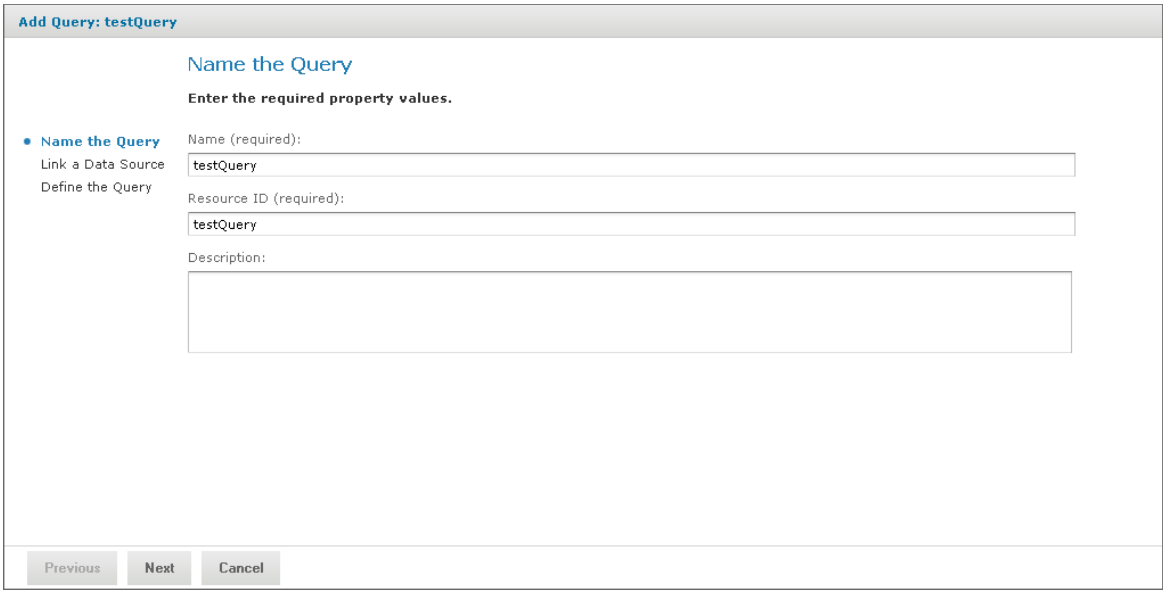
6.4.2.5 Adding a Query-Based Input Control

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.

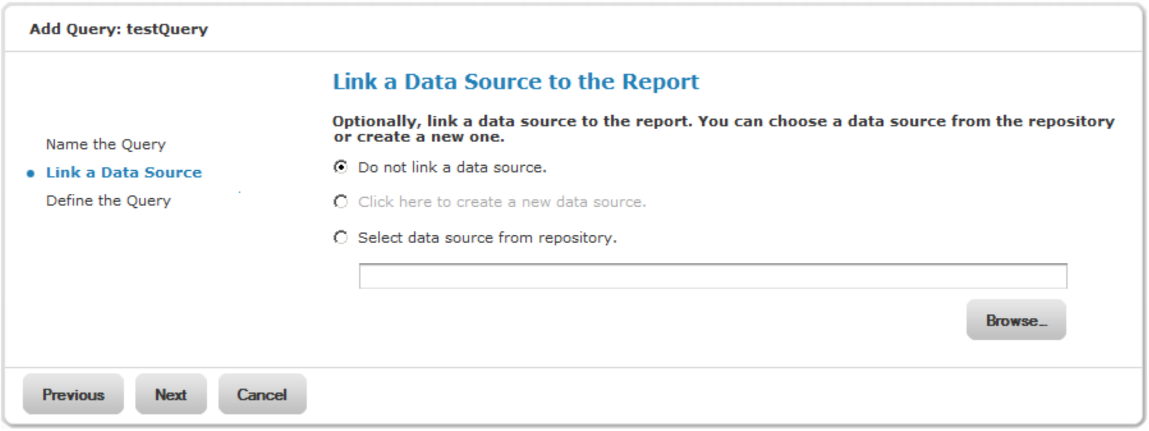
Name (required): testQuery

Resource ID (required): testQuery

Description:

Figure 6-19 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 196**.



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.

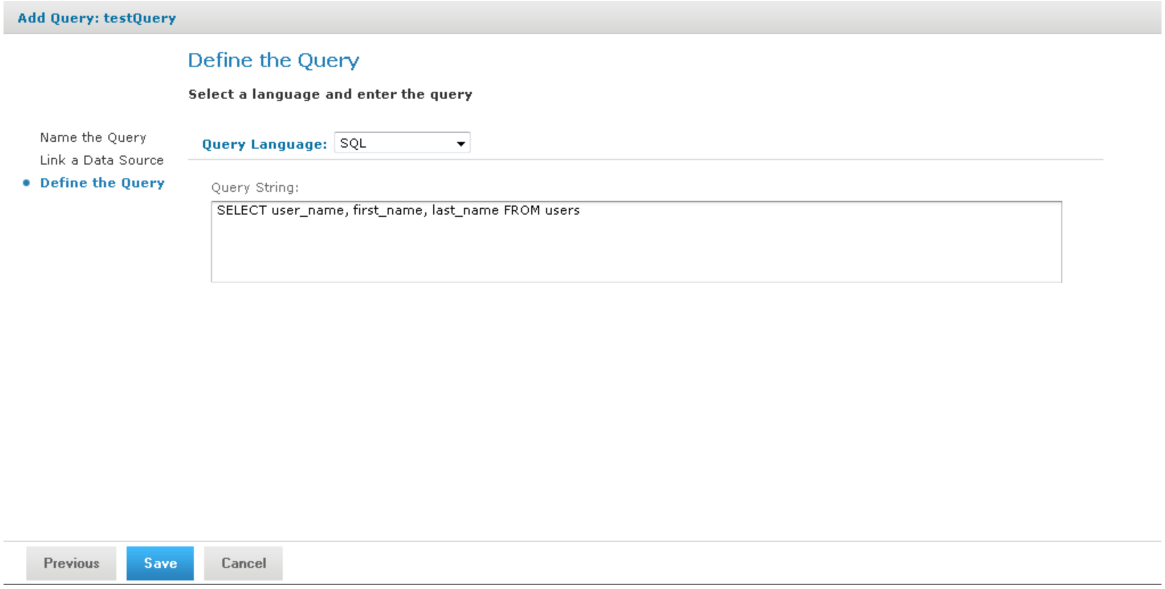
○ Click here to create a new data source.

○ Select data source from repository.

Figure 6-20 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 6-21 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 6-22 on page 196** shows these resources.

6.4.2.6 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 6-22 on page 196** shows these options.

To configure the appearance of the input controls for the complex report example:

1. Select **Pop-up window**.

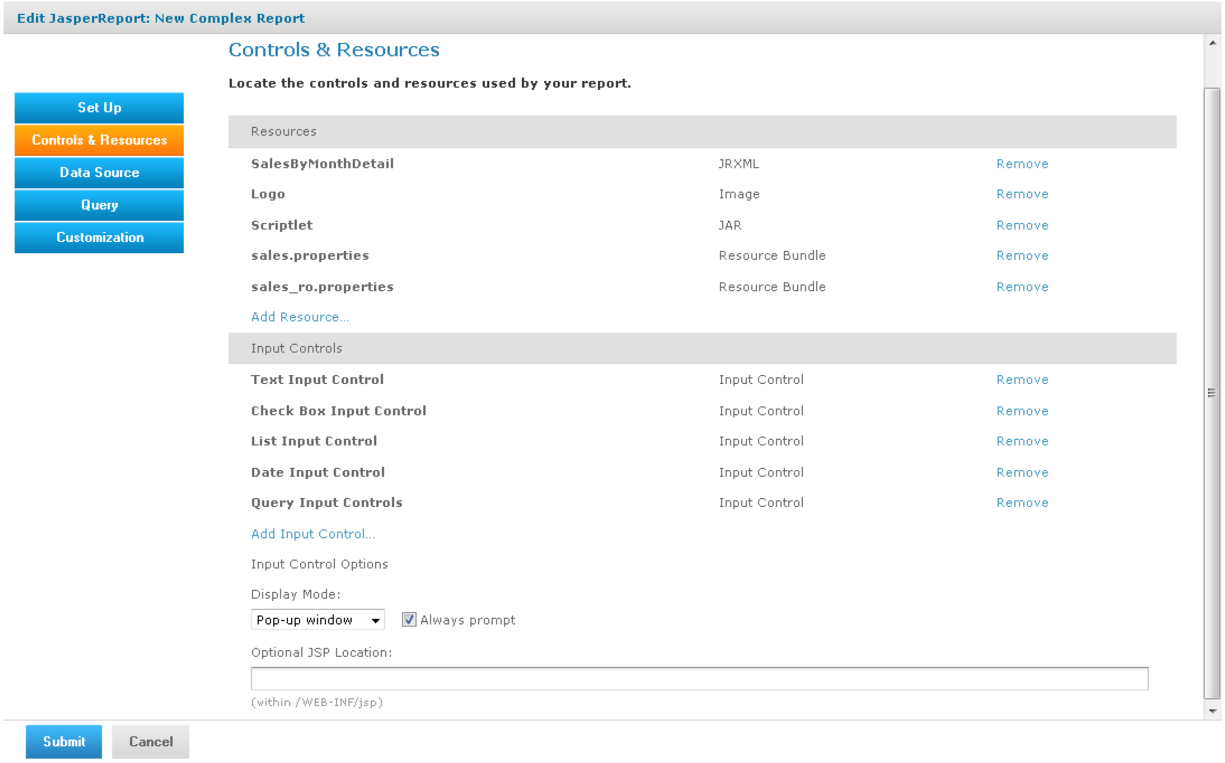
> You can also select **Separate page** to display the input controls in a separate browser window, **Top of page** to display them above the report, or **In page** to display them on the side of the report.

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 6-22 Input Controls and Resources

Select the data source to finish the complex report example.

6.4.3 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 6-23**:

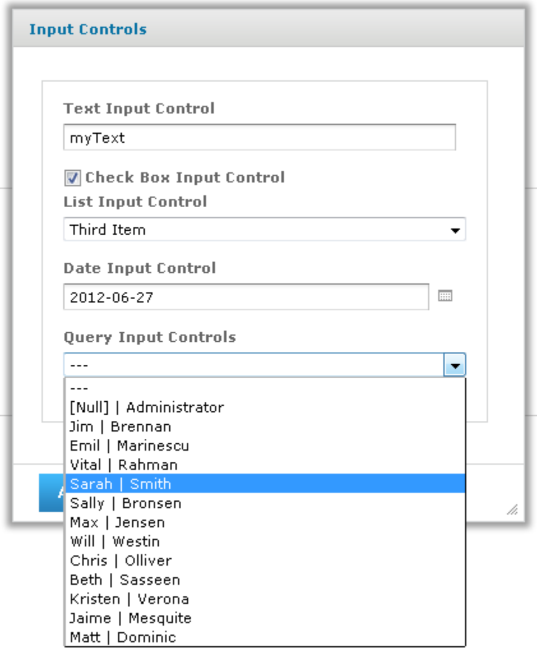
   a. Text Input Control: `myText`

   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

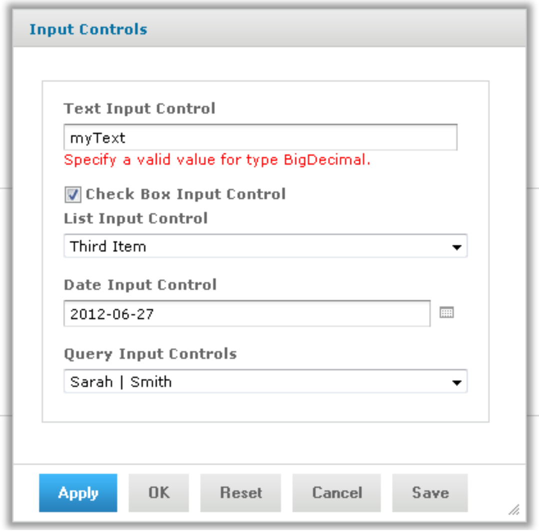
Date Input Control: 2012-06-27

Query Input Controls: ---

Figure 6-23 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 6-24**.



Input Controls

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

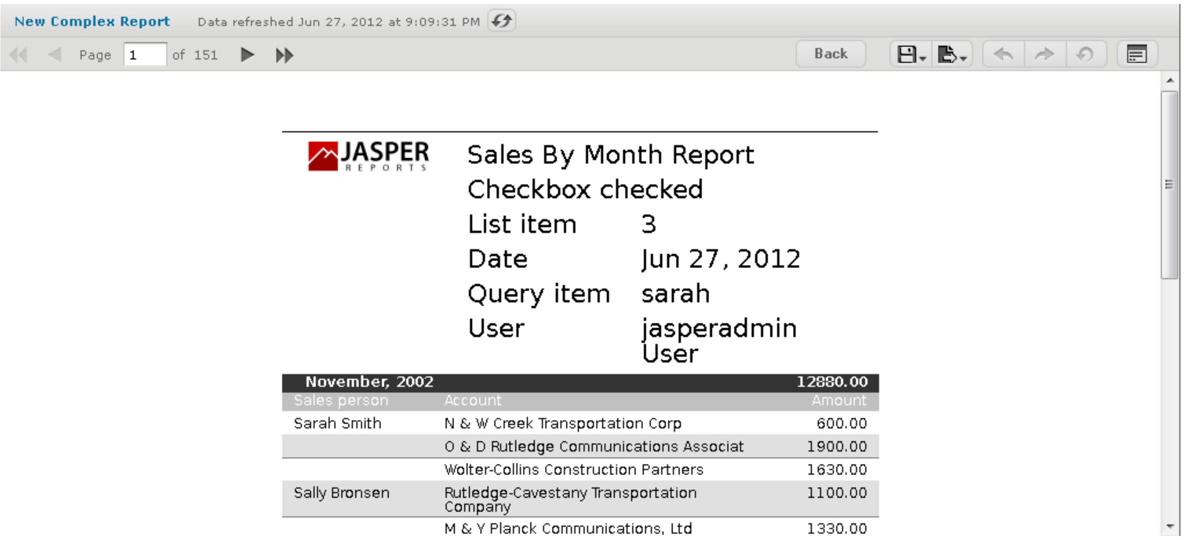
Query Input Controls

Sarah | Smith

Figure 6-24 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.



New Complex Report   Data refreshed Jun 27, 2012 at 9:09:31 PM

Page 1 of 151

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 6-25 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 199**. 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.

6.5 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 75**.

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*.

6.6 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:

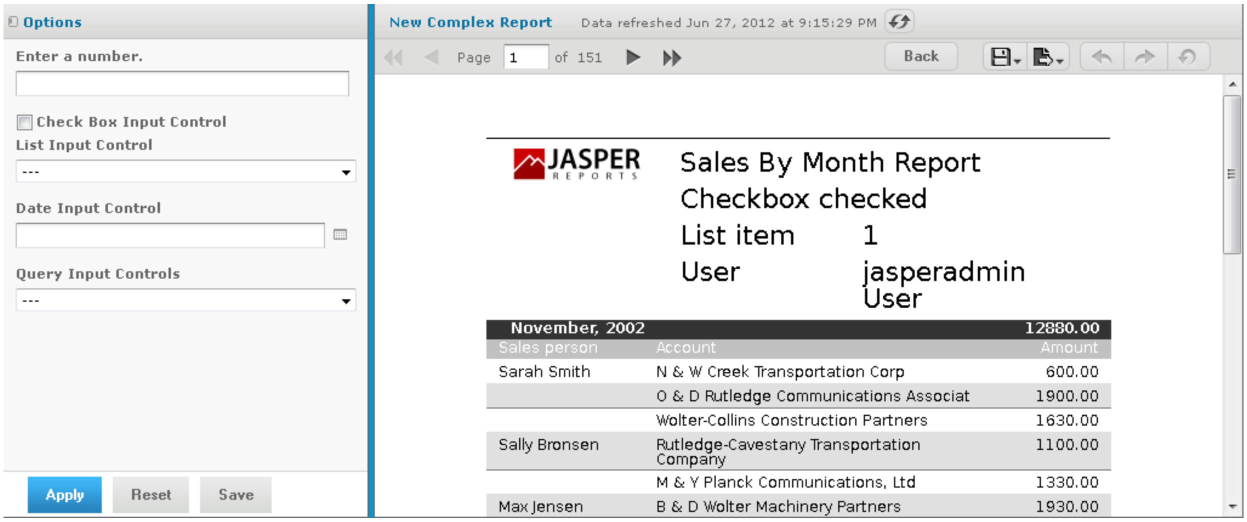
1. Log into the server as an administrator and select **View > Repository**

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 6-26 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.



Enter a number.

□ Check Box Input Control

List Input Control: ---

Date Input Control:

Query Input Controls: ---

Data refreshed Jun 27, 2012 at 9:15:29 PM

Page 1 of 151

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 6-26 Output of the Modified Report

6.7 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.

6.7.1 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 183**.

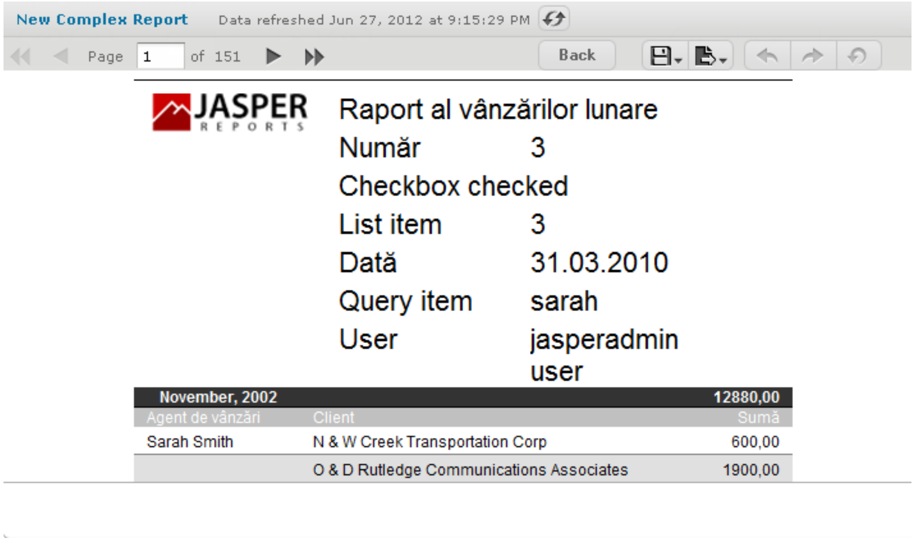
To run the Romanian version of the complex report:

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 6-27**, 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 6-27 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*.

6.7.2 Reusing Domain Localization Files

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 **"Localizing Domains" on page 302**).
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.

6.7.3 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 7 Working with the Domain Designer

A Domain is a metadata layer that models and presents data accessed through a data source. You can use Domains to structure the data and present it in business terms appropriate to your audience. You can also limit access to data based on the security permissions of the person running the report and provide translations for static report text. Domains in JasperReports Server are used to create Ad Hoc views and reports.

This chapter covers the process of creating a Domain and defining its contents. For instructions about creating views based on Domains in the Ad Hoc Editor, see **"Creating a View from a Domain" on page 160**. Domains defined in the server can be accessed through Jaspersoft Studio as well.

This chapter contains the following sections:

- **Understanding Domains**
- **Opening the Domain Designer**
- **Overview of the Domain Designer**
- **The Data Management Tab**
- **The Joins Tab**
- **Understanding Joins and Join Trees**
- **The Pre-filters Tab**
- **The Data Presentation Tab**
- **The Options Tab**
- **Derived Tables and Calculated Fields**
- **Using Attributes in the Domain Designer**
- **Editing Domains**
- **Importing and Exporting Domain Design Files**
- **Domains and Data Virtualization**

7.1 Understanding Domains

Domains model and present data that report creators may not understand in raw form. Production databases typically contain data in tables optimized for storage and retrieval. A Domain is a virtual view, created and stored in JasperReports Server without modifying the data source, that models and presents the data in a way that is more accessible to report creators and readers. Columns with data relevant to users can be joined across several tables, filtered by business need, secured against unauthorized access, and labeled with user-friendly names. A Domain also supports the localization of static text and allows you to limit the data users can access permissions based on data values.

7.1.1 Who Uses Domains?

Domains help make data accessible to different types of users:

- Database administrators or business analysts create Domains, choosing and modeling the data to expose, and creating user-friendly labels where appropriate. These users have an understanding of the raw data as it appears in the data source.

In the default server installation, Domain creators must have organization-level admin privileges. See *JasperReports Server Administrator Guide* for more information.

- Report designers use Domains to create Ad Hoc views in JasperReports Server or reports in Jaspersoft Studio. These users understand the data in the context of their specific business case, but might not understand the raw data in the original data source.
- Report readers see only the data they need and understand.

7.1.2 What Does a Domain Do?

A Domain has multiple levels that allow you to model and present data:

- Select schemas and tables from your data source on the Data Management tab, as well as create derived tables using SQL queries. See **7.3.4, "The Data Management Tab," on page 212** and **7.4.1, "Derived Tables," on page 241** for more information.
- Define joins between any included tables and/or derived tables on the Joins tab. See **7.3.5, "The Joins Tab," on page 217** for more information.
- Set conditions on field values to limit the data accessed through the Domain on the Pre-filters tab. See **7.3.6, "The Pre-filters Tab," on page 229** for more information. You can also filter data using calculated fields; see **7.4.2, "Calculated Fields," on page 242** for more information.

A Domain design often includes data filtering, but report creators can filter data even further or create input controls for report readers to use.

- Choose which tables and columns to expose to users on the Data Presentation tab, and optionally define or change display properties. See **7.3.7, "The Data Presentation Tab," on page 231** for more information.
- Upload files for security and localization on the Options tab. See **7.3.8, "The Options Tab," on page 237** for more information.

Each Domain is stored in the repository as an XML design file along with the files the Domain references, such as a data source, locale bundles for localization, or a security file. You can create Domains in the Domain Designer, or you can upload a Domain design file in XML format. Localization bundles and security files must be created in text format and uploaded. See **7.3.8, "The Options Tab," on page 2378.3, "Localizing Domains," on page 302**, and **8.4, "The Domain Security File," on page 308** for more information.

7.1.3 Planning 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, choose your data virtualization strategy. Either define each data source in the repository and then create a virtual data source that combines them, or use Jaspersoft Advanced Data Services and define a data source based on that. See **7.7, "Domains and Data Virtualization," on page 253** for more information.

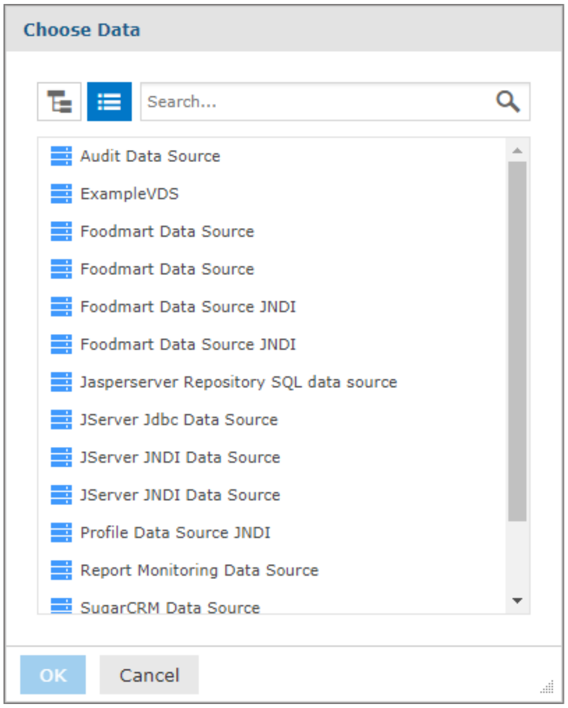
- If you want to use attributes, make sure they are defined and that you understand how they will work. This is especially important if you are using attributes for schema names. See **7.5, "Using Attributes in the Domain Designer," on page 245** for more information.
- Know the elements of your design: tables and columns to choose, joins to perform, pre-filters to define, and items you want to present to users.
- Determine whether the users of the Domain need localized resources, such as translated labels and local formats.
- Determine a security policy for the data retrieved through the Domain.

7.2 Opening the Domain Designer

To open the Domain Designer and create a new Domain:

1. Log in as an administrative user.
2. Create a Domain in one of the following ways.
   - Select **Create > Domain** from the main menu.
   - Click **Create** in the Domains section of the home page.
   - Right-click a repository folder and choose **Add Resource > Domain**.

   The Choose Data dialog appears.



Audit Data Source
ExampleVDS
Foodmart Data Source
Foodmart Data Source
Foodmart Data Source JNDI
Foodmart Data Source JNDI
Jasperserver Repository SQL data source
JServer Jdbc Data Source
JServer JNDI Data Source
JServer JNDI Data Source
Profile Data Source JNDI
Report Monitoring Data Source
SugarCRM Data Source

Figure 7-1 The Choose a Data Source dialog

3. Select a data source in the Choose Data dialog and click **OK**.
   - Use only data sources in the repository for which the intended user has at least execute permission.

- If you installed the sample data, you can use the sample JDBC, JNDI, and virtual data sources in **Organization > Data Sources** or **Organization > Analysis Components > Analysis Data Sources**.
- You can use the icons at the top of the Choose Data dialog to switch between a repository tree view and a flat list of data sources:

  – Display data sources as a repository view.

  ☰ – Display data sources as a list.
- For performance reasons, the Choose Data dialog has an upper limit on the number of data sources it displays. If the data source you want does not appear in the list, use the **Search...** bar 🔍 to locate it.

4. Click **OK** to open the Domain Designer with the selected data source.

To edit an existing Domain:

1. Log in as an administrative user.
2. Display a list of Domains in one of the following ways:
   - Navigate to a folder in the Repository that contains Domains.
   - Click **View list** in the Domains section on the home page.
3. Right-click the Domain you want and select **Edit** from the menu.

   The Domain Designer displays the Data Management tab for the Domain you selected.

7.3 Overview of the Domain Designer

The Domain Designer is a tool for defining the components of a Domain. Tabs let you choose schemas and tables in your data source, join tables, pre-filter data, and choose the data to present to report creators and users. You can also create calculated fields and derived tables, and upload text files to translate static text or apply security to the Domain.

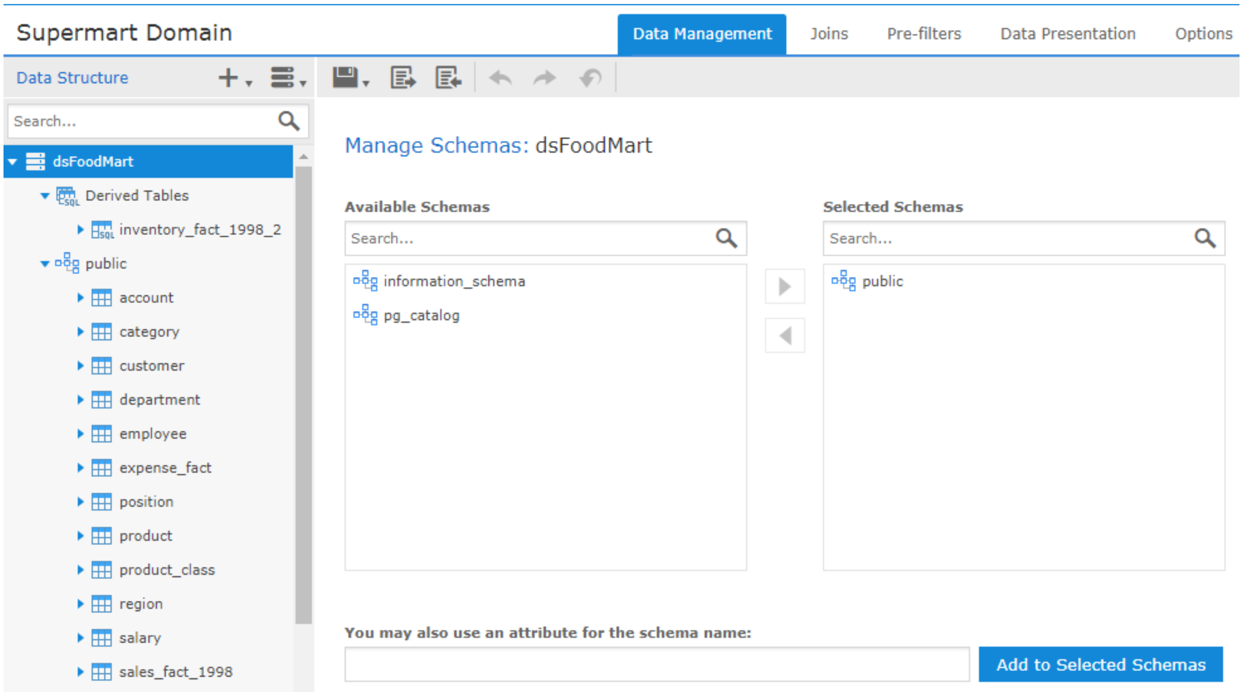


Figure 7-2 User Interface of the Domain Designer

7.3.1 Domain Designer Tabs

Use the tabs at the top of the Domain Designer to view and edit various aspects of the design. To navigate among tabs, click a tab name at the top of the Domain Designer:

- Data Management tab – Select schemas and tables you want to use in the Domain, including tables you refer to but might not want to expose. See **7.3.4, "The Data Management Tab," on page 212** for more information.
- Joins tab – Define joins between any included tables and/or derived tables. See **7.3.5, "The Joins Tab," on page 217** for more information.
- Pre-filters tab – Set conditions on field values to limit the data accessed through the Domain. See **7.3.6, "The Pre-filters Tab," on page 229** for more information.
- Data Presentation tab – Create sets that specify tables, columns, and items to expose to users. Optionally define or change display properties, such as names and descriptions. See **7.3.7, "The Data Presentation Tab," on page 231** for more information.
- Options tab – Upload files for security and localization. See **7.3.8, "The Options Tab," on page 237** for more information.

All tabs except the Options tab have at least two panels, from left to right:

- **Data Structure**, which displays the schemas, tables, columns, and joins available on the current tab. Use the search bar at the top of the Data Structure tab to locate a Domain element. Not all elements are available on all tabs.
- A design panel on the right, with a working area specific to the current task. In some cases, the design panel is divided into sub-panels.

7.3.2 Domain Designer Tool Bar

Use the icons on the tool bar to create derived tables and calculated fields, change the data source, save the Domain, and import and export Domain design files.

Table 7-1 Domain Designer Tool Bar Icons

Icon	Name	Description
+ ▾		Place the cursor over this icon to display a menu with options for creating constant fields and derived tables.
☰ ▾	Data Source	Place the cursor over this icon to display a menu of options for changing data.
💾 ▾	Save	Place the cursor over this icon to display a menu of options for saving the Domain.
📝➜	Export	Click this icon to export a Domain design file.
📝⬅	Import	Click this icon to import a Domain design file.
↩	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 Domain to its state when you last saved.

Before you can save the design, you must add at least one table to the Domain. The 💾 ▾ button on the tool bar validates the design and saves it in the location you specify. For more information, see **“Domain Validation” on page 252**. After saving a Domain, you can continue modifying it using the Domain Designer.

7.3.3 The Data Structure Panel

The Data Structure panel contains a hierarchical view of the available schemas, tables, columns, and joins. The available elements and the layout depend on the current tab. A search bar lets you search for items in the panel by name.

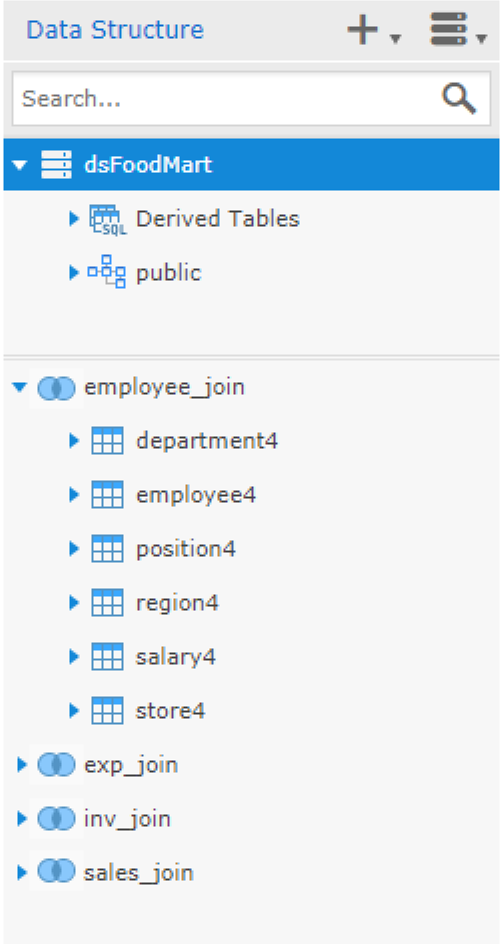


Figure 7-3 Example of Data Structure panel on Joins tab

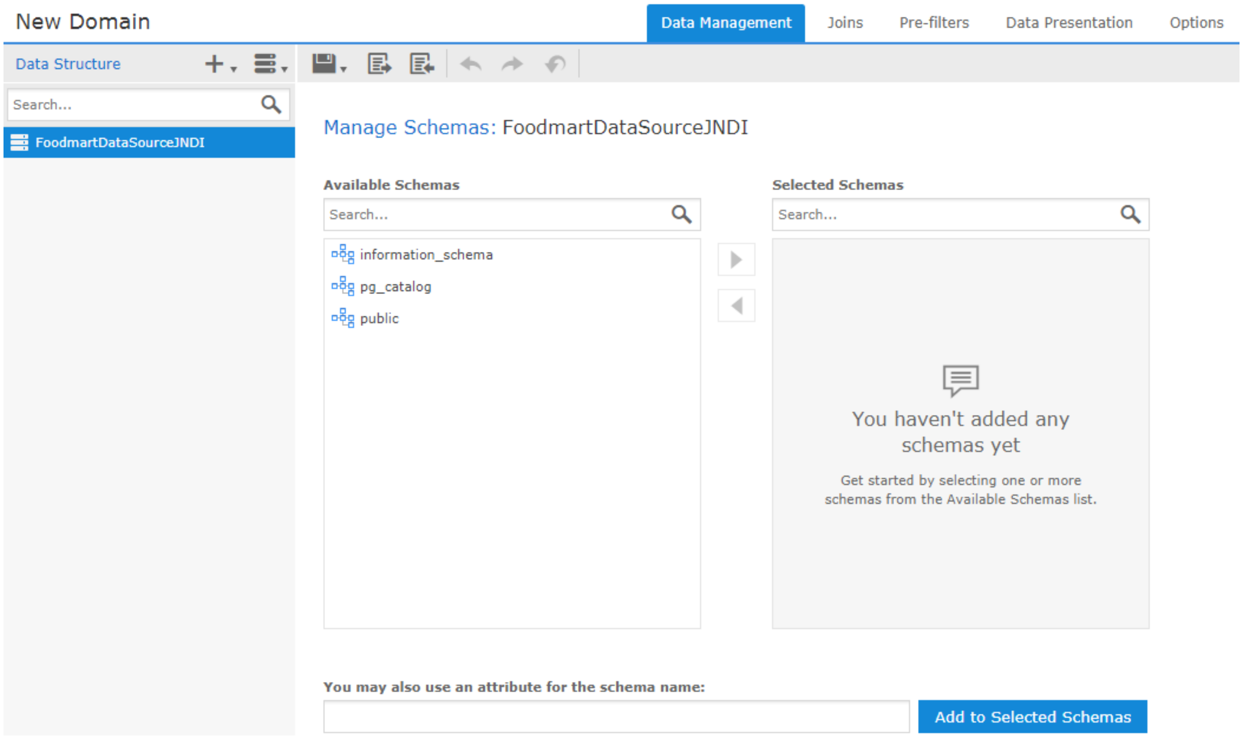
The following icons to indicate element type can appear on the Data Structure panel or elsewhere in the Domain Designer:

Icon	Description
≡	The data source for the Domain. There is only one data source for a Domain.
⊞	A database schema that has been added to the Domain.
attr	A database schema that has been added using an attribute for the schema name.
▦	A table that has been added to the Domain.
▯	A column (field) in the Domain.

Icon	Description
▦f	Constant Fields node. Only appears if you have created one or more constant calculated fields.
f	A calculated field or constant field that has been added to the Domain. Constant fields appear under the Constant Fields node; other calculated fields appear under the join or table where they were created.
▦SQL	Derived Tables node. Only appears if you have created one or more derived tables.
SQL	A derived table that has been added to the Domain.
⚭	A join tree. A Domain may contain many join trees, but when a user creates an Ad Hoc view from a Domain, they can only select one join tree to use in the view.
▦	A data island. A data island contains the tables and columns from a specific join tree that you have chosen to expose to users. Data Islands are only visible on the Data Presentation tab.
❐	A set, that is, a named collection of items grouped together for ease of use in the Ad Hoc Editor. Sets are visible only on the Data Presentation tab.
▯	An item, that is, the representation of a database field or a calculated field along with its display name and formatting properties. Items are visible only on the Data Presentation tab. Items can be grouped in sets and are used in the creation of Ad Hoc views.

7.3.4 The Data Management Tab

The Data Management tab is where you manage the schemas and tables from your data source and choose which ones to include in your Domain.



Manage Schemas: FoodmartDataSourceJNDI

You haven't added any schemas yet

Get started by selecting one or more schemas from the Available Schemas list.

You may also use an attribute for the schema name:

Figure 7-4 The Data Management tab before a schema has been added to the Domain

7.3.4.1 The Data Structure Panel on the Data Management Tab

On the Data Management tab, the Data Structure panel displays the data source and the schemas, tables, and columns from the data source that you have added to the Domain. It also displays any constant fields and derived tables you have created.

The Data Structure panel on the Data Management tab does not display internally-defined Domain elements, such as joins, calculated fields, or copied tables. It does display derived tables.

The Data Source panel shows columns that have a supported type listed in **Supported Types**. 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*.

- Expand the data source to see the schemas you have added to your Domain design. If you have added any derived tables, they appear under a separate node.
- Select the data source to display the Manage Schemas panel and add or remove schemas from your design.
- Right-click the data source to display a context menu with the following choices:
  - **Create Derived Table...** – Select this to open the New Derived Table dialog and create a derived table. See **7.4.1, “Derived Tables,” on page 241** for more information.
- Expand a schema to see which of its tables you have added to your design.
- Select a schema to display the Manage Tables panel for that schema and add or remove tables from your design.
- Expand the Derived Tables node to view the derived tables you have created.

- Right-click a derived table to copy, edit, or remove it.
- Expand a table or derived table to view its columns.

> You cannot select tables or columns in the Data Structure panel on the Data Management tab. You can select them on the Joins, Pre-filters, and Data Presentation tabs.

7.3.4.2 Managing Schemas

If your data source supports database schemas, like Oracle RDBMS, you need to choose one or more schemas to use in the Domain. You also need to select schemas when using a virtual data source.

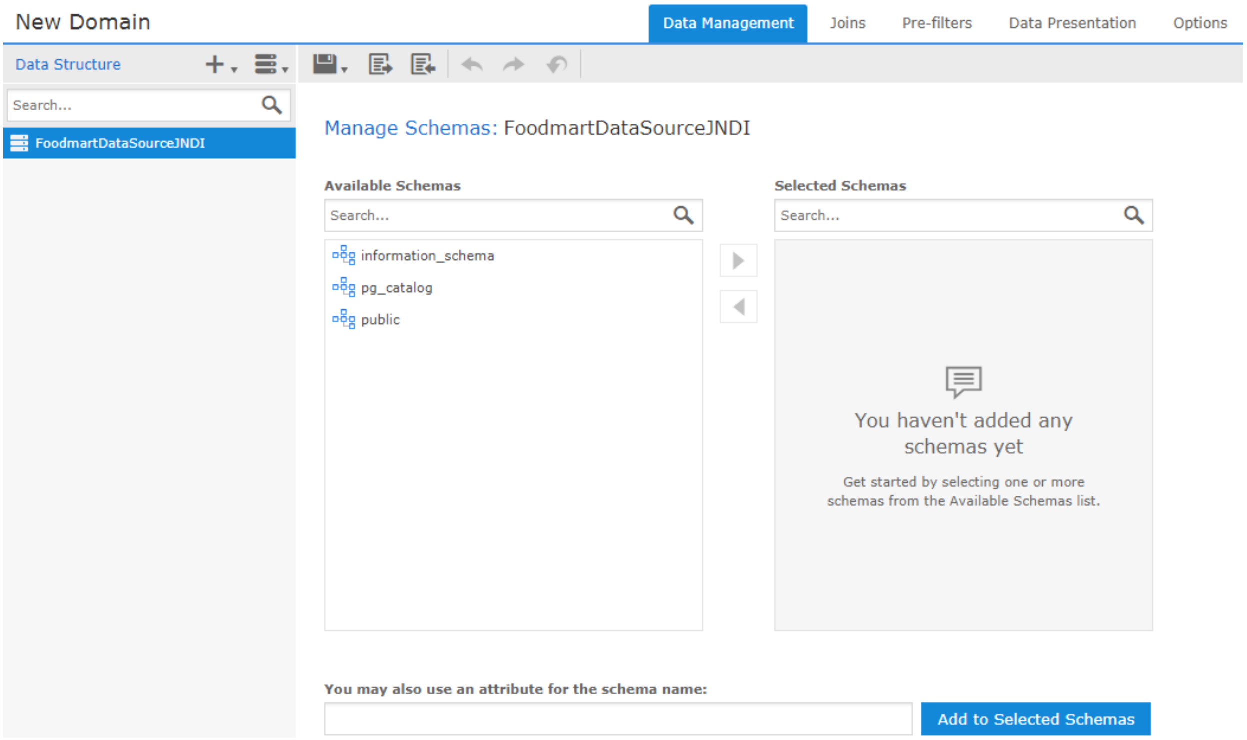


Figure 7-5 Select Database Schemas

To add schemas, first select the data source in the Data Structure panel on the Data Management tab. The Manage Schemas panel appears in the Data Design panel.

> The Manage Schemas panel appears automatically when you create a new domain.

- The **Available Schemas** list displays the available schemas in your data source.
- The **Selected Schemas** list shows the schemas you selected.
- Selected schemas appear in the Data Structure panel. If you have not yet added any tables from the schema to the Domain design, ❗ appears shown to the right of the schema name. See 7.3.4.3, “Managing Tables,” on page 216 for more information.
- You can move a schema back and forth between the lists by dragging, double-clicking, or selecting the item and clicking an arrow button, such as ▶ .

7.3.4.2.1 Using Attributes

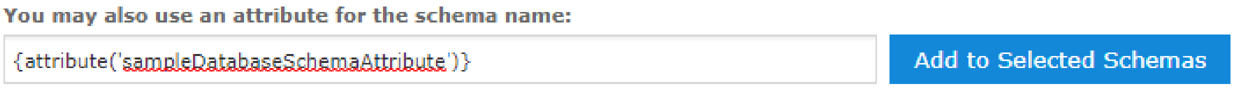
You can use an attribute for a schema name. When you use an attribute for the schema name, the Domain Designer displays current name of the attribute as the schema and retrieves the tables and columns from that schema. See **7.5, “Using Attributes in the Domain Designer,” on page 245** for more information.

When you use an attribute for a schema name, the Domain will fail if the schema is missing. You can optionally set a default schema in the Domain design file. The default schema is used whenever the schema is missing or undefined. See **8.1.5.5.1, “Default Schema,” on page 264** for more information.

To use an attribute for a schema:

1. Make sure you have defined the attribute you want. See the *JasperReports Server Administrator Guide* for information about creating attributes.
2. Enter the correct syntax for the attribute in the **You may also use an attribute for the schema name:** text box, depending on the attribute level:
   - `{attribute('attributeName'), 'server'}` for a server-level attribute.
   - `{attribute('attributeName', 'organization')}` for an organization-level attribute.
   - `{attribute('attributeName', 'user')}` for a user-level attribute.

   If no level is specified, the server will search for the attribute hierarchically, starting at the `'user'` level:
   - `{attribute('attributeName')}`



You may also use an attribute for the schema name:

{attribute('sampleDatabaseSchemaAttribute')}

Figure 7-6 Using a server-level attribute for a schema name

If you forget the attribute syntax, enter any string in the **You may also use an attribute for the schema name:** text box and click **Add to Selected Schemas** to see a hint with the correct syntax.

3. Click **Add to Selected Schemas** to add the attribute to the Selected Schemas list.

   The attribute is resolved to its current value and the corresponding schema is shown in the Selected Schemas list. A special icon appears.

You can view the attribute information for a schema, if any, by going to the Data Management tab, clicking on the data source node, and then selecting the schema in the Selected Schemas list. The attribute information is show in the **You may also use an attribute for the schema name:** text box.

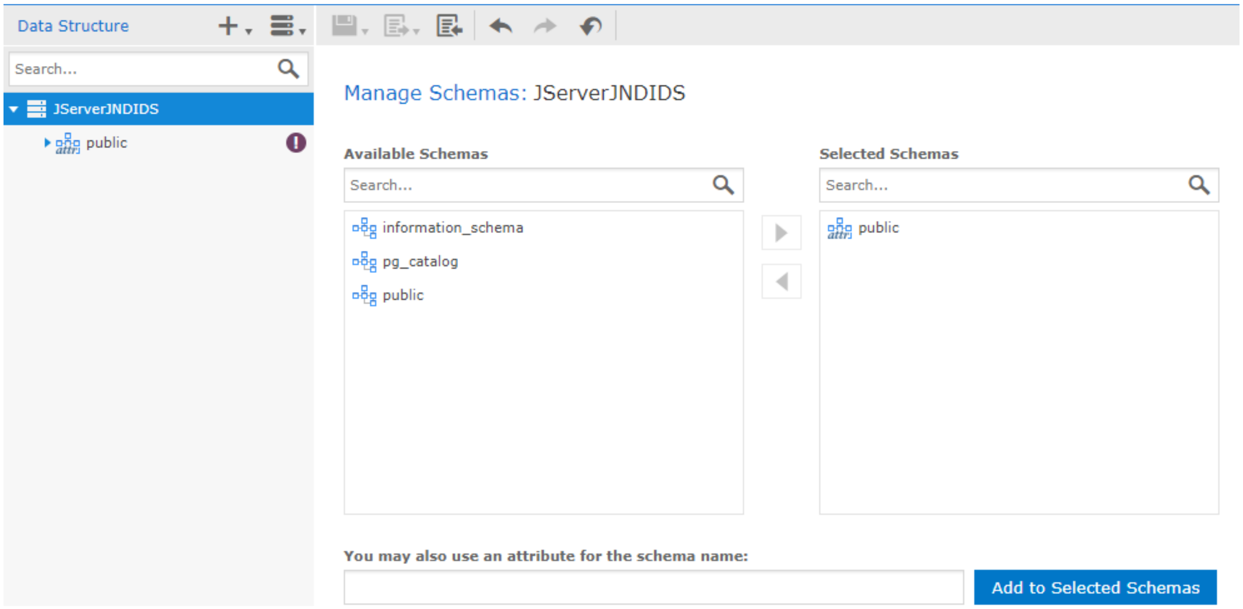


Figure 7-7 Result of using an attribute for a schema name

7.3.4.3 Managing Tables

You need to choose one or more tables to use in the Domain. Typically, you select the following tables for use in the Domain:

- Tables that you want to expose to report designers and users.
- Tables that you want to join to other tables.
- Tables that you want to reference in the Domain design, even if their columns do not appear directly in the Domain. For example, make sure to 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. Once you have added a table, you can expand it in the Data Structure tab to verify it contains the columns you want.

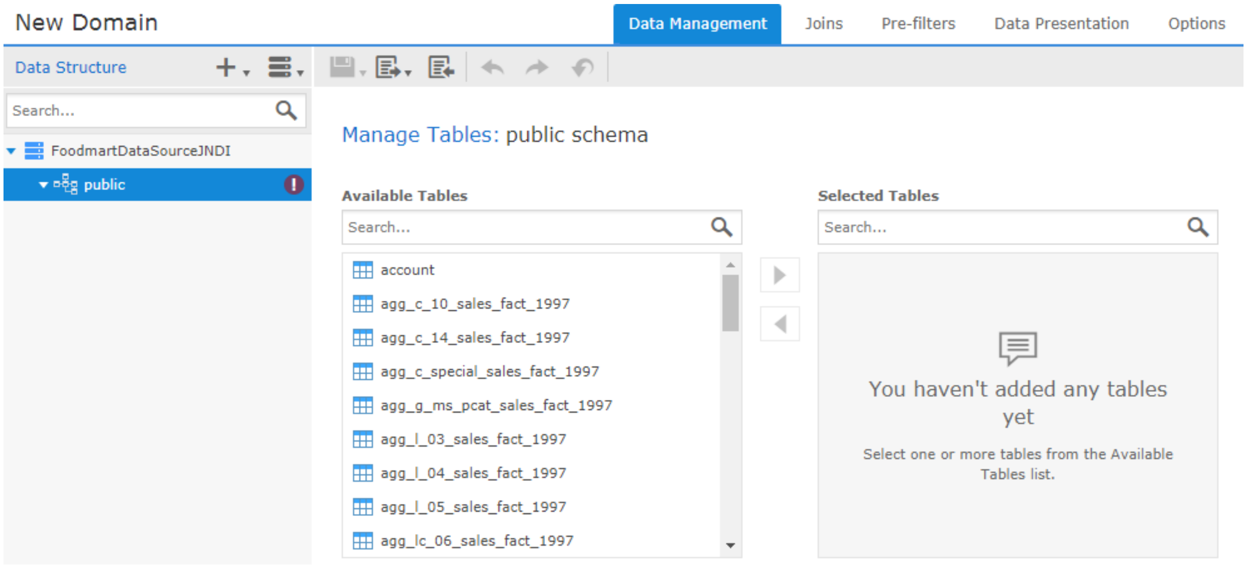


Figure 7-8 Select Tables

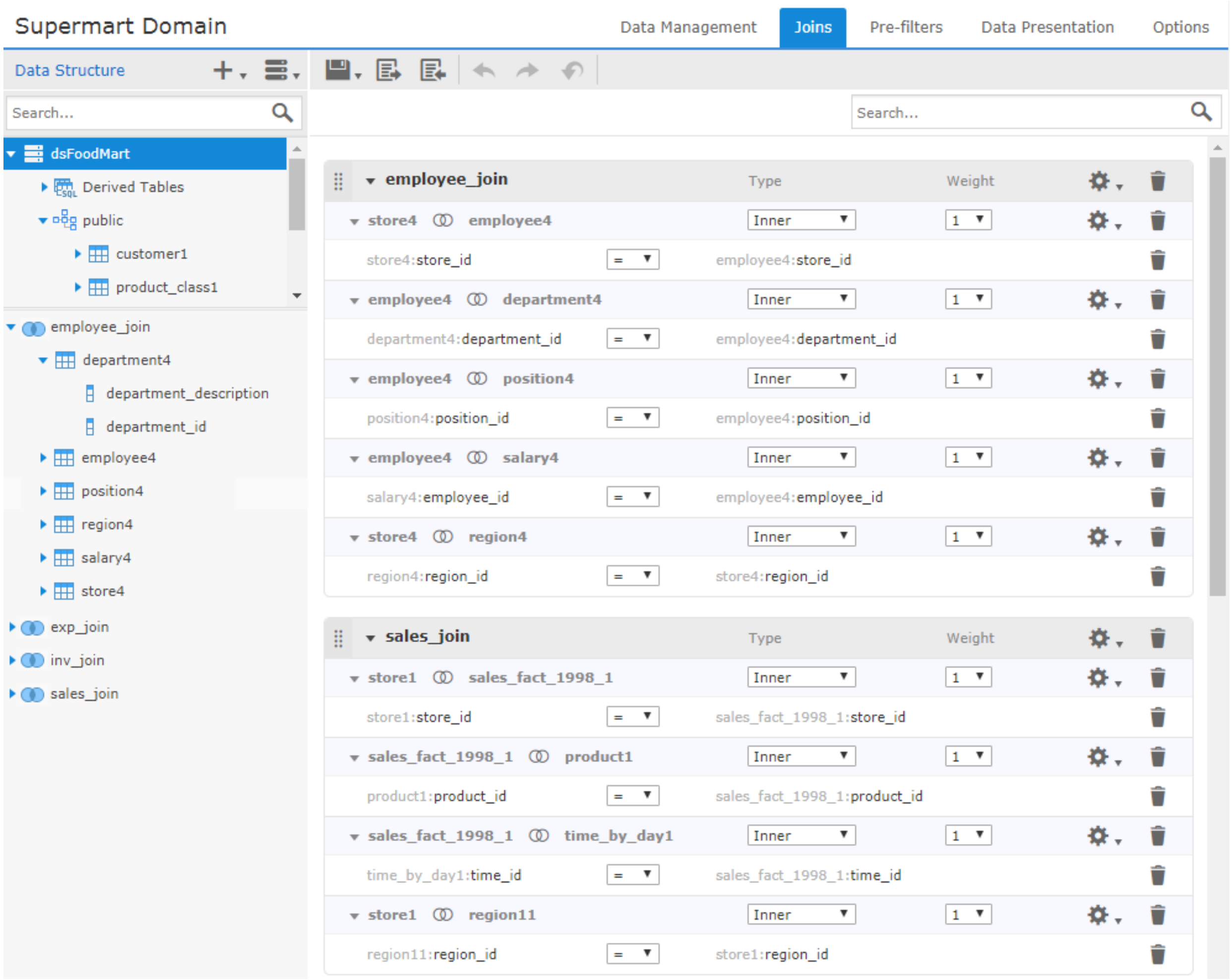
To work with tables, first select a schema in the Data Structure panel on the Data Management tab. The Manage Tables panel appears on the right and presents two lists:

- The **Available Tables** list displays the available tables in the selected schema.
- The **Selected Tables** list shows the tables in the selected schema that you have added to the Domain. Initially, this list is blank. When you add a table to this list, the table name also appears in the Data Structure panel.
- Selected tables appear in the Data Structure panel under the correct schema.
- To select tables, 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. To remove a table, double-click or drag it out of **Selected Tables**.

On the Data Management tab, you can select only entire tables. On the Joins, Pre-filters, and Data Presentation tabs, you can make table- and column-level selections.

7.3.5 The Joins Tab

The Joins tab is where you create associations between tables so that you can present them together in the same report. Multiple joins associate columns across many tables to create powerful data visualizations when used in reports. The number and complexity of joins in the Domain depends on your business needs.



employee_join			Type	Weight
store4 ⟷ employee4			Inner	1
store4:store_id	=	employee4:store_id		
employee4 ⟷ department4			Inner	1
department4:department_id	=	employee4:department_id		
employee4 ⟷ position4			Inner	1
position4:position_id	=	employee4:position_id		
employee4 ⟷ salary4			Inner	1
salary4:employee_id	=	employee4:employee_id		
store4 ⟷ region4			Inner	1
region4:region_id	=	store4:region_id		

sales_join			Type	Weight
store1 ⟷ sales_fact_1998_1			Inner	1
store1:store_id	=	sales_fact_1998_1:store_id		
sales_fact_1998_1 ⟷ product1			Inner	1
product1:product_id	=	sales_fact_1998_1:product_id		
sales_fact_1998_1 ⟷ time_by_day1			Inner	1
time_by_day1:time_id	=	sales_fact_1998_1:time_id		
store1 ⟷ region11			Inner	1
region11:region_id	=	store1:region_id		

Figure 7-9 Layout of the Joins Tab

Joins in the Domain Designer are grouped in join trees. A join tree is a group of tables that are all connected directly or indirectly through joins. Different join trees have no connections between them. The Joins tab shows tables organized by join tree in the Data Structure panel, and joins organized by join tree in the Design panel. You create joins by dragging fields from the Data Structure panel to the Design panel.

7.3.5.1 The Data Structure Panel

On the Joins tab, the Data Structure panel displays the tables in your Domains, organized into join trees. In this view, a single join tree contains a group of tables that are all connected directly or indirectly through joins. Unjoined tables and columns appear at the top (underneath the data source node), and joined tables and their columns appear at the bottom. The following actions are available:

- Expand the data source node to see the unjoined tables and columns in your Domain.
- Drag the divider between the data source tables and the join trees.
- Expand a join tree to see the tables it contains.
- Expand a table or derived table to see the columns it contains.
- Drag a column to the Joins design panel to use it in a join.

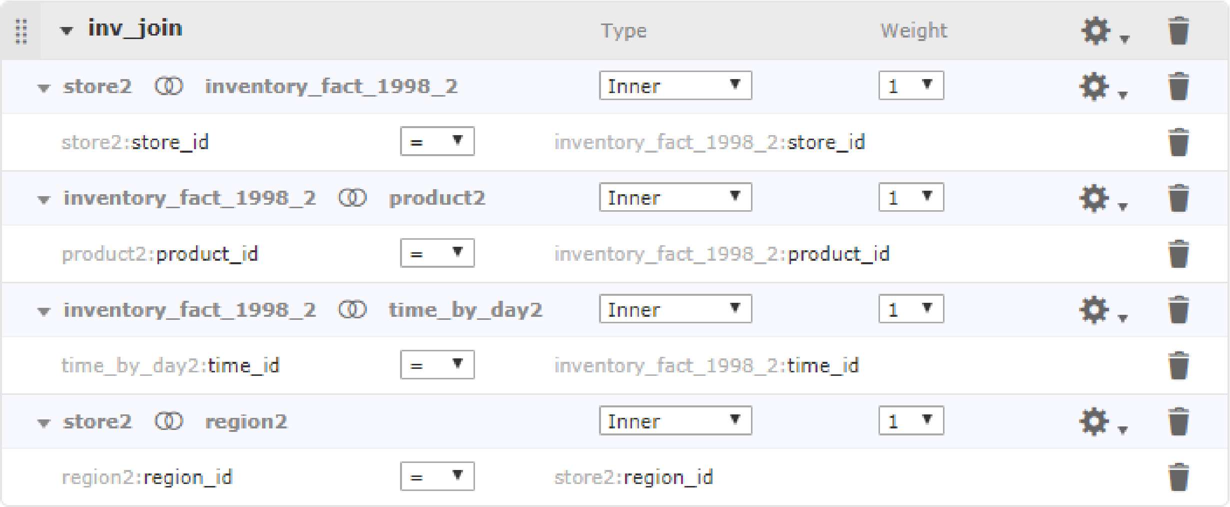
- Right-click the data source node to display a context menu with the following choices:
  - **Generate Joins** – Automatically creates joins based on foreign keys. If the database does not have foreign keys, you see an error and no joins are generated.
- Right-click a join tree to display a context menu with the following choices:
  - **Create Calculated Field** – Opens the Calculated Field dialog box to create a calculated field using columns from the tables in the join. See **7.4.2, "Calculated Fields," on page 242** for more information.
- Right-click a table to display a context-sensitive menu that includes some of the following:
  - **Copy Table** – Copies the selected table and gives it a name with sequential numbering.
  - **Rename Table** – 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.
  - **Remove Table** – (Copy of table only.) Removes the selected table copy from the Domain. To remove the original table from the Domain, use the Data Management tab.
  - **Edit Derived Table…** – (Derived table only.) Opens the Edit Derived Table dialog, where you can change the table name, query, and selected fields. If you change the name, the new name becomes the ID of the table throughout the Domain and is updated everywhere it appears in the Domain Designer.
  - **Remove Derived Table…** – (Derived table only.) Removes the selected derived table from the Domain.
  - **Always Include Table** – (Table in a join tree only; not available for unjoined tables.) Include this table in all Ad Hoc views that use this join tree, even if the table is not explicitly added by the Ad Hoc user. Enable **Use Minimum Joins** when you select this option.
  - **Create Calculated Field** – Open the Calculated Field dialog box to create a calculated field using the columns in the table. See **7.4.2, "Calculated Fields," on page 242** for more information.
- Hover over a table or table copy to see the name of the original table in the data source.

7.3.5.2 The Joins Design Panel

The Joins design panel shows the join trees and joins you have added to your Domain.

7.3.5.2.1 Working With Join Trees

In the Joins panel, the join trees in your Domain appear as containers with individual joins inside them. The selected join options are displayed for each join. See **7.3.5.3, "Understanding Joins and Join Trees," on page 222** for more information.



inv_join			Type	Weight
store2 ⟗ inventory_fact_1998_2			Inner ▼	1 ▼
store2:store_id	= ▼	inventory_fact_1998_2:store_id		
inventory_fact_1998_2 ⟗ product2			Inner ▼	1 ▼
product2:product_id	= ▼	inventory_fact_1998_2:product_id		
inventory_fact_1998_2 ⟗ time_by_day2			Inner ▼	1 ▼
time_by_day2:time_id	= ▼	inventory_fact_1998_2:time_id		
store2 ⟗ region2			Inner ▼	1 ▼
region2:region_id	= ▼	store2:region_id		

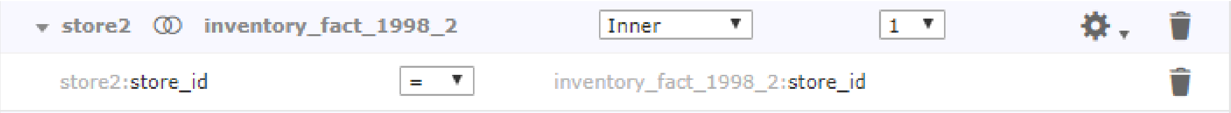
Figure 7-10 A join tree with multiple joins

The join tree menu bar at the top of each join tree lets you perform the following actions:

- ⠿ – Drag to move the join tree container to a new location and change the order of the join trees. A thick blue line appears in locations where you can place the join tree.
- ▸ ▾ – Use the arrows to expand and collapse the join tree.
- **Type**, **Weight** – Read-only headings that refer to properties of the individual joins in the join tree.
- ⚙ ▾ – Join tree settings menu. Click to perform the following actions:
  - **Rename Join Tree** – Displays the Rename Join Tree dialog, where you can give your join a more useful name.
  - **Use minimum path joins** – Helps you choose how joins are chosen in Ad Hoc views. Deselected by default; we recommend that you select this. See **7.3.5.4, "Prioritizing Joins," on page 226** for more information.
  - **Use all joins** – Forces an Ad Hoc view to include all joins in the join tree. Included for backwards compatibility. We recommend you do not select this option. See **7.3.5.4, "Prioritizing Joins," on page 226** for more information.
- 🗑 – Click to remove the join tree and all the joins it contains. If there are other Domain items that depend on the join tree, such as pre-filters or sets based on the join, you are prompted to remove them. The tables and columns remain in your Domain.

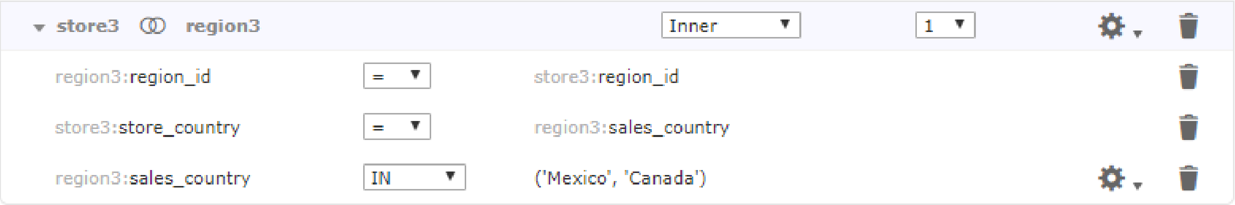
7.3.5.2.2 Working with Joins

Individual joins appear in their join tree container. A join includes a title bar and entries for the joined columns. If you create multiple joins between the same two tables, the new join component is added to the existing join. See **7.3.5.3.5, "Composite Joins," on page 224** for more information.



store2 ⟗ inventory_fact_1998_2			Inner ▼	1 ▼
store2:store_id	= ▼	inventory_fact_1998_2:store_id		

Figure 7-11 A Single Join Between Two Tables



▾ store3	⚭ region3	Inner ▾	1 ▾	⚙ ▾	🗑
region3:region_id	= ▾	store3:region_id			🗑
store3:store_country	= ▾	region3:sales_country			🗑
region3:sales_country	IN ▾	('Mexico', 'Canada')		⚙ ▾	🗑

Figure 7-12 Multiple Joins For the Same Two Tables

The join title bar at the top of each join includes the following:

- ▸ ▾ – Use these arrows to expand and collapse the join.
- Left column name (read-only) – Displays the table used in the left side of the join.
- **Type** drop-down – Sets the type of your join: **Inner**, **Left Outer**, **Right Outer**, or **Full Outer**. See 7.3.5.3.3, “Join Type,” on page 223.

  Full Outer is not supported for MySQL databases.

- Right column name (read-only) – Displays the table used in the right side of the join.
- **Weight** drop-down (default = 1) – Influences which joins are used in an Ad Hoc view. Higher weights indicate less desirable joins. To change this value, you must enable **Use minimum path joins** for this join tree. See 7.3.5.4.4, “Join Weights,” on page 228 for more information.
- ⚙ ▾ menu – Selecting **Create Custom Joins...** opens the **New Custom Join** dialog, which allows you to set a constant condition on a single column. See 7.3.5.3.6, “Custom Joins,” on page 224 for more information.
- 🗑 – Deletes the join. If there are other Domain items that depend on the join, s pre-filters or sets based on the join, you are prompted to remove them. The tables and columns remain in your Domain.

The entry for each component displays the following:

- Left column name (read-only) – Displays the table and column for the left side of the join.
- Join comparison menu – Drop-down that lets you select a comparison operator, based on the type of the join. Operators include =, ≠, >, <, >=, and <=.

  Comparison operators other than = (such as ≠, >, <, >=, <=) generate a large amount of rows (similar to a Cartesian product) when used on their own. Best practice is to use them as constraints in conjunction with a = join on the same pair of tables. See 7.3.5.3.4, “Comparison Operators,” on page 223 for more information.

- Right column name (read-only) – Displays the table and column for the right side of the join. For a custom join, displays the constant.
- ⚙ ▾ menu (custom joins only) – Lets you select **Edit Custom Join...**, which opens the Edit Custom Join dialog box.
- 🗑 – Deletes the join component. If there is only one component in the join, deletes the entire join.

7.3.5.2.3 Creating a Join

1. In the Data Structure panel, find the column you want to use for the left-hand side of your join and drag it into the Design panel in the location you want:
   - To create a new join on an empty canvas, drag the column anywhere on the Design panel. A new join is created, along with the join tree that contains it.
   - To add a join to an existing join tree, drag the column to the join tree. The target join tree is outlined with a dashed line to show it is selected.
   - To add a new join tree to existing join trees, drag the column to a blank space above, between, or below the existing join trees. The target area is highlighted with a thick blue line.

   A join is added to the Design panel. The column you dragged is added on the left, and the right join displays **Drag a field here**.

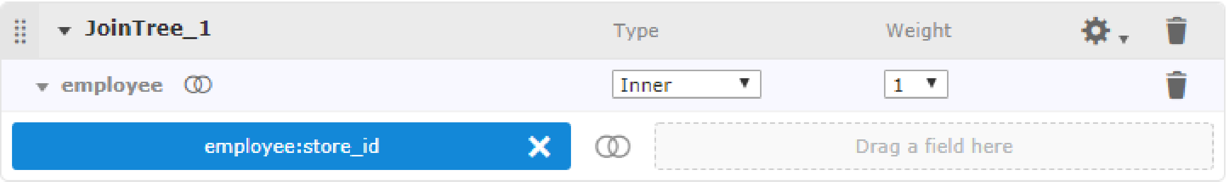


Figure 7-13 Creating a New Join

2. Select the join type (**Inner**, **Left Outer**, **Right Outer**, **Full Outer**) you want from the **Type** menu. This can be changed later. See 7.3.5.3.3, "Join Type," on page 223 for more information.
3. Drag the column for the right table to the **Drag a field here** box. Make sure to choose a column that's compatible with the column you already chose. The column is added to the join.

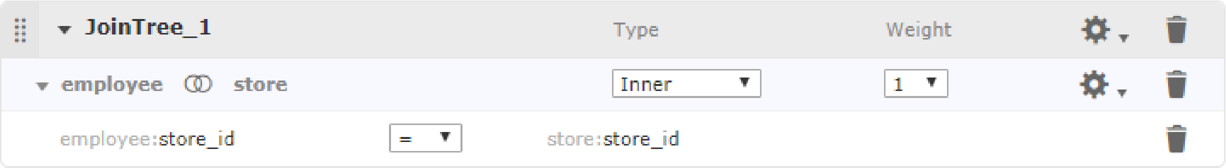


Figure 7-14 Adding the Right Table to a Join

4. If you want to change the comparison operator for the join, select a different operator from the list. The list of available operators depends on the field type.

   Comparison operators other than = (such as ≠, >, <, >=, <=) generate a large amount of rows (similar to a Cartesian product) when used on their own. Best practice is to use them in conjunction with a = join on the same pair of tables. See 7.3.5.3.4, "Comparison Operators," on page 223 for more information.

5. If you want to add a weight to the join, first make sure that **Use minimum path joins** is selected in the join tree settings. Then select the weight you want from the **Weight** menu. Remember that a higher weight means a less desirable join. See 7.3.5.4.4, "Join Weights," on page 228 for more information.

7.3.5.3 Understanding Joins and Join Trees

Joins are associations you make between tables so that their rows can 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 and complexity of joins in the Domain depends on your business needs.

7.3.5.3.1 Understanding Join Trees

A group of tables that are all connected directly or indirectly through joins is called a join tree. A Domain may contain several join trees, but when a user creates an Ad Hoc view from a Domain, they can only select one of them to be available in the view. Different join trees have no connections between them.

Join trees appear as top-level nodes in the Data Structure panel on the **Joins**, **Pre-Filters** and **Data Presentation** tabs. Tables that are not joined appear as children of the data source node at the top of the Data Structure panel.

The options on the Join Tree menu help you prioritize which join paths are chosen when multiple options are possible in an Ad Hoc view. These options are primarily used to avoid circular joins. See **7.3.5.4, "Prioritizing Joins," on page 226** for more information.

7.3.5.3.2 Understanding Join Options

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, the `store_id` column in the `store` table can be joined with the `store_id` column in the `employee` table.

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.

The XML design file supports additional join features not supported in the Domain Designer, including:

- NOT or OR operator between joins inside a composite join.
- Joins within the same table.

If you have a Domain design file in XML format that uses these features and you open it in the Domain Designer, these joins will be displayed as read-only joins. See **7.3.5.3.7, "Read-Only Joins," on page 226** for more information.

7.3.5.3.3 Join Type

The Domain Designer supports the four most common join types:

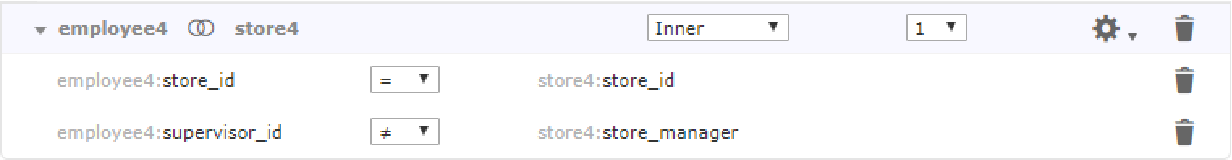
- **Inner** – The result contains only rows where the values in the chosen columns are equal.
- **Left Outer** – The result contains all the rows of the left table, paired with a row of the right table when the values in the chosen columns are equal or contain blanks.
- **Right Outer** – The result contains 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.
- **Full Outer** – The result contains all rows from both tables, paired when the joined columns are equal, and filled with blanks otherwise. Not available in MySQL.

7.3.5.3.4 Comparison Operators

The Domain Designer supports the following comparison operators between fields. You can't compare fields of different data types:

=, ≠, >, <, >=, <= .

Comparison operators other than = often generate a large amount of rows (similar to a Cartesian product) when used on their own. We recommend that you use them in a composite join in conjunction with an = join. See **7.3.5.3.5, "Composite Joins," on page 224** for more information.



employee4 ⟗ store4 — Inner — 1

employee4:store_id = store4:store_id

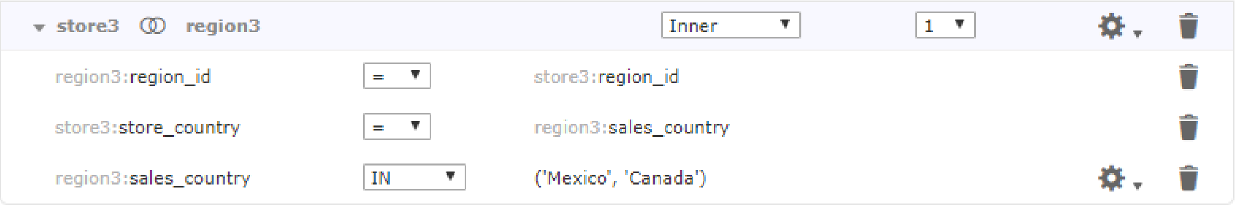
employee4:supervisor_id ≠ store4:store_manager

Figure 7-15 Example of a Join that Uses an Inequality

7.3.5.3.5 Composite Joins

Composite joins implement multiple join conditions for the same pair of tables. You can create composite joins in the following ways:

- Add a join to a join tree that already contains a join between those tables.
- Select **Create Custom Join...** from the ⚙▾ menu of the join.



store3 ⟗ region3 — Inner — 1

region3:region_id = store3:region_id

store3:store_country = region3:sales_country

region3:sales_country IN ('Mexico', 'Canada')

Figure 7-16 Example of Composite Join

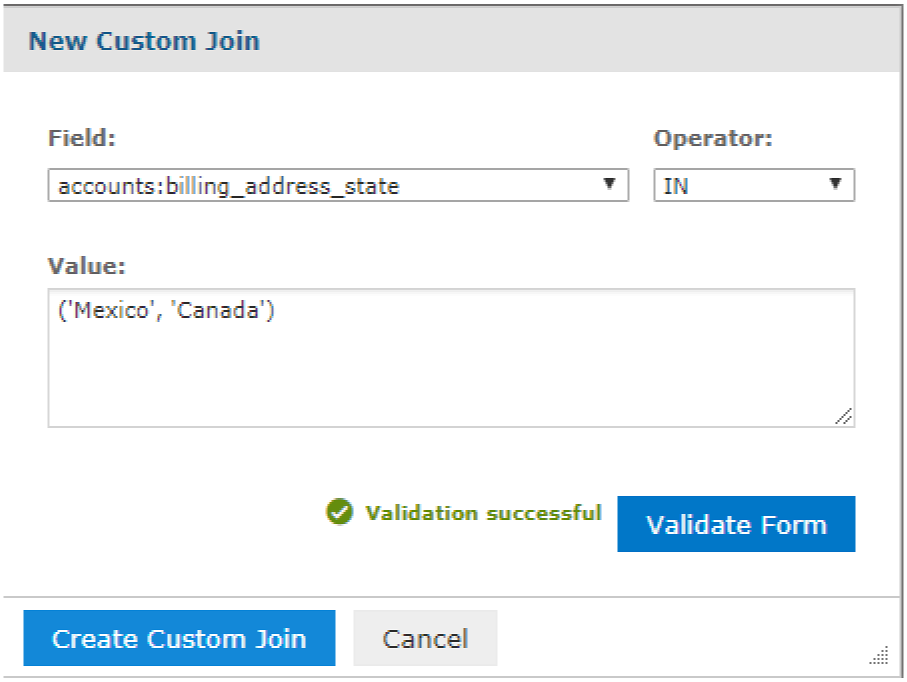
The join options you select on the menu bar for the join, such as join type and weight, apply to the entire composite join. In addition, when the Domain is used in an Ad Hoc view or report, the composite join is treated as a single join with multiple components. See **7.3.5.4, "Prioritizing Joins," on page 226** for more information. When the Domain is exported as XML, composite joins are represented as a single join expression with multiple components.

7.3.5.3.6 Custom Joins

You can add custom joins to an existing join. Custom joins let you place constant conditions on a single column from the joined tables.

To add a custom join to an existing join:

1. Click ⚙▾ at the upper right of the join and select **Create Custom Join...**. The New Custom Join dialog appears.



New Custom Join

Field: accounts:billing_address_state

Operator: IN

Value: ('Mexico', 'Canada')

Validation successful

Figure 7-17 New Custom Join dialog

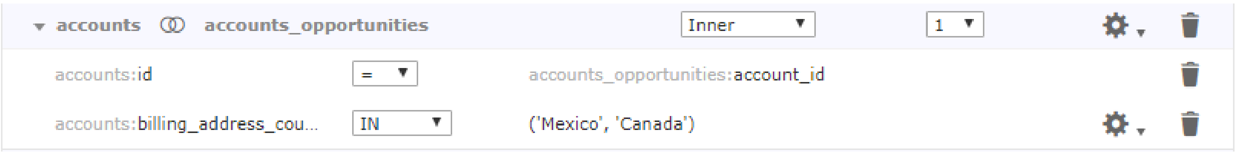
2. Select a column from the Field list. You can select a column from either table in the join.
3. Select an operator. The available operators depend on the column type.
4. Enter the range, set, or value you want for the field, based on the operator you chose.
   - = or ≠ (available for all column types)

     >, <, >=, or <= (available for numeric and date columns only)

     Enter a constant value or an attribute that takes a single value. Strings are enclosed in single quotes. For example:
     - `5000`
     - `'Mexico'`
     - `attribute('CountryAttribute')`

   For more information about using attributes in Domains, see **7.5, “Using Attributes in the Domain Designer,” on page 245**.

   - IN or NOT IN – Enter one of the following:
     - A set of strings or values, enclosed in parentheses and separated by commas. Strings are enclosed in single quotes. For example:
       - `(1,2,3,4,5)`
       - `('San Francisco','Portland', 'Seattle')`
       - `('true')`
     - A range of values, separated by a colon (numeric and date columns only). For example:
       - `(7000 : 8000)`
5. To verify that your syntax is correct, click **Validate Form**. Fix errors if necessary.
6. Click **Create Custom Join**. The join is added below the existing join as part of a composite join.



accounts ⊘ accounts_opportunities Inner 1

accounts:id = accounts_opportunities:account_id

accounts:billing_address_cou... IN ('Mexico', 'Canada')

Figure 7-18 A custom join in the design panel

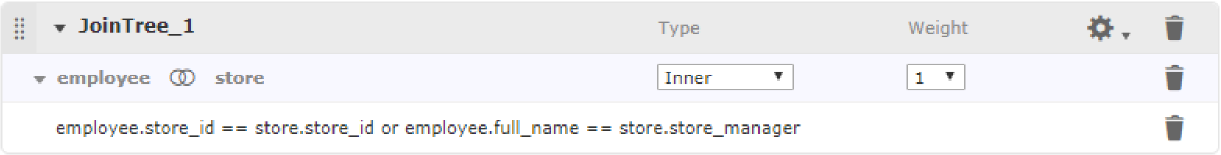
7. To edit or delete a custom join:
   - Click ⚙ ▾ and select **Edit Custom Join...** to open the Edit Custom Join dialog and modify the join.
   - Click 🗑 to delete a custom join.

7.3.5.3.7 Read-Only Joins

The design file format (XML) supports additional join features not supported in the Domain Designer, including:

- NOT or OR operator in a single join expression.
- Joins between different columns in the same table.

If you have a Domain design file in XML that uses these features and you open it in the Domain Designer, these joins are displayed as read-only joins. The join expression is shown as a string. You can still set the join type and weight.



JoinTree_1 Type Weight

employee ⊘ store Inner 1

employee.store_id == store.store_id or employee.full_name == store.store_manager

Figure 7-19 Example of Read-Only Join

To create or modify joins outside of the Domain Designer, you must export the Domain in XML format and modify it in a text-editor. See **7.8, "Importing and Exporting Domain Design Files," on page 253** and **Chapter 8, "Working with Domain Files," on page 257** for more information.

7.3.5.4 Prioritizing Joins

The Domain Designer and Domain Design file do not impose any limits on the number of joins you can specify in a join tree. 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 (also known as loops). 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. However, if your design requires circular joins, JasperReports Server includes features that help you influence how joins are prioritized when they are used in an Ad Hoc view, Topic, or report.

The options on the Join Tree menu, along with specific table and join options, give you influence over which join paths are chosen in an Ad Hoc view when multiple options are possible.

7.3.5.4.1 Best Practices for Join Trees

Wherever possible, follow these practices to avoid loops or to minimize their impact on Ad Hoc views:

- For a join tree with N tables, use N-1 joins. A composite join is considered a single join. See **7.3.5.3.5, "Composite Joins," on page 224** for more information.
- Create copies of tables you need to use more than once in the join tree. You can copy a table by right-clicking it in the Data Structure panel on the Joins tab and selecting **Copy Table** from the context menu.
- Enable **Minimum Path Joins** for each join tree. (This option is not the default.)
- For better join performance, prioritize joins that involve indexed columns. To do this, assign low join weights to the preferred joins and high join weights to less desirable joins.

7.3.5.4.2 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 more than N-1 joins among N tables, your design will include circular joins. The following example shows a circular join where table A is joined to table B and table B is joined to table C which is in turn joined to table A.

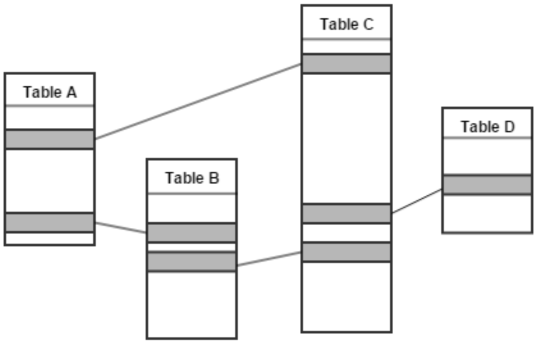


Figure 7-20 Circular Joins in a Domain

7.3.5.4.3 Join Tree Options

Often, an Ad Hoc view created from a join tree in your Domain does not need to use all the tables and joins in the join tree. The ⚙ ▾ menu on the join tree includes options you can use to change how Ad Hoc views select joins. Only one of these options can be enabled at a time.

Suppose you create an Ad Hoc view using two tables from a Domain with a loop. JasperReports Server attempts to avoid circular joins in the SQL by choosing a path between the tables. However, by default, you cannot be sure which join path the Ad Hoc view will use. If you are using tables A and B, you do not know whether the server will use the direct join from A–B or the indirect join A–C–B. The following options influence which joins are used by the Ad Hoc view:

- **Use minimum path joins** (unselected by default) – When this option is selected, an Ad Hoc view created from the join tree uses a minimum join length. When **Use minimum path joins** is selected, you can optionally assign weights to the joins in the join set. Higher weights indicate less desirable joins. For most situations, the best practice is to enable this option to avoid circular joins. Deselect it only if you need backwards compatibility with legacy Domains.
- **Use all joins** (unselected by default) – Available for backwards compatibility. When this attribute is selected, an Ad Hoc view created from the data island includes all the joins in the join tree. For most situations, the best practice is to disable this option (default).

7.3.5.4.4 Join Weights

When **Use minimum path joins** is enabled, each join in the join tree is given a default weight of 1. You can optionally change the weights of some or all of the joins in the join tree. Higher weights indicate lower-priority, less-desirable joins. When **Use minimum path joins** is used with weights, JasperReports Server calculates the total weight of a join path by summing the weights of its individual joins. It uses join path of the possible lowest total weight to generate the data for the Ad Hoc view, Topic, or report.

Increasing the weight increases the cost of a particular join and lowers its priority. 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.

In this situation, you can enable **Use minimum path joins**. This automatically assigns 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. You create an aliases, or copy, by right-clicking a table in the Data Structure panel on the Joins tab and selecting **Copy Table** from the context menu.

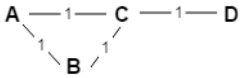


Figure 7-21 Circular Join with Default Weights

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.

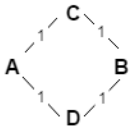


Figure 7-22 Circular Joins with Equal Weights Between A and B

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 joins in your join tree. Adding a weight increases the cost of a particular join, and lowers its priority. 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.

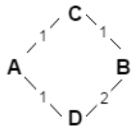


Figure 7-23 Circular Joins with Weights

7.3.5.4.5 Always Include Table

When **Use minimum path joins** is selected, you can also **Always Include Table** to specify that any Ad Hoc view created from this join tree must include the indicated table in its join path. This option takes effect even when no columns from the table are used in the Ad Hoc view. You can enable this option on multiple tables in the same join tree.

Technically, you can select this property whether or not you have set **Use minimum path joins**. However, it is most useful when **Use minimum path joins** is selected.

7.3.6 The Pre-filters Tab

A pre-filter on one or more columns restricts the data that the Domain retrieves from the data source. Pre-filtering irrelevant data reduces the size of query results and processing time. In addition, building often-used pre-filters into the Domain design avoids the need for each report designer to define filters independently and also reduces the chance for errors.

You can define a pre-filter on a column you don't plan to expose in the Domain. The pre-filter remains active and restricts the data that appears to report users. For example, you can pre-filter data to select a single country, in which case it doesn't make sense for the column to appear in the Domain's presentation data. However, you should clearly document such data restrictions in the description of the Domain, so users understand what data is accessible through the Domain.

Ad Hoc views and reports based on the Domain can define their own filters. When you use a pre-filter, the data is never retrieved from the database. When you filter on the Ad Hoc level, the data is retrieved, but may not be displayed to the user.

You have two options for defining parameters for a pre-filter:

- Define the parameters statically, so they are the same for every user. For example, you could create a static filter that filters out data for every country except the USA.
- Have the server derive the parameter of the filter at run time based on attributes you provide. For example, you can create a filter that only shows data for the logged-in user's country, based on a Country attribute set in your users' profiles. If the specified attribute has not been defined anywhere on the server, the filter will fail. If the attributes is NULL, the filter will not retrieve any data.

You can also filter the data using calculated fields. The primary difference between pre-filters and calculated fields is that calculated fields are visible to report designers and users, and can be used as the basis for additional filters in the Ad Hoc Designer. See **7.4.2, “Calculated Fields,” on page 242**

For more information on attributes, see **7.5, “Using Attributes in the Domain Designer,” on page 245**.

To define a pre-filter:

1. Go to the Pre-filters tab in your Domain.

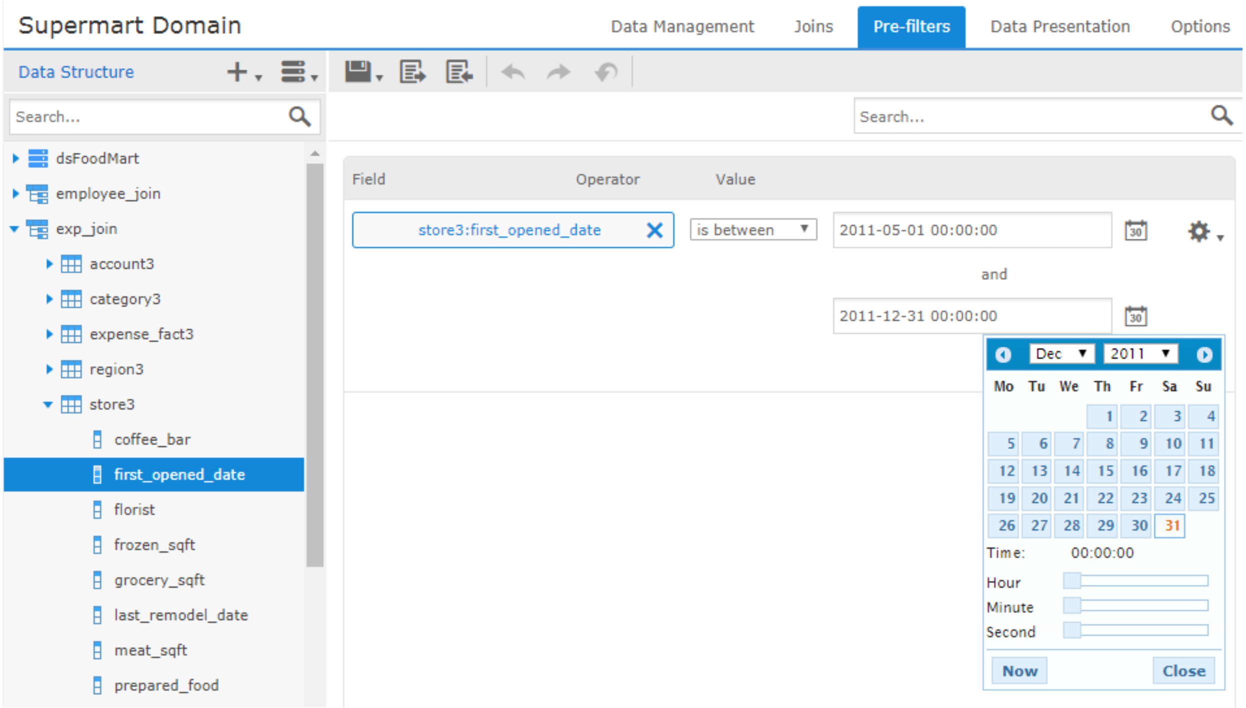
2. Drag a column from the Data Structure panel to the Pre-filters panel.

   The column appears in the Pre-filters design panel with a list of comparison operators you can apply to that column.
3. Choose the comparison operator from the drop-down.

   In the Pre-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.
4. Select **Field to Value Comparison** or **Field to Field Comparison** from the ⚙▾ menu.

   If you select **Field to Field Comparison**, the second entry box changes to **Drag a field here**.
5. Select or enter a value for your filter. For a filter that compares fields, drag field(s) of the same type to the Pre-filters panel.

   The format of the filter value changes depending on your previous selections. 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:



Supermart Domain

Field: store3:first_opened_date

Operator: is between

Value: 2011-05-01 00:00:00 and 2011-12-31 00:00:00

Figure 7-24 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.

If you want to use an attribute for a parameter, enter the attribute in the field using the syntax `attribute ('<attrName>')`. For example, if you want to use an attribute called Cities in the filter, enter `attribute`

`('Cities')` into the field. If you want to specify that this is a user attribute, and not an organizational or server one, enter `attribute('Cities', 'user')`.

6. Click **OK** to define the filter.

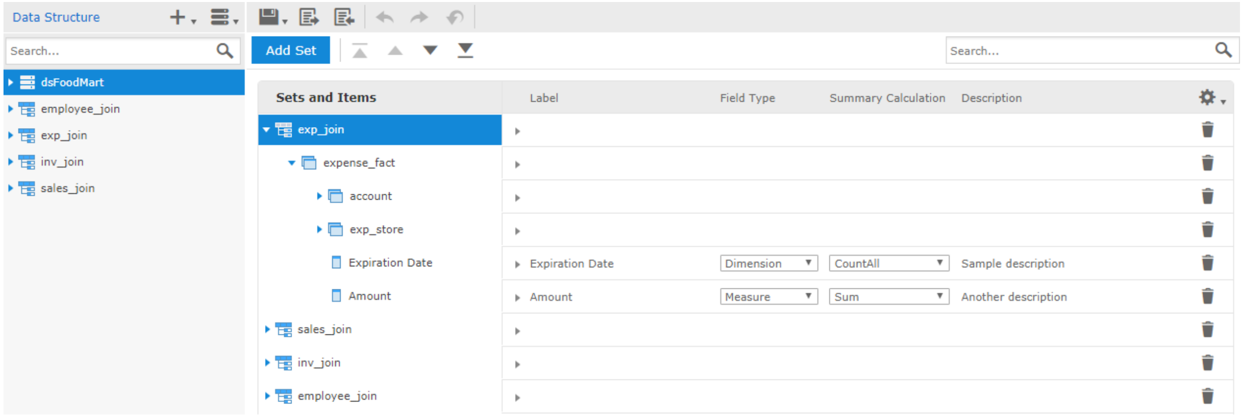
   The Pre-filters panel shows all the filters you have defined. The overall filter applied to the data is the logical AND of all conditions you defined.

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**.

Pre-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 create more complex filters in the Domain Design file, but these filters are read-only and are not editable in the Domain Designer.

7.3.7 The Data Presentation Tab

The Data Presentation tab is where you specify the columns and calculated fields you want visible to users. You can also create user-friendly names and descriptions and set other column display properties. Typically, you expose only columns that are useful in building or filtering reports. Columns that you don't display are still part of the Domain and can affect the data retrieved.



Sets and Items	Label	Field Type	Summary Calculation	Description
exp_join				
expense_fact				
account				
exp_store				
Expiration Date	Expiration Date	Dimension	CountAll	Sample description
Amount	Amount	Measure	Sum	Another description
sales_join				
inv_join				
employee_join				

Figure 7-25 The Data Presentation Tab

The Data Presentation tab contains the following:

- **Data Structure** panel – Displays available tables and fields for this Domain, including table copies, derived tables, and calculated fields. Joined tables appear as join trees. Dragging a join tree, table, or column from Data Structure to Sets and Items does not remove it from the Data Structure panel. This reflects the fact that you can add a resource to more than one set.

You cannot delete tables from the Domain on the Data Presentation tab. You must do this on the Data Management or Joins tab.

- **Sets and Items** list – Lists resources that will appear to report creators who use the Domain.
- **Properties** pane – Displays the properties of the associated set or item. This area can be expanded or collapsed.

To specify which columns and calculated fields are exposed to users of the Domain, drag them from the **Data Structure** panel to the **Sets and Items** panel. The following icons show the different resources:

- – A data island. Corresponds to a join tree or unjoined table, but does not necessarily have the same structure. For example, a data island does not need to contain all the tables or columns from its join tree. In addition, it can contain sets that do not correspond to any table, and it can contain the same field or table several times across different sets. However, a data island can only contain resources from a single join tree or unjoined table, and a join tree or unjoined table can correspond to at most one data island. Can contain sets and items.

When a user creates an Ad Hoc view, they can only choose sets from the same data island.

- – A set. A group of resources; can contain items and other sets. Sets can be created by dragging a table, in which case they can only contain items from that table; or they can be created by clicking **Add Set**, in which case they can contain items from different tables in the same join tree.
- – An item. A column or calculated field that you want to appear in the Domain.

7.3.7.1 The Data Structure Panel

On the Data Presentation tab, the Data Structure panel displays the tables in your Domains organized into join trees. In this view, a single join tree contains a group of tables that are all connected directly or indirectly through joins. Unjoined tables and columns appear at the top (underneath the data source node), and joined tables and their columns appear at the bottom. The following actions are available:

- Expand the data source node to see the unjoined tables and columns in your Domain.
- Expand a join tree to see the tables it contains.
- Expand a table or derived table to see the columns it contains.
- Use Ctrl-click or Command-click to select multiple items; use Shift-click to select a range of items.
- Drag selected items to the Sets and Items design panel to display it to the users.
- Hover over an item to see its ID.

In previous versions, you were not required to create data islands. Now, adding an item automatically adds it to the correct data island or creates a new data island if it does not yet exist. Opening and saving a Domain with the previous model updates it the current model. We recommend that you update older Domains.

7.3.7.2 The Sets and Items List

The Sets and Items list shows the resources you want the user to see, organized into data islands, sets, and items in a nested hierarchy. If there are sets and items at the same level, items appear first. The menu bar at the top of the Sets and Items list shows the following:

- **Add Set** – Creates a new subset of a selected set. Not available when no sets have been added.
- ⤒ ▲ ▼ ⤓ – Move the current selection as follows: to the top, up, down, or to the bottom. Data islands are moved relative to other data islands; items and sets are moved within their containing set or data island. You can also move a set or item by dragging. You can reorder sets and items by dragging them to another location in the same set; however, when you close and reopen a Domain, sets always appear below items. Data islands can be dragged above or below other data islands. You can move items and lower-level sets to any level in the same data island. If you drag a set or item between sets, it appears at the top of the set that it was moved to.

⚙ ▾ – Controls the display of sets and items and their properties:

- The following selections control the properties you view when a set or item is collapsed. When a resource is expanded, all available properties are visible.

Menu Item	Description	Visible Properties
Default Properties	Displays an overview of the resource.	Label, Field Type, Summary Calculation, Description
Identification Properties	Lets you view the ID and edit the label and description.	Label, ID, Description
Bundle Keys Properties	Lets you view and edit properties related to bundle keys for localization.	Label Key, Descriptions Key
Data Properties	Lets you select whether an item is a dimension or measure; lets you view and edit properties specific to measures.	Source, Field Type, Summary Calculation, Data Format

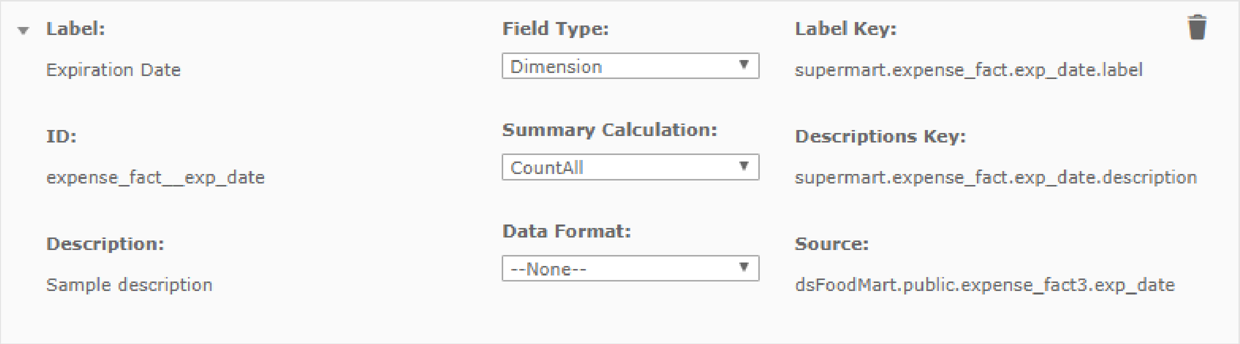
- The following selections let you expand and collapse the list of sets and items and their properties.
  - **Expand All Properties** – Expands all sets and items and displays expanded properties for all.
  - **Collapse All Properties** – Collapses all sets and items and displays collapsed properties for all.

  The properties shown depend on the selection you made.

The following selections are available for resources:

- ▾, ▸ – Expands or collapses a node in the Sets and Items list.
- ▾, ▸ – Expands or collapses properties. Click in any property to start editing it.
- 🗑 – Removes the selected resource.
- 🔍 – Searches items and sets and shows matches if any of the following contain the search string: label, ID, description, label key, descriptions key. Can be used to quickly find similar items and compare and edit their properties.

7.3.7.3 Properties



Label:
Expiration Date

ID:
expense_fact__exp_date

Description:
Sample description

Field Type:
Dimension

Summary Calculation:
CountAll

Data Format:
--None--

Label Key:
supermart.expense_fact.exp_date.label

Descriptions Key:
supermart.expense_fact.exp_date.description

Source:
dsFoodMart.public.expense_fact3.exp_date

Figure 7-26 Item Properties on the Data Presentation Tab

The Properties area of the Data Presentation tab lets you refine the Domain's appearance by renaming and providing descriptions for data islands, sets, and items. The following table describes the available properties.

Property	Appears On	Description
Label	Data Island, Set, Item	User-friendly name displayed in the Data Chooser and the Ad Hoc Editor.
ID	Data Island, 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. Presentation 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. You should not change IDs for a Domain that has already been used to create Ad Hoc views or Topics.
Description	Data Island, 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.
Field Type	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.
Summary Calculation	Item	Default summary calculation for 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.
Data Format	Item	Default numerical format (such as number of decimal places) for the item when used in a report. Numeric and dates only.
Label Key	Data Island, Set, Item	Internationalization key for the label property; locale bundles associate this key with the localized text for the label.
Descriptions Key	Data Island, Set, Item	Internationalization key for the description property; locale bundles associate this key with the localized text for the description.
Source	Item	References the data source, schema, table, and field associated with this item; the syntax is `datasource.schema.table.field`. Not editable.

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 must match the property names of strings in locale bundles. Keys may use only characters from the ISO Latin-1 set, digits, and underscores (_). If you create internationalization keys on the Data Presentation tab, you can download the keys as a template file for locale bundles by clicking **Download Template** on the Options tab.

7.3.7.4 Working With Sets and Items

Adding a set or item:

To move a resource to Sets and Items, drag it from the Data Structure panel and drop it in Sets and Items. A blue bar appears in the target location. If you can't see the blue bar, then you can't drop the resource in that location.

- To drop a resource in an existing data island or set, drag it anywhere in the data island or set. It appears at the bottom.
- To add a resource that does not belong to the current data islands, drag the resource to the top or bottom of the Sets and items list, or in the area between any two data islands. The resource is added, along with the data island that contains it. If you drag a join tree, it creates the corresponding data island.

To move most of the tables and columns in a join tree or a table to Sets and Items:

1. Drag-and-drop a join tree or table from the Data Structure panel to Sets and Items.
   - If you drag a join tree, it is added as a data island, all tables are added as sets, and columns are added as items within the sets.
   - If you drag a table, the table is added as a set, and all columns in the table are added as items with the set.
2. Remove any unwanted sets or items individually from Sets and Items using 🗑.

To move a few columns from a join tree or a table to Sets and Items:

1. Drag a single column from the Data Structure panel to Sets and Items.
   - If the column is part of a join tree, the join tree is added as a data island, and the column is added as an item directly under that data island.
   - If the column is part of an unjoined table, the table is added as a data island and the column is added as an item.
2. Continue to drag columns to the data island that was created in the first step.

   Columns are added directly to the data island.

To create an empty set:

You can create empty sets. If the parent is a data island or a set , the set can contain items from different tables.

1. Click on the set or data island you want as a parent.
2. Click **Add Set**. You must have a parent selected to use **Add Set**. This button is not active if you do not have any data islands.

   The new set appears at the bottom of the parent set you chose. Drag resources from the same join tree or unjoined table to add them to the set.

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.

You can also reposition an item or set by dragging.

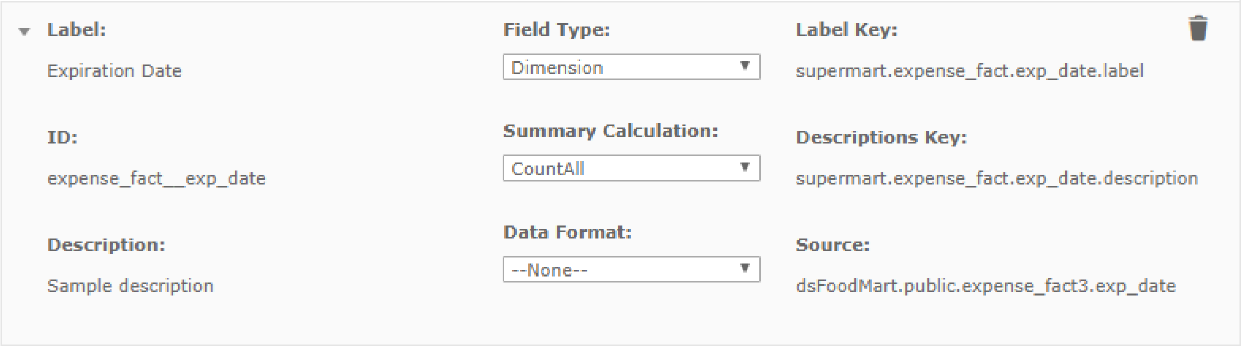
Individual items inside a set always appear above subsets. You can't move individual items below sets.

7.3.7.4.1 Understanding Data Islands

Elements in the Data Presentation design panel are grouped into data islands. Each data island comes from a join tree or unjoined table, but data islands differ from join trees in a number of important ways. Join trees represent the logical model for data in the Domain and data islands represent the business model presented to the users.

Join Tree (or Unjoined Table)	Data Island
Each join tree is unique and does not connect to any other join tree.	Each data island is unique and does not connect to any other data island.
Structure is organized in tables and fields.	Structure is organized in sets and items.
Directly reflects the structure of the tables and columns in the database, along with derived tables, calculated fields, and table and column copies (aliases).	Lets you group elements in sets to create a structure that better reflects your business model. You can exclude tables and columns you do not want to expose to the user, without removing them from the original join tree.
Table and column names come from database or internal Domain Designer naming (for calculated fields, derived tables, and table copies).	Sets and items can be renamed. Names that appear to users can be defined in localization bundles based on strings assigned in the data island.

7.3.7.5 Properties



Label: Expiration Date	Field Type: Dimension	Label Key: supermart.expense_fact.exp_date.label
ID: expense_fact__exp_date	Summary Calculation: CountAll	Descriptions Key: supermart.expense_fact.exp_date.description
Description: Sample description	Data Format: --None--	Source: dsFoodMart.public.expense_fact3.exp_date

Figure 7-27 Item Properties on the Data Presentation Tab

The Properties area of the Data Presentation tab lets you refine the Domain's appearance by renaming and providing descriptions for sets, subsets, and items. The following table describes the available properties:

Property	Appears On	Description
Label	Set, Item	User-friendly name displayed in the Data Chooser and the Ad Hoc Editor.
ID	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. You should not change IDs for a Domain that has been used to create Ad Hoc views or Topics.
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.
Field Type	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.
Summary Calculation	Item	Default summary calculation 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.
Data Format	Item	Default numerical format (such as number of decimal places) for the item when used in a report. Numeric and dates only.
Label Key	Set, Item	Internationalization key for the label property; locale bundles associate this key with the localized text for the label.
Descriptions Key	Set, Item	Internationalization key for the description property; locale bundles associate this key with the localized text for the description.
Source	Item	References the data source, schema, table, and field associated with this item; the syntax is `datasource.schema.table.field`. Not editable.

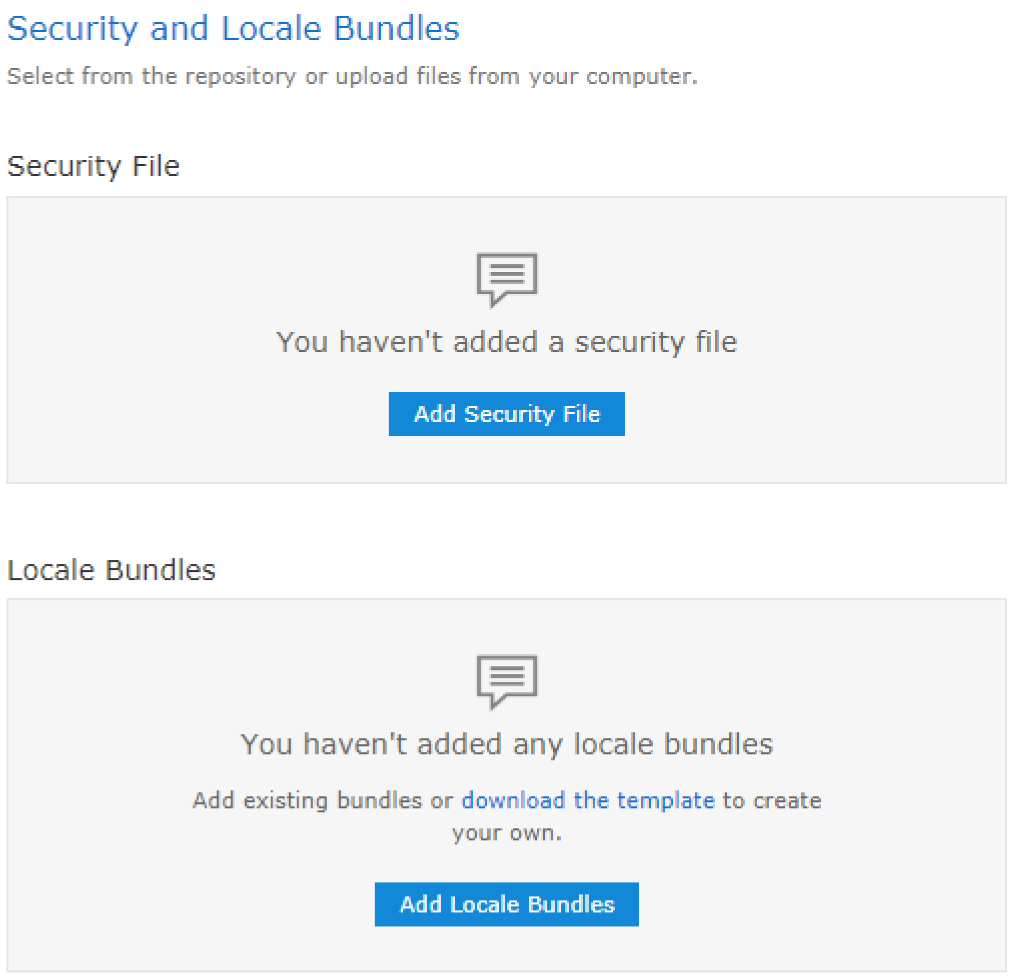
Labels and descriptions are visible to users of the Domain. Descriptions of sets and items appear as tooltips in the Ad Hoc Editor to help report creators understand their purpose.

The internationalization keys must match the property names of strings in locale bundles. Keys may use only characters from the ISO Latin-1 set, digits, and underscores (_).

7.3.8 The Options Tab

The Options tab lists resource files you have attached to the Domain and lets you add or remove resource files. You can attach the following resource files to your Domain:

- 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.



Security and Locale Bundles

Select from the repository or upload files from your computer.

Security File

You haven't added a security file

Locale Bundles

You haven't added any locale bundles

Add existing bundles or download the template to create your own.

Figure 7-28 Options Tab When No Resources Are Attached

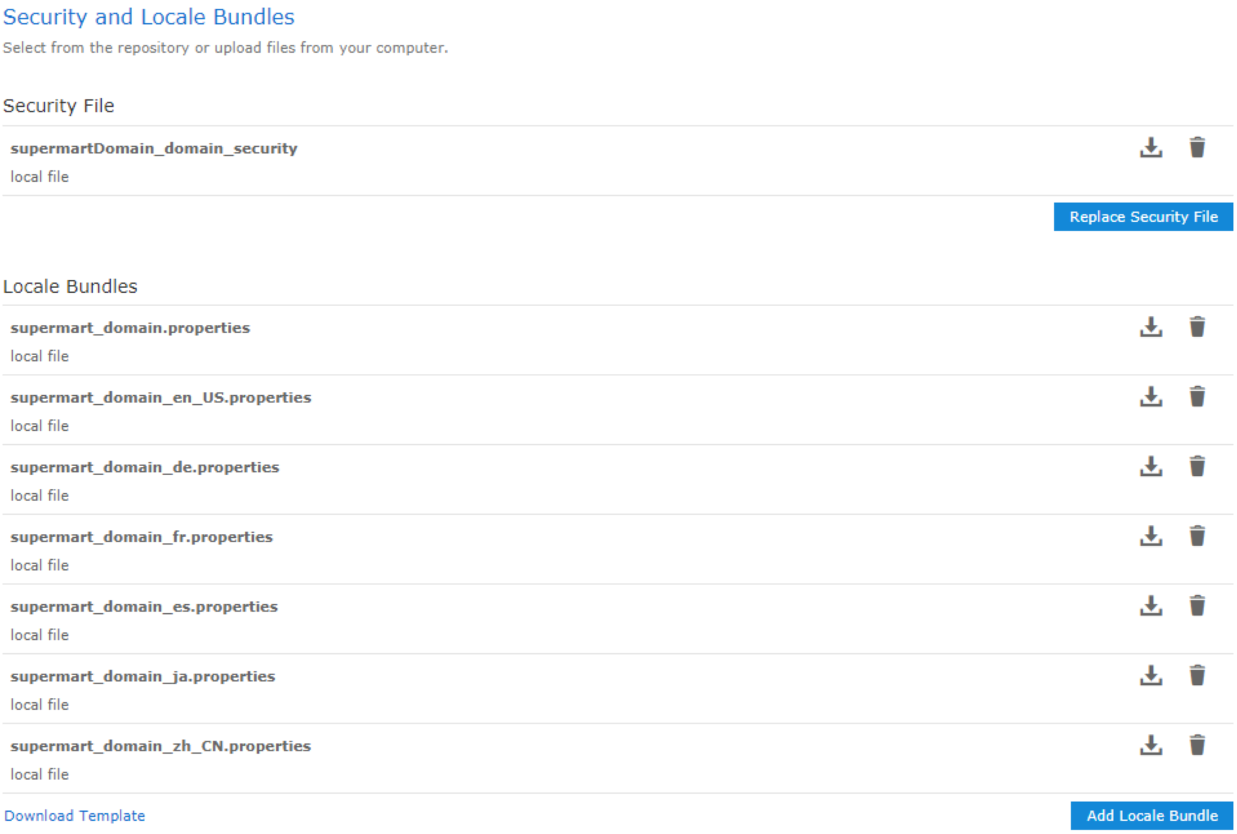
The security file and locale bundles must be created separately in the correct text-based format. Files can be uploaded locally and attached to the Domain, or you can add a file directly to the repository and reference it from multiple Domains:

- A Domain can have only one security file but any number of locale bundles.
- Locale bundles are uploaded as type **File > Resource Bundle** and Domain security files are type **File > Access Grant Schema**.
- Files in the repository are referenced by the Domain; local files are uploaded directly into the Domain:
  - Local files are attached to the current Domain and can't be accessed by other Domains. This ensures that users of other Domains won't modify your file to meet their needs.
  - Files in the repository can be accessed by multiple Domains. This allows a security file or locale bundle to be used by multiple Domains.
    - 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 the file with a different or modified file.
    - Replacing a shared security file can affect the security of Domains that use it. If your security file is replaced or modified, your security design may fail without any visible error.

- Care must be taken to ensure that the Label and Descriptions Keys are consistent across all Domains that share a locale bundle.

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.

This section only describes how to attach and replace Domain resources. For information about uploading files to the repository, see the *JasperReports Server Administrator Guide*. For information about the syntax of resource files, see **"The Domain Security File" on page 308** and **"Localizing Domains" on page 302**.



Security and Locale Bundles

Select from the repository or upload files from your computer.

Security File

supermartDomain_domain_security
local file

Replace Security File

Locale Bundles

supermart_domain.properties
local file

supermart_domain_en_US.properties
local file

supermart_domain_de.properties
local file

supermart_domain_fr.properties
local file

supermart_domain_es.properties
local file

supermart_domain_ja.properties
local file

supermart_domain_zh_CN.properties
local file

Download Template

Add Locale Bundle

Figure 7-29 Options Tab Showing Resources

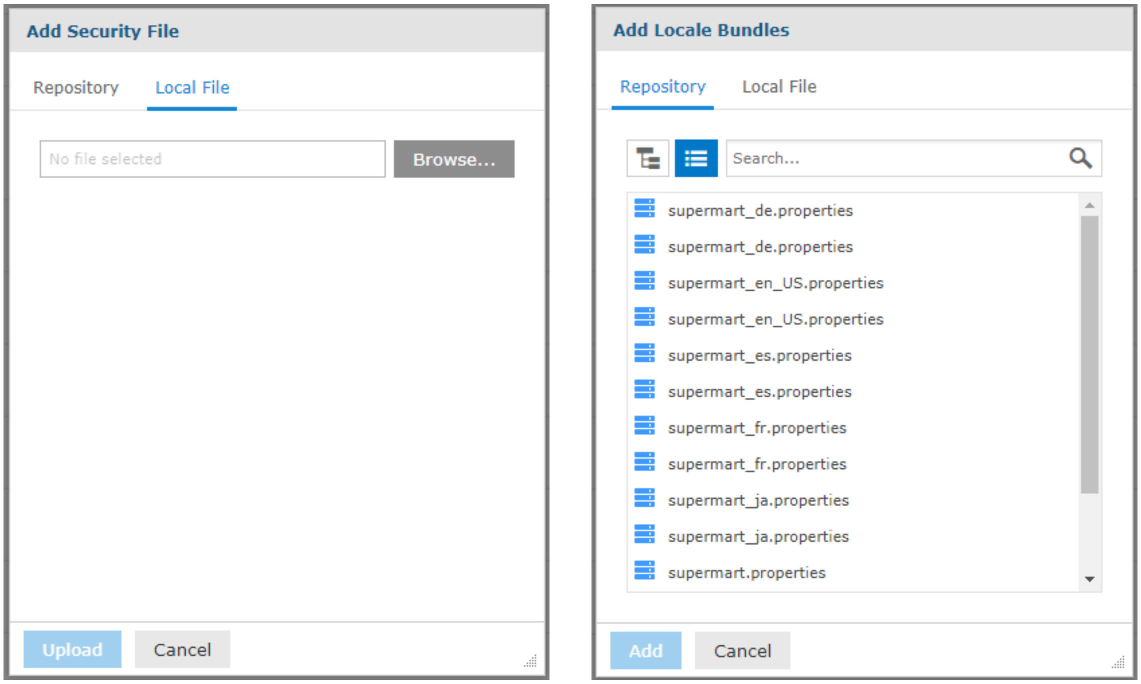
Based on the current resources attached to the Domain, some or all of the following actions are available:

- **Add Security File** or **Replace Security File** – Attaches a new security file. If a security file is already attached, it is removed from the Domain and the new one is attached. Repository resources are not removed from the repository.
- **Add Locale Bundle**– Attaches a new locale bundle. Existing local bundles are preserved.
- 🗑 – Removes the resource from the Domain. Repository resources are not removed from the repository.
- ⤓ – Downloads the selected resource as an XML or .properties file.

- **Download Template** or **download the template** – Downloads a file with bundle keys based on the labels and descriptions created on the Data Presentation tab.

To attach a security file or locale bundle files to a Domain:

1. If you want to use a shared, non-local file, make sure that the file has been uploaded to the repository. If you want to attach files locally, skip this step.
2. Create a new Domain or edit an existing one.
3. Go to the Options tab in the Domain.
4. Click the action you want: **Add Security File** or **Replace Security File** for a security file or **Add Locale Bundle**. The corresponding dialog box appears.



supermart_de.properties
supermart_de.properties
supermart_en_US.properties
supermart_en_US.properties
supermart_es.properties
supermart_es.properties
supermart_fr.properties
supermart_fr.properties
supermart_ja.properties
supermart_ja.properties
supermart.properties

Figure 7-30 Add Security File and Add Locale Bundle Dialog Boxes

5. Select the **Local File** or **Repository** tab and then browse to select a file or repository object.

   In repository view, your view is restricted to specific file types: XML files for security files and .properties files for locale bundle objects. If you are attaching a security file, make sure to select an XML file created explicitly as a security file (uploaded as Access Grant Schema).
6. Click **Select** to upload or attach 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 fundamental syntax error, the file is not added to the list of resources and you must select another file or click **Cancel**.
7. To upload or attach additional files, repeat **step 4** through **step 6**.

If you modify 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*.

7.4 Derived Tables and Calculated Fields

You can add structure to your domain using SQL queries (as derived tables) and simple calculations (as calculated fields). Once created, derived tables and calculated fields can be used like any other Domain components, for example, to create joins, define additional calculated fields, or create pre-filters. They can also be referenced in the Domain's security file and locale bundles.

7.4.1 Derived Tables

You create a derived table in a Domain using a custom query. 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.

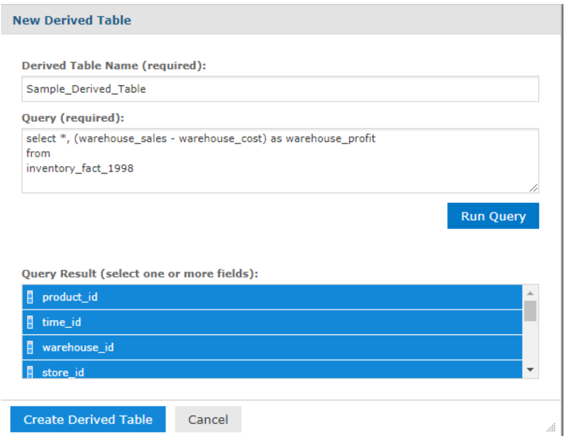
For example, with a JDBC data source, you can write an SQL query that includes complex functions for selecting data. You can use the items in a derived table for other operations on the Domain, such as joining tables, defining a calculated field, or filtering. The items in a derived table can also be referenced in the Domain's security file and locale bundles.

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. Right-click the data source node on the Data Management tab and select **Create Derived Table...** or select **Create Derived Table...** from the + menu.

   The New Derived Table dialog appears.



Derived Table Name (required): Sample_Derived_Table

Query (required):

```
select *, (warehouse_sales - warehouse_cost) as warehouse_profit
from
inventory_fact_1998
```

Query Result (select one or more fields):

- product_id
- time_id
- warehouse_id
- store_id

Figure 7-31 New Derived Table dialog

2. Type a name for the table in the **Derived Table Name** field.
3. Enter a valid SQL query in **Query**. Only queries that begin with the `SELECT` statement are allowed. Stored procedures and functions are supported. Do *not* include a closing semi-colon (`;`).

   To use an attribute, enter `{attribute('AttributeName')}` or `{attribute('AttributeName', 'Level')}`. This must be a single-valued attribute; collections can't be used. See **7.5, “Using Attributes in the Domain Designer,” on page 245** for more information.

4. When the query is complete, click **Run Query** to test it. If the query is successful, the resulting fields are displayed in the Query Result list. By default, all columns in the result are selected.
5. Ctrl-click fields in the Query Result list to change the selection. If you want only a few columns out of many, it may be easier to specify the column names in the `SELECT` clause of the query.
6. Click **Create Derived Table**.

   The derived tabled is added under the Derived Tables node in the Data Structure panel. A distinctive icon identifies it as a derived table.

To copy a derived table:

1. Right-click the derived table in the Data Structure panel, and select **Copy Table**.

To edit a derived table:

1. Right-click the derived table in the Data Structure panel, and select **Edit Derived Table...**. The Edit Derived Table dialog appears.
2. Edit the Derived Table Name, Query, and selected tables in the Query Result list.
3. Click **Update Derived Table** to save changes.

To remove a derived table:

1. Right-click the derived table in the Data Structure panel, and select **Remove Derived Table...**.

You can't copy, edit, or remove derived tables on the Data Presentation tab.

7.4.2 Calculated Fields

You create a calculated field for a Domain in two ways:

- By writing an expression that computes a value based on the data in one or more columns in a single table or join tree. All columns in the expression for a calculated field must be from the same join tree.
- By creating a constant expression. 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.

Calculated field expressions use the Domain Expression Language, fully described in **"The DomEL Syntax" on page 297**.

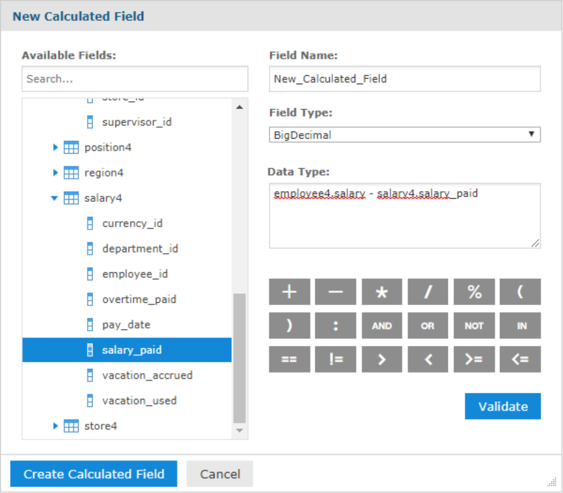
Calculated fields appear in the Data Structure panel on the Joins, Pre-filters, and Data Presentation tabs. They are identified by a distinctive icon , and the field name is shown in bold. In addition, if a calculated field is used by another calculated field, its name is also in italics. For example, supposed you have defined calcField1 and then you create calcField2, using calcField1 in its expression. calcField1 will be shown in bold and italics in the Data Structure panel.

Calculated fields from a table or join tree appear under their parent in the Data Structure panel. Global constant fields appear above the data source.

Calculated fields may be used to compute other calculated fields.

To create a calculated field for a table or join tree:

1. Right-click on the table or join tree on the Joins tab and select **Create Calculated Field**. The New Calculated Field dialog appears.



Field Name: New_Calculated_Field

Field Type: BigDecimal

Data Type: employee4.salary - salary4.salary_paid

Figure 7-32 New Calculated Field

2. In **Field Name**, enter the name you want to use for the calculated field. This name becomes the field's ID in the Domain.

   After it has been created, you can give the calculated field a more meaningful label and full description on the Data Presentation tab.

3. In **Data 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 for the calculated field in the Formula text box:
   - To insert a reference to the value of another column, find it in the **Available Fields** list and double-click the column name. The column name appears in the expression at the cursor, qualified by its table name. You can also type a column name directly in the Formula box, in the format `tablename.fieldname`.
   - To add a supported operator, click the operator icons below the Formula box, or type the operator directly.
   - To use an attribute, enter `attribute('AttributeName')` or `attribute('AttributeName', 'Level')`. This must be a single-valued attribute; collections can't be used. See **7.5, "Using Attributes in the Domain Designer," on page 245** for more information.

   Calculated field expressions use the Domain Expression Language, fully described in **"The DomEL Syntax" on page 297**.

5. Click **Validate** to verify the calculated field is valid. You must fix any errors before you can save; the error message can help you with this.

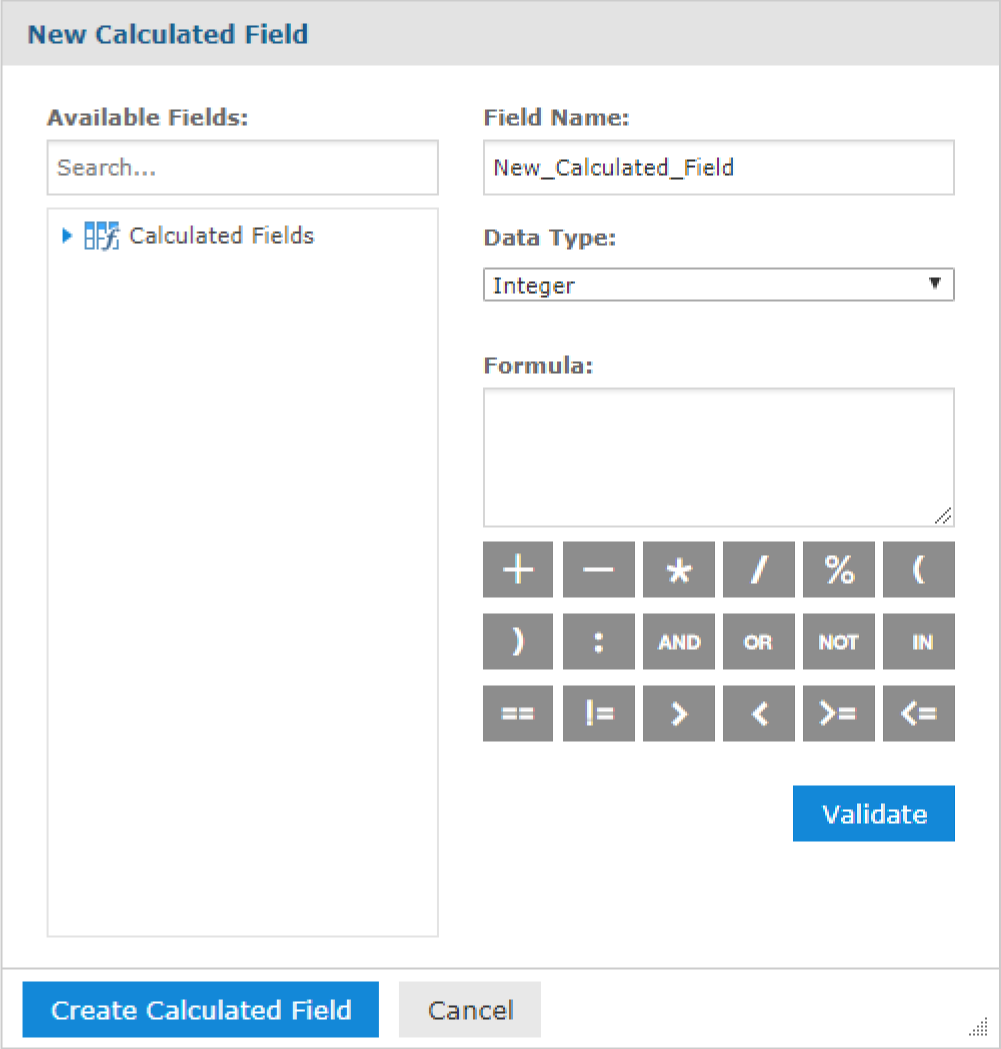
6. Click **Create Calculated Field** to save the new calculated field.

If the calculated field is valid, it appears in the Data Selection panel under the table or join tree you chose. A distinctive icon identifies it as a calculated field.

To create a constant field:

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 appear at the same level as join trees in the Data Structure panel.

1. Select **Create Constant Field...** from the + menu. The New Calculated Field dialog appears. The Available Fields list shows any other constant fields you have created.



New Calculated Field

Available Fields:

Search...

Calculated Fields

Field Name: New_Calculated_Field

Data Type: Integer

Formula:

Figure 7-33 Creating a Constant Field

2. In **Field Name**, enter the name you want to use for the calculated field.
3. In **Data Type**, select a datatype for the calculated field.

4. Enter a constant expression for the calculated field in the Formula text box:
   - To add a supported operator, click the operator icons below the Formula box, or type the operator directly.
   - To use an attribute, enter `attribute('AttributeName')` or `attribute('AttributeName', 'Level')`. This must be a single-valued attribute; collections can't be used. See **7.5, "Using Attributes in the Domain Designer," on page 245** for more information.
5. Click **Validate** to verify the calculated field is valid. You must fix any errors before you can save; the error message can help you with this.
6. Click **Create Calculated Field** to save the new calculated field.

If the constant field is valid, it appears in the Data Selection panel under the Constant Fields node . A distinctive icon identifies it as a calculated field.

To edit a calculated field:

1. Right-click the calculated field in the Data Structure panel on the Joins tab and select **Edit Calculated Field**. The Edit Calculated Field dialog appears.
2. Modify its name, its type, or its expression.
3. Click **Update Calculated Field** to save your changes.

To delete a calculated field:

1. Right-click the calculated field in the Data Structure panel on the Joins tab and select **Remove Calculated Field**.

7.5 Using Attributes in the Domain Designer

You can use attributes in the Domain Designer to have the server derive values at run time, based on the user, organization, or server. The Domain Designer supports attributes in the following places:

- Schema names
- Derived table queries
- Calculated fields
- Custom joins
- Pre-filters

In addition, you can use attributes in a data source definition. This does not appear directly in the Domain Designer, but allows you to change the data source based on the attribute value.

7.5.1 Attribute Syntax

When referring to an attribute, you can specify the attribute categorically or hierarchically:

- Categorical reference – If you specify a category for the attribute value, the server attempts to find that particular value of the attribute. You can specify these attribute categories:
  - User – In the attributes defined on the logged-in user.
  - Tenant – In the attributes defined on the organization of the logged-in user.
  - Server – In the attributes defined at the server-level.
- Hierarchical reference - If you don't specify a category for the attribute, the filter searches attributes hierarchically and uses the value for the first attribute it finds with the given name. This search starts with

the logged-in user, then proceeds to the user's organization and parent organization, and finally to the server level.

For join expressions, calculated fields, and pre-filters, the following syntax is used:

- `attribute('attributeName', 'server')` for a server-level attribute.
- `attribute('attributeName', 'tenant')` for an organization-level attribute.
- `attribute('attributeName', 'user')` for a user-level attribute.

If no level is specified, the server will search for the attribute hierarchically, starting at the `'user'` level:

- `attribute('attributeName')`

The simplified syntax is used in locations that support the internal Domain syntax, DomEL. See **8.2, “The DomEL Syntax,” on page 297**.

For schema names and derived tables, the attribute must be escaped with curly brackets ({}):

- `{attribute('attributeName'), 'server'}` for a server-level attribute.
- `{attribute('attributeName', 'tenant')}` for an organization-level attribute.
- `{attribute('attributeName', 'user')}` for a user-level attribute.
- `{attribute('attributeName')}` to search for an attribute hierarchically.

For more information on attributes, see the section "Managing Attributes" in the *JasperReports Server Administrator Guide*.

7.5.2 Things to Remember When Using Attributes

Note the following when using attributes:

- The attribute value must match the expected data type. For example, if you are creating a calculated field of type Integer, the attribute value must be an Integer.
- Attribute collections are not supported in the main Domain design. The attribute must be a single value. Attribute collections can be used in Domain security, however.
- If you are using attributes for the schema, make sure that the tables and columns referenced in the Domain are available for every possible value of the attribute. See **7.6.1, “Changing the Data Source,” on page 249** for more information.

If you use an attribute in your data source definition, you must ensure that referenced schemas, tables, and columns will be available. Data source attributes are defined outside the Domain Designer. See the section on defining attributes in the *JasperReports Server Administrator Guide* for more information.

- If a specified attribute does not exist, any item that uses it will fail. For example, if you specify the attribute `Country` for a schema name, and a `Country` attribute does not exist anywhere on the server, you will be unable to load the schema. This is similar to referring to column that does not exist.
- If an attribute exists but is not defined (empty) for this particular user, the behavior depends on where the attribute is used. An empty attribute can occur, for example, if you have set up a `Country` attribute, but the current user has no value defined for `Country`.
  - For pre-filters, an empty attribute is interpreted based on the field's data type as described below.
    - For a field of type String, an empty attribute is interpreted as an empty string: `""`
    - For a Boolean field, and empty attribute is interpreted as `FALSE`.
    - For numeric and date fields, an error will occur.

- For derived table queries and custom joins, the attribute will fail in most cases. However, in some cases, JasperReports Server will consider the resulting expression to be valid, with potentially unexpected results, so be cautious when using attributes in this part of your design.
- For schema names, the default schema is used. The default schema must be defined in the XML Domain design file. See **8.1.5.5.1, “Default Schema,” on page 264** for more information.

7.5.3 Using Attributes for Pre-Filters

For example, you can use attributes to create a pre-filter for a country. Instead of a static filter that filters out data for every country except the USA, you could create a filter that only shows data for the logged-in user's country. To do this, you would create a Country attribute for your users in their profiles and enter the name of their country, which must match one of the country names used in the Domain's data source. Then, when creating the pre-filter, you can use the parameter `attribute('Country')` to filter out data for all countries except the user's country.

For pre-filters, the data type for the attribute is automatically based on the filtered column's field type. For example, if you are creating a filter based on a Date column, the attribute parameter will return a Date value.

7.5.4 Using Attributes for Schema Names

For example, you could use attributes to choose the schema in your database. For example, suppose you have schemas named dev and qa in your database. You could create a server-level attribute and set it to `dev` on one server and `qa` on another. You can then use the attribute for the database schema in your Domain. Similarly, in a multi-organization deployment, you could have different schemas for different organizations, and set an organization-level attribute to choose which schema to access.

The Domain Designer needs a concrete schema in order to read and display tables. When you use an attribute for a schema name, the Domain Designer automatically substitutes the current value of the attribute. For the example above, if you use the Domain Designer on a server where the attribute is set to `dev`, the Domain Designer displays the `dev` schema in the Selected Schemas list and in the Select Data panel. You will see the tables and columns in the `dev` schema in the Domain Designer UI. Any joins, pre-filters, or presentation sets you create are based on that `dev` schema.

- To find the underlying attribute for a schema, go to the Data Management tab, select the database in the Data Structure panel, and then click on the schema name in the Selected Schemas list. The attribute details appear in the **You may also use an attribute for the schema name:** text box.

When you use an attribute for a schema name, the Domain will fail if the schema is missing. You can optionally set a default schema in the Domain design file. The default schema is used whenever the schema is missing or undefined. See **8.1.5.5.1, “Default Schema,” on page 264** for more information.

7.6 Editing Domains

You can add, change, and delete domain components.

Use caution when editing Domains that have been used for Ad Hoc views or Domain Topics. By default, if you remove or change Domain elements that are used in an Ad Hoc view or Topic, any dependencies are removed the next time the Ad Hoc view or topic is opens. In some cases, you might want to create a placeholder field to preserve the structure of the Ad Hoc view.

To edit a Domain:

1. Log into the server as an administrator and navigate to the Domain location in the repository.
2. Right-click the Domain and select **Edit** from the context menu.

   The Domain Designer opens on the Data Presentation tab. Click the tab with the information you want to edit and make your changes.
3. If you want to change the data source, select **Replace Data...** from the ☰ ▾ menu, select the data source you want in the Choose Data dialog, and click **OK**.

   If you change to a data source with a different database, the definitions in the Domain design may become invalid and you can't save the Domain. See **7.6.1, “Changing the Data Source,” on page 249** for more information.

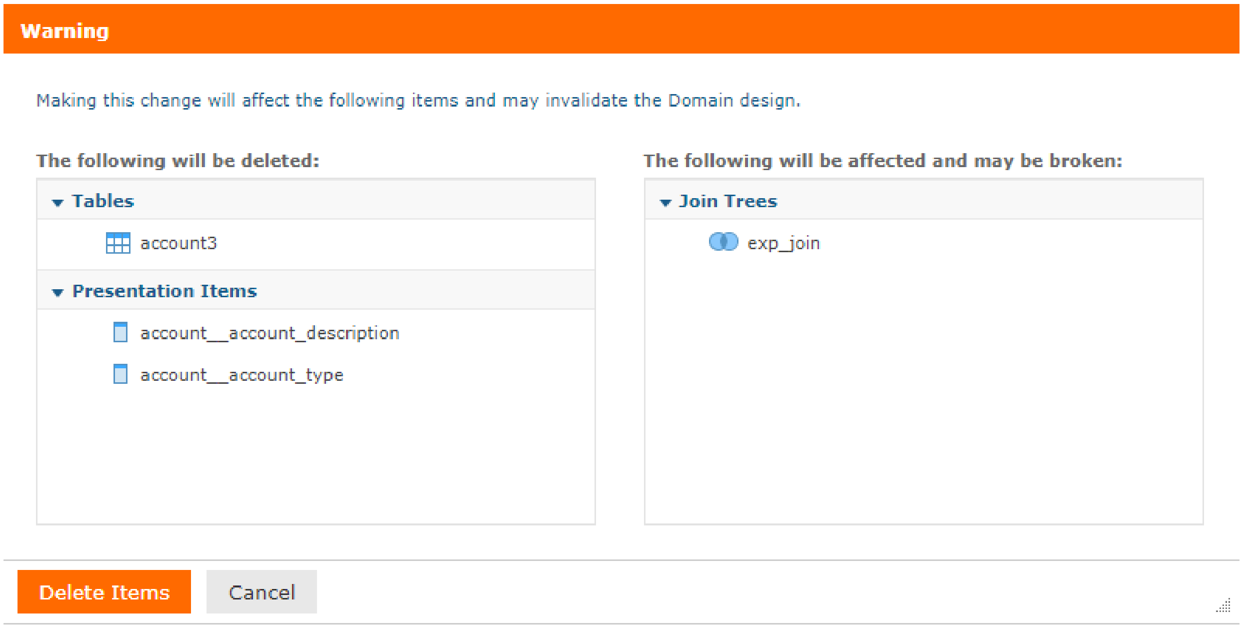
   Before you switch the data source for a Domain, back up the Domain by exporting a Domain design file.
4. Save the Domain.

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*.

7.6.0.1 Removing Schemas, Tables, and Joins

If you attempt to remove a schema, table, or join that is used in the Domain Designer, a warning dialog lists the affected components of your design. Click **Delete Items** to continue and delete all the items listed, or click **Cancel** to keep the original data source.

If a schema or table is used in a join tree, the individual joins that reference the table are deleted, but other joins in the join tree are preserved. This may affect your design. You should examine the updated join tree to see how it has been affected.



Warning

Making this change will affect the following items and may invalidate the Domain design.

The following will be deleted:

- Tables
  - account3
- Presentation Items
  - account__account_description
  - account__account_type

The following will be affected and may be broken:

- Join Trees
  - exp_join

Figure 7-34 Deleted Table Warning

7.6.1 Changing the Data Source

When you open an existing Domain with the Domain Designer, the Domain Designer reads the current state of the database and uploads it to the design. The Domain Designer does not detect changes to the database schemas, tables, and columns in real-time, but it will detect them the next time you open the Domain in the Domain Designer. The data source or structure can change for any of the following reasons:

- You manually change the JasperReports Server data source by selecting Replace Data from the menu.
- The selected data source or a schema uses an attribute and the attribute changes or you log in with a different attribute value.
- The structure of the underlying database or other data source has changed. For example, if a table or column is added to or deleted from the database, it is detected the next time you open the Domain Designer.

Missing schemas, tables, or columns in the new or updated data source are automatically deleted from the design along with any elements, such as calculated fields, joins, or sets, that depend on them.

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.

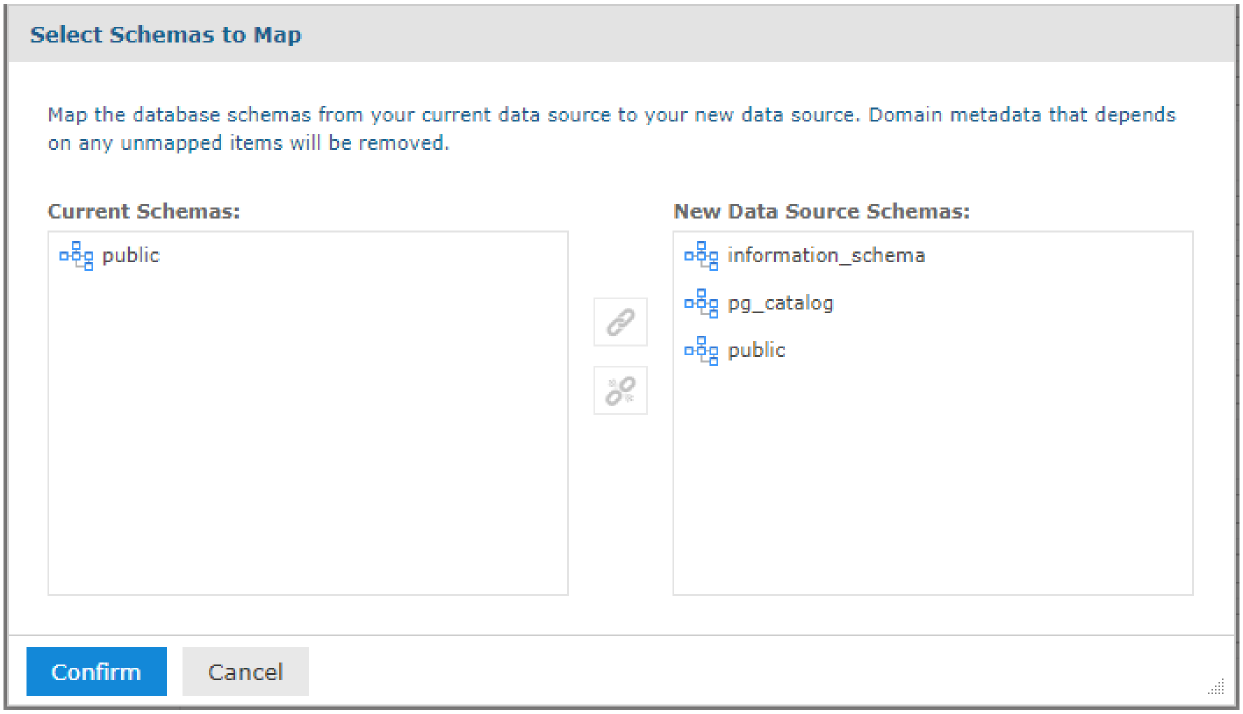
Before you select a new data source manually or use attributes to specify data sources and/or schemas, do your best to ensure that the schema structures are identical. You should also export your Domain design before changing the data source manually.

7.6.1.1 Selecting a New Data Source

To select a new data source:

1. Select **Replace Data...** from the ☰ ▾ menu
2. Select the data source you want in the Choose Data dialog, and click **OK**.

   The Select Schemas to Map dialog appears. This dialog shows the schemas used by your Domain on the left, and the schemas available in the data source on the right.



Select Schemas to Map

Map the database schemas from your current data source to your new data source. Domain metadata that depends on any unmapped items will be removed.

Current Schemas:
- public

New Data Source Schemas:
- information_schema
- pg_catalog
- public

Figure 7-35 Selecting schemas to map

If you are switching between a schemaless data source (such as MySQL) and a schema-based data source (such as Oracle), the dialog is different, as follows:

- If the original data source is schemaless and the new data source supports schemas, the Current Schemas list is omitted. Select a new schema from the available schemas.
- If the original data source has schemas and the new data source is schemaless, the New Data Source Schemas list is omitted. Select one of the schemas in your Domain to use for the new data source. In addition, if your Domain has only one schema and you are selecting a schemaless data source, the schema is mapped automatically and you do not see this dialog.

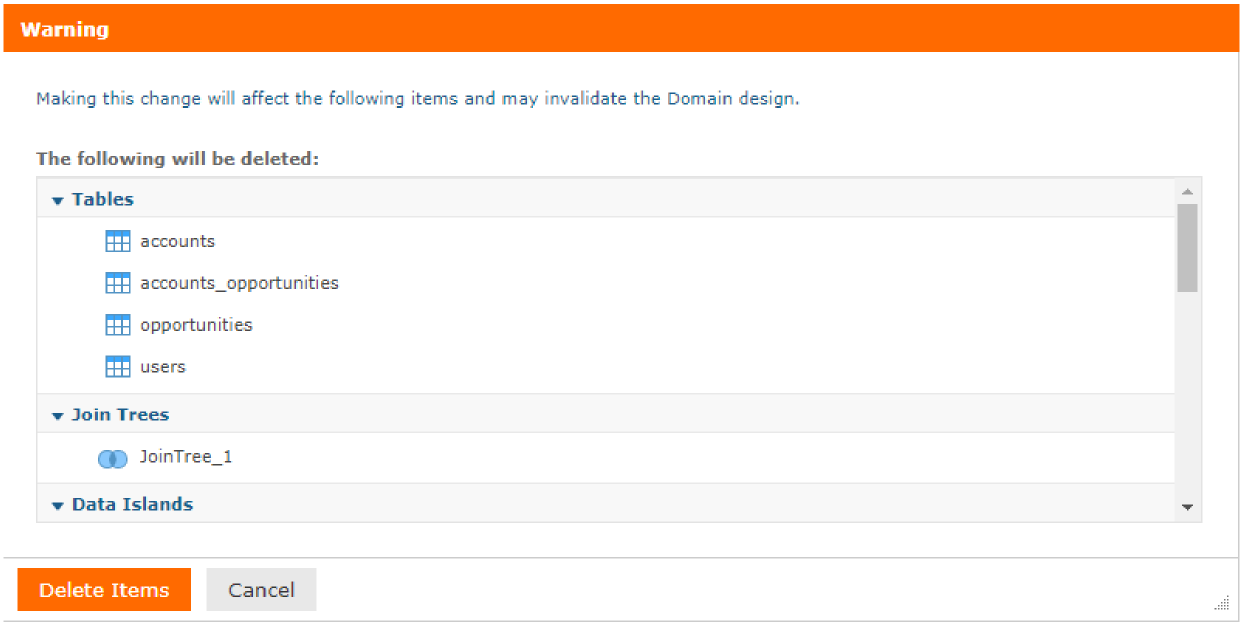
3. To associate an existing schema in your Domain with a schema in the data source:
   a. Select the schema in the Domain in the Current Schemas list.
   b. Select the schema you want to replace it with in the New Data Source Schemas list.

   c. Click 🔗.

      The schemas are associated. A number is added after the schema name to make it easier to manage multiple schemas.

4. Repeat the previous step for each schema you want to map. If you want to unlink two schemas, click ⛓️‍💥.
5. Click **Confirm** to map the schemas.

   JasperReports Server retrieves the data structure from the data source and validates the Domain. If all the tables and columns in your Domain are found, the Domain Designer opens at the Data Management tab.
6. If there are missing tables or columns, a warning dialog lists the affected components of your design. Click **Delete Items** to continue and delete all the items listed, or click **Cancel** to keep the original data source.



Warning

Making this change will affect the following items and may invalidate the Domain design.

The following will be deleted:

- **Tables**
  - accounts
  - accounts_opportunities
  - opportunities
  - users
- **Join Trees**
  - JoinTree_1
- **Data Islands**

Figure 7-36 Warning dialog for changing data source

If you click **Delete Items**, JasperReports Server deletes all the displayed items. There may still be some errors you need to resolve before the Domain is valid.

7.6.1.2 Changes Due to Attributes

If you use attributes to define a database schema, a calculated field, or another Domain element, you may see the following when you log in as a different user:

- If a schema is missing, the Select Schemas to Map dialog appears. See **7.6.1.1, "Selecting a New Data Source," on page 250** for more information. This can happen because the attribute is no longer defined or because the attribute is defined but resolves to a non-existent schema.
- If you use an attribute for the schema name and you later select a different schema in the Select Schemas to Map dialog, the design will always use that schema; the schema will no longer be attribute-based.
- If any tables or fields are missing in the new data source or schema, JasperReports Server displays a warning dialog. Clicking **Delete Items** deletes the missing tables and columns and any Domain design elements that depend on them.

- If an attribute is not defined or does not resolve to a usable value, JasperReports Server displays a warning dialog. Clicking **Delete Items** deletes any Domain design elements that depend on the undefined or invalid attribute(s).

> If you use an attribute in your data source definition, you may see similar problems. Data source attributes are defined outside the Domain Designer. See the section on defining attributes in the *JasperReports Server Administrator Guide* for more information.

7.6.1.3 Changes to the Underlying Data Source Structure

To update a Domain after changes to the underlying data structure, refresh the browser, or close the Domain Designer then open the Domain for editing again.

Keep the following points in mind regarding data source updates:

- You will be prompted to remap any schemas no longer found in the database.
- If your design uses tables and columns that are no longer found in the data source, JasperReports Server displays a warning dialog. Clicking **Delete Items** deletes the missing tables and columns and any Domain design elements that depend on them.
- New schemas, tables, and columns appear in the Data Structure panel on the Data Management tab. JasperReports Server has no way of identifying schemas, tables, or columns that have been added since the last time you edited the Domain.
- To add a new column to the Domain, move its table to Selected Tables. If the table is already selected, the column is automatically available. Table selection works only with entire tables.

7.6.2 Domain Validation

Validating a Domain ensures the consistency of all its components. The Domain Designer checks the syntax of files when they're uploaded, and overall consistency is also checked when saving a new or edited Domain.

During validation, the Domain Designer does the following:

1. Verifies that the tables and columns of the Domain design exist in the data source (validation against 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, you will not be able to save the Domain. 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 occurs at the following times:

- When you open a Domain for editing: JasperReports Server validates the Domain syntax and validates the Domain against the data source.

- When you replace the data source: JasperReports Server validates the Domain syntax and validates the Domain against the data source.
- When you upload a schema from an XML file: JasperReports Server validates the Domain syntax and validates the Domain against the data source.
- When you save the Domain: JasperReports Server validates the Domain syntax and verifies permissions.

When validation fails a message appears to help you correct the error.

7.7 Domains and Data Virtualization

There are two ways to integrate data virtually from multiple sources using Domains in JasperReports Server:

- For simple data virtualization with low volumes of data and limited join complexity, create a JasperReports Server virtual data source and build your Domain on top of it. In this case, you define the relationships that connect the tables of the different data sources using the Domain Designer. See the *JasperReports Server Administrator Guide* for more information about virtual data sources in JasperReports Server. This is a lightweight solution that doesn't require any additional server or installation.
- For more complex virtualizations, or where performance and scale are essential, use Jaspersoft Advanced Data Services to access and combine your different data sources. Once you have designed and created a integrated data source in Jaspersoft Advanced Data Services, you create a JasperReports Server JDBC data source to access it and then use the Domain Designer to choose the tables you want and create user-friendly names. Using Jaspersoft Advanced Data Services offers better performance and scaling, with management tools designed specifically for combining multiple data sources. Contact your sales representative for information about licensing for Jaspersoft Advanced Data Services. This requires a separate installation for Jaspersoft Advanced Data Services.

7.7.1 Notes on Using Virtual Data Sources

Normally, you combine data sources that have relationships you want to explore. To create a useful Domain using a virtual data source, you need to define relationships that connect the tables of the different data sources. You do this by adding at least one schema from each data source and then creating a join between tables in the different schemas. If the data sources don't have a column that matches exactly, you can create derived tables that allow you to implement the join you want. For example, if you use the sample SugarFoodMartVDS virtual data source, which combines the Foodmart and SugarCRM JDBC data sources, you can 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 can join the two data sources on `city` to combine them.

Domains built from virtual data sources may have additional syntax restrictions. For example, Domains built from a virtual data source do not work with table names that contain \.

7.8 Importing and Exporting Domain Design Files

You can export and import Domain design files in XML format. This lets you copy or back up Domains, even between different servers. Domain design files can be also created or edited manually in a text editor or programmatically.

The results of editing or using a Domain based on an inconsistent or incorrect design file are unpredictable. Before you modify Domain design files you need to be familiar with the Domain syntax and structure. See **Chapter 8, “Working with Domain Files,” on page 257** for more information on the XML syntax.

7.8.1 Exporting a Domain Design File

1. Log into the server as an administrator and edit an existing Domain or create a new one.
2. Save any changes you have made to the Domain.
3. Click on the menu bar to export a Domain schema file, schema.xml, in XML format.

7.8.2 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.

Uploading a design file overwrites the existing Domain and may break Ad Hoc views, Topics, or reports that depend on the Domain.

To upload a Domain design file:

1. Log into the server as an administrator and edit an existing Domain or create a new one.
2. Click on the menu bar.
3. **Browse** to find the design file you want to upload, select it, and click **Open**.

   A warning dialog appears.
4. Click **Update Domain Design** to continue.

   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 syntax of the uploaded file. If there are syntax or semantic errors, the current design is not replaced. . You may also see warnings if you have changed the data source, schema names, or deleted schemas, tables, or columns from the Design file. If there are any errors or inconsistencies, you should make changes to the design 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 problems, or are unsure that the result will be what you want, click **Cancel**.
5. After the design appears correctly in the Domain Designer, make any further modifications on any of the tabs.
6. Select **Save Domain** from the ▾ menu 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*.

7.8.3 Exporting a Domain and Resources

The export option on the Domain Designer menu exports the Domain design file, which describes all the features visible in the Domain Designer. This is a good option if you want to modify the Domain in a text editor. However, a Domain in the JasperReports Server needs additional information, including: permissions, the data source in the JasperReports Server repository, the security file (if it exists), and any translation files. To export a Domain with all its associated resources:

1. Log into JasperReports Server as an administrator.
2. Browse in the repository to the Domain you want to export.
3. Right-click on the Domain and select **Export** from the context menu. The Export Resources dialog appears.
4. Enter a name for the exported file, select the options you want, and click Export. The file is exported in .zip format to the Downloads folder set by your browser.

See the *JasperReports Server Administrator Guide* for more information on exporting repository files.

Chapter 8 Working with Domain Files

The Domain Designer offers a graphical interface for creating and editing Domains, but you can also export a Domain in XML format and edit it outside of JasperReports Server. Some features, such as default schemas and joins between two columns in the same table, can be configured only in the design file. In other cases, it may be easier to work directly with the XML to accomplish certain tasks. Design files can also simplify the sharing of Domains between systems.

> The results of using a Domain based on an inconsistent or incorrect design file are unpredictable and can be difficult to resolve. Make sure you thoroughly understand Domains and Domain syntax before editing any Domain design files.

Domain security and translation of Domain labels and descriptions are also configured via optional text-based files that you 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.

This section describes the structure of the XML Domain design file and the structure and use of properties files for internationalization. For information about Domain security and Domain security files, see the *JasperReports Server Security Guide*.

This chapter contains the following sections:

- **Understanding Domain Design Files**
- **XML Design File Reference**
- **The DomEL Syntax**
- **Localizing Domains**
- **The Domain Security File**

8.1 Understanding Domain Design Files

When you design a Domain, you specify the tables in the data source, derived tables, joins, calculated fields, and filters, as well as how those elements appear to users. You can design a Domain in JasperReports Server interactively through the Domain Designer, or you can use the file-based format to create or modify Domains. A text file containing a Domain design in XML is called a Domain 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.

For information about exporting and uploading design files, see **7.8, "Importing and Exporting Domain Design Files," on page 253**.

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.

8.1.1 Design File Use Cases

Due to the complexity of a valid design file, it is usually much easier to begin with a basic design file exported from the Domain Designer or to modify an existing design file. Common use cases for working with design files include:

- **Enhancing a design** – You can use the design file to add features that can't be defined in the Domain Designer. To do this, use the Domain Designer to define as much of the design as possible, then export the design file and modify the exported file directly. Some features can only be defined in the design file. You can still upload these files to the Domain Designer and you can modify the other features of the Domain.
- **Defining presentation for 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 may be faster when they appear in a single design file.
- **Making repetitive changes** – If the database changes or you want to move a design file to a different system, you may need to edit many elements in the same way. Using search-and-replace in an external editor can speed up this process.
- **Programatically creating Domain designs** – If you are creating many similar Domains, you may be able to start with a valid design file and generate similar design files that meet the constraints of JasperReports Server.
- **Creating files for security and translations** – These optional files refer to elements of the Domain design, and it is often more convenient to copy-paste them between external files. For translation files, you can export a template from the Domain Designer as a basis for the translation files. See **7.8, "Importing and Exporting Domain Design Files," on page 253**.

After editing, a design file can be uploaded, validated, and used to define the design of a new or existing Domain. With a few exceptions, when you open the design in the Domain Designer again, the modifications are editable in the Designer. For example, a description you added in the XML design file appears on the Data Presentation tab and you can edit it in the Designer. Other elements of the XML file appear on some or all of the Domain Designer tabs.

8.1.2 Features That Cannot Be Defined in the Domain Designer

The following elements can be edited only in the design file. Design files with these elements can still be opened in the Domain Designer, and other elements can be edited:

- Joins that use `or` or `not` between two join expressions. These appear in the Domain Designer as read-only joins. See **7.3.5.3.7, "Read-Only Joins," on page 226** for more information.
- Joins between two columns in the same table. These appear in the Domain Designer as read-only joins. See **7.3.5.3.7, "Read-Only Joins," on page 226** for more information.

- The `defaultSchema` property when using JasperReports Server attributes for a schema name. This does not appear in the Domain Designer directly, but is used when the attribute exists but has null value. See **8.1.5.3, "schemaMap," on page 263** for more information.
- Individually-selected columns of a table. You can only select whole tables on the Data Management tab of the Domain Designer.

8.1.3 Structure of XML Domain Design Files

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 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.
- The following symbols can't be entered directly; they must be escaped using their HTML encoding:

  & (`&amp;`), " (`&quot;`), < (`&lt;`), > (`&gt;`);
- It must be valid with regard to the XML schema. The XML schema defines allowed XML element and attribute names and how they're nested in a hierarchical structure. This ensures that the elements and attributes are those used by JasperReports Server. The allowed versions of the XML schema of a Domain design are given in the XSD files located in <install-dir>/samples/domain-xsd/. This document describes the functionality supported by the following version:

  <install-dir>/samples/domain-xsd/schema_1_3.xsd
- The design file must be internally consistent. You can upload a partial file, but to use the Domain, you must define all the necessary elements of a Domain design. These constraints cannot be expressed in the XSD file because they are outside the scope of an XML schema.

It is possible to create or upload partial Domain files, but in order to be usable, a Domain file must include certain elements, such as: a data source, one or more tables, and one or more presentation items.

- 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 may fail when used in a Domain with a different data source.

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. 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.
- 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.
- Expressions for filters and calculated fields must be valid programmatic expressions in Jaspersoft's Domain Expression Language (DomEL). This format is documented in **"The DomEL Syntax" on page 297**.

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.

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_3.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.

No editor can enforce the internal and external constraints on a design file. The following sections explain the elements and attributes of an XML design file and the various constraints that apply to each of them.

8.1.4 XML Design File Reference

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.

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 object.

Technically, the only requirement to save or upload a Domain design file is a data source reference. However, for the file to be usable, many more elements are needed. As a short-hand, elements you need to create a usable Domain are marked "required".

The following symbols can't be entered directly; they must be escaped using their HTML encoding:

& (`&amp;`), " (`&quot;`), < (`&lt;`), > (`&gt;`);

8.1.4.1 schema

The `schema` element is the outermost container element of an XML Domain design file. The `schema` element is required.

The `schema` element refers to the XML schema. It has no relationship to database schemas in the data source.

8.1.4.1.1 schema Hierarchy

The following hierarchy is used for the `schema` element:

```
<schema xmlns="http://www.jaspersoft.com/2007/SL/XMLSchema" version="1.3"
       schemaLocation="schema_1_3.xsd">
       <dataIslands> (0...1)
       <dataSources> (0...1)
       <itemGroups> (0...n)
       <items> (0...n)
       <resources> (1)
```

When you export a file from the Domain Designer, these elements appear alphabetically in the file. In this reference, they are presented in the following order, based on their function in the Domain:

- Data sources and database schemas (`dataSources` element).
- Domain model (`resources` element); includes tables, derived tables, calculated fields, joins, and pre-filters.
- Domain presentation (`dataIslands`, `itemGroups`, and `items` elements).

8.1.4.1.2 Child Elements

Element Name	Description
`dataIslands`	Container for a list of data islands in the Domain design. See **8.1.12, "Representing Sets and Items in XML," on page 286**.
`dataSources`	Container for the Domain data source and schemas. See **8.1.5, "Representing Data Sources and Database Schemas in XML," on page 262**.
`itemGroups`	Containers for sets; each data island listed in the `dataIslands` element must have a corresponding `itemGroups` element. See **8.1.12, "Representing Sets and Items in XML," on page 286**.
`items`	Containers for items that aren't in any set. Items in different `dataIslands` must be in different `items` elements. See **8.1.12, "Representing Sets and Items in XML," on page 286**.
`resources`	(Required) Container for all element definitions that make up the Domain model in the design, including: • tables (see **8.1.7, "Representing Tables in XML," on page 267**) • derived tables (see **8.1.8, "Representing Derived Tables in XML," on page 271**) • joins (see **8.1.9, "Representing Joins in XML," on page 274**) • calculated fields (see **8.1.11, "Representing Calculated Fields in XML," on page 283**) • pre-filters (see **8.1.10, "Representing Pre-filters in XML," on page 282**) Because the elements under `resources` refer to database objects, they must be externally consistent with the data source for the Domain.

8.1.4.1.3 XML Attributes

Attribute	Type	Description
`schemaLocation`	String	Sometimes added by XML editors to locate the XSD file.
`version`	String	Version of the XSD used to create this design file. Included in design files exported from Domain Designer.
`xmlns`	String	XML namespace URI. This string is unique to Jaspersoft, but does not correspond to a valid URL. For more information, see http://www.w3.org/TR/REC-xml-names. Included in design files exported from Domain Designer.

8.1.5 Representing Data Sources and Database Schemas in XML

Data sources and database schemas are declared under the `dataSources` element. A data source for a Domain is declared using `jdbcDataSource` and a data source for a Topic is declared using `jrQueryDataset`.

8.1.5.1 dataSources

`dataSources` is a container for the `jdbcDataSource` element. A Domain can only have at most one data source. `dataSources` is a child of the top-level `schema` element.

`dataSources` is not required for schemaless JDBC databases such as MySQL unless you are using an attribute for the database schema name.

8.1.5.1.1 Child Elements

One of the following elements is required for `dataSources`. Only one of these can be declared.

Element Name	Description
`<`**`jdbcDataSource`**`>`	Declares a data source for a Domain and the id used to reference it.
`<`**`jrQueryDataset`**`>`	Declares a data source for a Topic and the id used to reference it.

8.1.5.2 jdbcDataSource

The `jdbcDataSource` element declares a data source for a Domain and is a container for the `schemaMap` elements used in the Domain. `jdbcDataSources` does not directly reference a data source in the repository; instead, the actual link to the repository data source is attached as metadata to the Domain design in the repository.

For Topics, use **8.1.5.7, "jrQueryDataset," on page 265**. There can only be one `jrQueryDataset` or `jdbcDataSource` element in a Domain design file.

8.1.5.2.1 Element Hierarchy for jdbcDataSource

The following hierarchy is used to represent JDBC data sources and their database schemas. This hierarchy is under the `schema` element.

```
<dataSources> (1)
  <jdbcDataSource> (1)
    <schemaMap> (1...n)
      <entry> (1)
        <string> (1)
```

8.1.5.2.2 Child Elements

Element Name	Description
<**schemaMap**>	(Required) Declares a database schema used in the Domain.

8.1.5.2.3 XML Attributes

Attribute	Type	Description
id	String	(Required) Unique identifier for the data source in the Domain design file.

8.1.5.3 schemaMap

The schemaMap element is a container for entry elements that represent database schemas used in the Domain.

8.1.5.3.1 Child Elements

Element Name	Description
<**entry**>	A database schema or the default schema declaration.

8.1.5.4 entry

The entry element in the schemaMap element declares a database schema for use in the Domain. The actual name of the schema is the value of the string child element, while the identifier used in the Domain is set as the key attribute.

If key is set to defaultSchema, the entry element is used to define the default schema.

8.1.5.4.1 Child Elements

Element Name	Description
<**string**>	The name of the schema in the database.

8.1.5.4.2 XML Attributes

Attribute	Type	Description
key	String	(Required) Unique identifier for the schema in the Domain design file or the reserved string "defaultSchema". For regular database schemas in the Domain, the key attribute is used by other elements in the presentation representation to identify the schema, via the referenceId. If key is set to defaultSchema, this entry element is used to define the default schema. See **8.1.5.5.1, "Default Schema," on page 264** for more information. The key XML attribute can contain alphanumeric characters along with any combination of the following: @#$^`_~? It cannot start with a digit.

8.1.5.5 string

The string element in the schemaMap hierarchy is a string defining the database schema to be used in the Domain. The string can be one of the following:

- The exact name of a schema in the database.
- A JasperReports Server attribute. Not supported for defaultSchema. See **8.1.13, "Using JasperReports Server Attributes in Design Files," on page 294** for more information.

8.1.5.5.1 Default Schema

If the key attribute of the parent entry element is set to defaultSchema, then entry/string declares the default schema for the Domain. The default schema is a fallback schema name that is substituted when a schema name is missing or not available. This can happen, for example, if you are using a JasperReports Server attribute for the data source or schema. It can also happen if you change the data source for some reason. You can only define one default schema for a database.

8.1.5.6 Examples

The following example shows a declaration of a data source and two schemas. Note that the data source name in the Design file does not need to be linked to the data source in the repository. That linkage is done outside of the Domain design file.

Note that in the key attribute, an underscore (_) has been substituted for unsupported characters. Periods (.) are not supported in schema key attributes; numbers are supported, but not in the first character.

```
<dataSources>
  <jdbcDataSource id="myDataSource">
    <schemaMap>
      <entry key="schema_1">
        <string>schema.1</string>
      </entry>
      <entry key="_234schema">
        <string>1234schema</string>
      </entry>
    </schemaMap>
    </jdbcDataSource>
  </dataSources>
```

The following example shows a file that uses an attribute for the database schema name, and sets a default schema name to use when the attribute is null.

```
<dataSources>
  <jdbcDataSource id="dsFoodMart">
    <schemaMap>
      <entry key="_attribute_schemaAttr_">
        <string>{attribute('schemaAttr')}</string>
      </entry>
      <entry key="defaultSchema">
        <string>public</string>
      </entry>
    </schemaMap>
  </jdbcDataSource>
</dataSources>
```

If you use an attribute for the database schema name, and the attribute is undefined on the server, the Domain will fail.

8.1.5.7 jrQueryDataset

The `jrQueryDataset` element declares the data source for a Topic. `jrQueryDataset` does not directly reference a data source in the repository; instead, the actual link to the repository data source is attached as metadata to the Domain design in the repository. There can only be one `jrQueryDataset` or `jdbcDataSource` element in a Domain.

`jrQueryDataset` does not support schemas. For more information about Topics, see **5.9, "Working with Topics," on page 165**.

8.1.5.7.1 Element Hierarchy for jrQueryDataset

The following hierarchy is used to represent JDBC data sources and their database schemas. This hierarchy is under the `schema` element.

```
<dataSources> (1)
  <jrQueryDataset> (1)
```

8.1.5.7.2 XML Attributes

Attribute	Type	Description
`id`	String	(Required) Unique identifier for the data source in the Domain design file.

8.1.6 Representing the Domain Model in XML

The Domain model is represented by `resources` and its children. The Domain model is configured in the first three tabs of the Domain Designer: Data Management, Joins, and Pre-filters.

Because the elements under resources refer to database objects, they must be externally consistent with the data source for the Domain. For example, the schemaAlias and datasourceTablename attributes of jdbcTable must match the schema and name of a table in the data source.

8.1.6.1 resources

The resources element is a container for all element definitions that make up the Domain model in the design, including:

- tables (see **8.1.7, "Representing Tables in XML," on page 267**)
- derived tables (see **8.1.8, "Representing Derived Tables in XML," on page 271**)
- joins (see **8.1.9, "Representing Joins in XML," on page 274**)
- calculated fields (see **8.1.11, "Representing Calculated Fields in XML," on page 283**)
- pre-filters (see **8.1.10, "Representing Pre-filters in XML," on page 282**)

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. In order for the Domain to be usable, there must be at least one table, whether as a jdbcTable or jdbcQuery element.

resources does not contain presentation elements. resources is a direct child of the schema element.

8.1.6.1.1 resources Hierarchy

The following hierarchy is used for the resources element:

```
<resources>
       <null> (0...1)
               <fieldList> (1)
                       <field> (1...n)
       <jdbcTable> (0...n)
               <fieldList> (1)
                       <field> (1...n)
               <filterString> (0...n)
       <jdbcQuery> (0...n)
               <fieldList> (1)
                       <field> (1...n)
               <filterString> (0...n)
               <query> (1)
       <jdbcTable> (0...n)
               <fieldList> (1)
                       <field> (1...n)
               <filterString> (0...n)
               <joinInfo> (1)
               <joinList> (1)
                       <join> (1...n)
               <joinOptions> (0...1)
               <tableRefList> (1)
                       <tableRef> (1...n)
```

8.1.6.1.2 Child Elements

Element Name	Description
jdbcTable	Represents one of the following: • A table in the data source or table copy in the Domain (see **8.1.7, "Representing Tables in XML," on page 267**). • A join in the Domain (see **8.1.9, "Representing Joins in XML," on page 274**)
jdbcQuery	(Only present if schema defines derived tables.) Represents a derived table that results from an SQL query. See **8.1.8, "Representing Derived Tables in XML," on page 271** for more information.
null	(Only present if schema defines constant calculated fields.) Container for constant fields.

8.1.7 Representing Tables in XML

To represent tables, use the jdbcTable element. A table is a child of the resources element.

The jdbcTable element is also used to describe join trees. See **8.1.9, "Representing Joins in XML," on page 274**.

8.1.7.1 Table Hierarchy

The following hierarchy is used for jdbcTable elements when representing a table from the data source. For the hierarchy when representing a join, see **8.1.9, "Representing Joins in XML," on page 274**.

```
<jdbcTable>
       <fieldList> (1)
              <field> (1...n)
       <filterString> (0...n)
```

8.1.7.2 jdbcTable

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, including tables that you need to construct the Domain but do not want to expose to the end user. jdbcTable is a child of the resources element.

8.1.7.2.1 Attributes

Attribute	Type	Description
datasourceId	String	(Required) Identifier for the data source, as set in the id attribute of **jdbcDataSource**. 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.
id	String	(Required) Unique identifier for the table in the Domain design file. If you copy a table to join it multiple times, each copy has the same datasourceId, schemaAlias, and datasourceTableName but must have a different id. To set a table id in the Domain Designer, right-click the table and select **Rename Table** from the context menu. The id XML attribute can contain alphanumeric characters along with any combination of the following: @#$^`_~? It cannot start with a digit. When generated via the Domain Designer, id is a representation of the table name in the data source, with unsupported characters replaced by underscores.
schemaAlias		Name of the database schema in the data source. Required if the data source uses schemas. • For schemaless data sources, such as MySQL, schemaAlias is not required. • For schema-based data sources, such as PostgreSQL and Oracle, the name is the literal name of the schema in the data source, for example, public. • For schemaless data sources within a virtual data source, the schema name is of the form dataSourcePrefix, where dataSourcePrefix is the data source prefix defined when the virtual data source was created. For example, FoodmartDataSource. • For schema-based data sources within a virtual data source, the name is in the form dataSourcePrefix_schema, where dataSourcePrefix is the data source prefix defined when the virtual data source was created. For example, FoodmartDataSource_public.
datasourceTableName	String	(Required) Literal name of the table in the data source.

8.1.7.2.2 Child Elements

Element Name	Description
<**fieldList**>	(Required) A container for the field elements in the table. Each jdbcTable element can contain only one fieldList element.
<**filterString**>	Expression that evaluates to true or false when applied to each row of values in the data source. For a table, the expression refers to columns using the field_name form of the column ID. See **8.1.10, "Representing Pre-filters in XML," on page 282** for more information.

8.1.7.3 fieldList

fieldList is a container for one or more field elements.

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.

8.1.7.3.1 Child Elements

Element Name	Description
<**field**>	(Required) A column from the table specified in **jdbcTable**.

8.1.7.4 field

The field element represents a column in the data source. The column must be from the table specified by jdbcTable. You must reference all the columns you use in the Domain, including columns that are used to construct the Domain but are not exposed to the end user.

8.1.7.4.1 Attributes

Attribute	Type	Description
fieldDBName	String	Literal name of the column in the data source.
id	String	(Required) Identifier for the column. 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. The id XML attribute can contain alphanumeric characters along with any combination of the following: @#$^`_~? It cannot start with a digit. When generated via the Domain Designer, id is a representation of the column name in the data source, where unsupported characters replaced by underscores.
type	String	(Required) The Java type of the column, as determined from the data source by the JDBC driver. The available types are shown below.

8.1.7.4.2 Supported Types

The following Java types are supported for the type attribute:

```
java.lang.Boolean
java.lang.Byte
java.lang.Character
java.lang.Double
java.lang.Float
java.lang.Integer
java.lang.Long
java.lang.Short
java.lang.String
java.math.BigDecimal
java.sql.Date
java.sql.Time
java.sql.Timestamp
java.util.Date
```

Unless you know the name and type of every column in the data source, it is often easier to select the tables you want in the Domain Designer and then export the XML design file. The Domain Designer accesses the data source to find the names of all tables and columns, as well as their types. You can then modify the exported 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.

8.1.8 Representing Derived Tables in XML

To represent derived tables, use the jdbcQuery element. This element is very similar to the **jdbcTable** element for tables, but contains an additional query element, used to represent the query for the derived table. jdbcQuery is a child of resources.

8.1.8.0.1 Table Hierarchy

The following hierarchy is used for jdbcQuery elements.

```
<jdbcQuery>
        <fieldList> (1)
                <field> (1...n)
        <filterString> (0...n)
        <query> (1)
```

8.1.8.1 jdbcQuery

The jdbcQuery element represents a derived table that results from an SQL query. jdbcQuery is similar to the jdbcTable element for a table, but has an additional child, query, used for the derived table query. jdbcQuery is a child of resources.

8.1.8.1.1 XML Attributes

Attribute	Type	Description
datasourceId	String	(Required) Identifier for the data source, as set in the id attribute of **jdbcDataSource**. 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.
id	String	(Required) Unique identifier for the derived table in the Domain design file. The id of a derived table is used everywhere the id of jdbcTable is used. If you copy a derived table to join it multiple times, each copy must have a different id. The id XML attribute can contain alphanumeric characters along with any combination of the following: @#$^`_~? It cannot start with a digit.

8.1.8.1.2 Child Elements

Element Name	Description
<**fieldList**>	(Required) A container for the **field** elements in the derived table. A jdbcQuery element can contain only one fieldList element.

Element Name	Description
<**filterString**>	Expression that evaluates to true or false when applied to each row of values in the data source. For a derived table, the expression refers to columns using the field_name form of the column ID. See **8.1.10, "Representing Pre-filters in XML," on page 282** for more information.
<**query**>	(Required) The SQL query sent to the database server.

8.1.8.2 fieldList

The fieldList is a container for the field elements in the table. 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.

8.1.8.2.1 Child Elements

Element Name	Description
<**field**>	(Required) A column from the table specified in **jdbcQuery**.

8.1.8.3 field

The field element represents in the results of the query. A field element of a derived table must reference a column that is returned by the query. Only the columns represented by a field element are available for reference by other elements.

8.1.8.3.1 Attributes

Attribute	Type	Description
id	String	(Required) 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. The id XML attribute can contain alphanumeric characters along with any combination of the following: @#$^`_~? It cannot start with a digit.
type	String	(Required) The Java type of the column, as determined from the data source by the JDBC driver. The available types are shown in **Supported Types**

8.1.8.4 query

The query element contains the SQL query that is 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. SQL queries for a derived table are with respect to the JDBC driver for the data source. The syntax for a valid SQL query does *not* include a closing semi-colon.

You can also use JasperReports Server attributes in queries. See **8.1.13, "Using JasperReports Server Attributes in Design Files," on page 294** for more information.

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.

In many cases, the easiest way to create a derived table in the XML design file is to create the initial derived table in the Domain Designer and then export the file. When you create a derived table in the Domain Designer, it runs the query and generates columns based on the result set. These columns are present in the export file.

8.1.8.4.1 Example

The following sample query in PostgreSQL selects some columns from the result of a join and orders the results. One of the columns includes a field calculated in the SQL.

Only fields selected in the query – in this case, exp_date, store_id, amount, currency, conv, and as_dollars – can be exposed as columns of the derived table. Fields not selected in the query cannot be referenced in the Domain design.

```
<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.

8.1.8.5 Using Derived Tables

A derived table provides an alternate way to create joins and calculated fields. Here are some things to keep in mind when deciding how to implement the Domain:

- Calculated fields created within derived tables may use any function call recognized by the RDBMS. Calculated fields created in the Domain using the dataSetExpression attribute of the field element are limited to the functions available in the DomEL language. See **"The DomEL Syntax" on page 297** for more information.
- 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.

8.1.9 Representing Joins in XML

A join is represented in the design file as a `jdbcTable` element. A join tree is a child of the `resources` element.

The `jdbcTable` element is also used to represent tables from the data source. See **8.1.7, "Representing Tables in XML," on page 267**.

8.1.9.1 Element Hierarchy for a Join Tree

The following hierarchy is used to represent a join tree.

```
<jdbcTable>
       <fieldList> (1)
               <field> (1...n)
       <filterString> (0...n)
       <joinInfo> (1)
       <joinList> (1)
               <join> (1...n)
       <joinOptions> (0...1)
       <tableRefList> (1)
               <tableRef> (1...n)
```

8.1.9.2 jdbcTable

`jdbcTable` represents a join tree, that is, the results of one or more joins between tables. There is one `jdbcTable` element for each join tree. `jdbcTable` is a child of `resources`.

`jdbcTable` can also be used to represent a table. See **8.1.7, "Representing Tables in XML," on page 267** for more information.

8.1.9.2.1 XML Attributes

Attribute	Type	Description
`id`	String	(Required) Unique identifier for the join tree 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. The `id` XML attribute can contain alphanumeric characters along with any combination of the following: @#$^`_~? It cannot start with a digit.

Attribute	Type	Description
datasourceId	String	(Required) Alias that identifies the data source for the Domain. 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.
schemaAlias		Name of the database schema for the first table in the join tree. Required if the data source uses schemas. • For schemaless data sources, such as MySQL, schemaAlias is not required. • For schema-based data sources, such as PostgreSQL and Oracle, the name is the literal name of the schema in the data source, for example, public. • For schemaless data sources within a virtual data source, the schema name is of the form dataSourcePrefix, where dataSourcePrefix is the data source prefix defined when the virtual data source was created. For example, FoodmartDataSource. • For schema-based data sources within a virtual data source, the name is in the form dataSourcePrefix_schema, where dataSourcePrefix is the data source prefix defined when the virtual data source was created. For example, FoodmartDataSource_public.
datasourceTableName	String	(Required) Literal name of the first table in the join tree.

If you use the jdbcQuery element to define derived tables, you may have additional join trees 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 columns you want to reference elsewhere in the Domain.

8.1.9.2.2 Child Elements

Element Name	Description
<**fieldList**>	(Required) A container for the field elements in the join tree. A jdbcTable element can contain only one fieldList element.
<**filterString**>	Expression that evaluates to true or false when applied to each row of values in the data source. For a join tree, the expression refers to columns using the table_ID.field_name form of the column ID. See **8.1.10, "Representing Pre-filters in XML," on page 282** for more information.

Element Name	Description
<**joinInfo**>	(Required) Gives the table ID and alias for the table specified by the `schemaAlias` and `datasourceTableName` attributes. 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.
<**joinList**>	(Required) Container for the `join` elements. A `jdbcTable` element can contain only one `joinList` element. The left join in the first `join` in the `joinList` must be the table specified by the `schemaAlias` and `datasourceTableName` attributes of `jdbcTable`.
<**joinOptions**>	Specifies options for the join tree that influence how the joins that are pushed down to SQL in an Ad Hoc view.
<**tableRefList**>	(Required) Container for the list of tables and aliases used in the join. Must include a `tableRef` tag for every table used. A `jdbcTable` element can contain only one `tableRefList` element.

8.1.9.3 fieldList

`fieldList` is a container for the `field` elements in the join tree. `fieldList` is a child of the `jdbcTable` element.

8.1.9.3.1 Child Elements

Element Name	Description
<**field**>	(Required) Represents a column in the join tree.

8.1.9.4 field

`field` represents a column in the join tree. When you export Domain from 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 required. This includes fields that may not appear directly in the Domain, such as fields used in joins. `field` is a child of the `fieldList` element.

`field` elements are also used to represent calculated fields. Calculated fields still appear as children of `fieldList`, but their usage and attributes are different. See **8.1.11, “Representing Calculated Fields in XML,” on page 283** for more information.

8.1.9.4.1 XML Attributes

Attribute	Type	Description
id	String	(Required) Identifier for the field, 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. The id XML attribute can contain alphanumeric characters along with any combination of the following: @#$^`_~? It cannot start with a digit.
type	String	(Required) The Java type of the column, identical to the type in its table definition.

8.1.9.5 joinInfo

joinInfo gives the table ID and alias for the table specified by the schemaAlias and datasourceTableName attributes of the **jdbcTable** element. This table is used as the first table in the join definition. The table named in joinInfo does not have a tableRef element.

Both attributes of joinInfo are required even if they are identical to the values in jdbcTable.

8.1.9.5.1 Attributes

Attribute	Type	Description
alias	String	(Required) Alias for the table identified in referenceId; used in the expression attribute of the join element. 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 join element that defines the join.
datasourceId	String	(Required) Alias for the data source for the Domain. 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.

8.1.9.6 joinList

<joinList> is a container for join elements. There is a join element for each join in the join tree. The left attribute of the first join element in joinList must be the table specified by the schemaAlias and datasourceTableName attributes of jdbcTable.

8.1.9.6.1 Child Elements

Element Name	Description
join	(Required) Representation of a join statement.

8.1.9.7 join

The join element represents a single SQL join statement. The left and right tables, join type, join expression, and optional weight are specified as attributes. Every join in the join tree must have a separate join element.

left="*left_table_ID*" right=*right_table_ID*" type="*join_type*" expr="*expression*"

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.

8.1.9.7.1 Attributes

Attribute	Type	Description
left	String	(Required) Identifier for the first table in the join definition. This table must have been declared previously in the jdbcTable element. The declaration can appear either in the <joinInfo> element or in another join element that precedes this join element in the joinList.
right	String	(Required) Identifier for the second table in the join definition.
join_type	String	(Optional, defaults to inner). One of inner, rightOuter, leftOuter, or fullOuter. fullOuter is not supported in MySQL.
expression	String	Expression that compares the columns on which the join is made. See **Options for the expression Attribute** for more information.
weight	Integer > 0	Integer that specifies a weight to use when calculating the minimum join path. 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. When suppressCircularJoins is set to true, weight is ignored.

Options for the expression Attribute

The expression XML attribute defines the expression used to compare 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, store.store_id == 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="(store.store_id == employee.store_id) AND
  (store.coffee_bar != store.salad_bar)"
  ```

  ```
  expr="(employee.employee_id == salary.employee_id) AND (salary.overtime_paid
  &gt; 2)
  AND (salary.overtime_paid &lt;= 2.1)"
  ```

You have to use the character entities for less than (&lt;) and greater than (&gt;)

- 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="(store.region_id == region.region_id) AND (store.store_city IN ('San
    Francisco','Portland', 'Seattle'))"
    ```

    ```
    expr="(store.store_id == employee.store_id) AND (NOT store.coffee_bar IN
    ('true'))"
    ```

  - a range of values, separated by a colon. For example:

    ```
    expr="(employee.employee_id == salary.employee_id) AND (NOT employee.salary IN
    (7000 : 8000))"
    ```

- joins between a table foreign key and a constant value. Must be used in conjunction with ==. For example:

  ```
  expr="(employee.employee_id == salary.employee_id) AND (employee.salary ==
  5000)"
  ```

Some join expressions, such as NOT, can not be edited in the Domain Designer. However they can be displayed. When you import the Domain into the Domain Designer, these joins are shown in read-only format. See **7.3.5.3.7, "Read-Only Joins," on page 226** for more information.

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.

8.1.9.8 joinOptions

The `joinOptions` element 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, which sets the options for the entire join tree. `joinOptions` lets you avoid circular joins in cases where there are N tables with N or more joins. See **7.3.5.4.3, "Join Tree Options," on page 227** for more information.

8.1.9.8.1 Attributes

Attribute	Type	Description
suppressCircularJoins	Boolean	(Default = false) When this attribute is set to true for a join tree, an Ad Hoc view created from the join tree uses a minimum join. When suppressCircularJoins is set to true, you can optionally assign weights to the joins in the join set. When suppressCircularJoins is set to false, all joins in the join tree are used in Ad Hoc views. 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 of the Domain syntax.
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. This attribute can only be set to true when suppressCircularJoins is false. For most situations, the best practice is to set this option to false (default).

8.1.9.9 tableRefList

The <tableRefList> element is a container for the list of tables and aliases used in the join tree. It must include a tableRef element for every table in the join tree.

8.1.9.9.1 Child Elements

Element Name	Description
<**tableRefList**>	(Required) Represents a table used in the join tree.

8.1.9.10 tableRef

The <tableRef> element represents a table used in the join tree. There must be a tableRef element for every table in the join tree.

8.1.9.10.1 Attributes

Attribute	Type	Description
tableId	String	The ID of a table within the design.
tableAlias	String	Alternative name for the table used within the join expression.

Attribute	Type	Description
`alwaysIncludeTable`	Boolean	(Default = `false`) Setting this to `true` forces this table to be included in all Ad Hoc views created from this data island. For most situations, the best practice is to set this option to `false` (default).

8.1.9.11 Example Join Tree

The following example shows a join tree with three joins between three tables. This example shows how you might use the following:

- To avoid circular joins, `suppressCircularJoins` is set to true.
- The join between `region` and `customer` has a join `weight` of 2. This makes it a less desirable join than the other joins in the tree.
- `alwaysIncludeTable` is set to `true` for the region table.

```
<schema xmlns="http://www.jaspersoft.com/2007/SL/XMLSchema" version="1.3">

<resources>

   <jdbcTable id="JoinTree_1" datasourceId="FoodmartDataSourceJNDI" schemaAlias="public" data-
sourceTableName="customer">
     <fieldList>
       <field id="customer.account_num" fieldDBName="account_num" type="java.lang.Long" />
    ...
     <fieldList>
          <joinInfo alias="store" referenceId="store" />
          <joinOptions suppressCircularJoins="true"/>

          <tableRefList>
            <tableRef tableId="store" tableAlias="store"/>
            <tableRef tableId="customer" tableAlias="customer"/>
            <tableRef alwaysIncludeTable="true" tableId="region" tableAlias="region"/>
          </tableRefList>

         <joinList>
            <join left="store" right="customer" type="inner"
                         expr="store.region_id == customer.customer_region_id"/>
            <join left="store" right="region" type="inner"
                         expr="store.region_id == region.region_id"/>
           <join weight="2" left="region" right="customer" type="inner"
                         expr="region.region_id == customer.customer_region_id"/>
         </joinList>

   </jdbcTable>

</resources>

</schema>
```

8.1.9.12 Working With Joins

Keep the following in mind when working with join syntax:

- 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 in the `jdbcTable` element. 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.

8.1.9.13 Tips for Using Joins

- 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/>
```

8.1.10 Representing Pre-filters in XML

Pre-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 pre-filter determines which rows are available when running the report.

8.1.10.1 filterString

`filterString` contains an expression that evaluates to true or false when applied to each row of values in the data source. The parent of `filterString` is one of the following:

- a **`jdbcTable`** element for a table
- a **`jdbcTable`** element for a join
- a **`jdbcQuery`** element

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 297**.

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 these filters are not editable in the Domain Designer.

8.1.10.1.1 Examples

The following is an example of `filterString` used in a `jdbcTable` element representing a table:

```
<jdbcTable datasourceId="SugarCRMDataSource" id="opportunities"
               schemaAlias="public" datasourceTableName="opportunities">
       <fieldList>
               ...
               <field id="opportunity_type" type="java.lang.String"/>
       </fieldList>
       <filterString>opportunity_type == 'Existing Business'</filterString>
</jdbcTable>
```

The following is an example of `filterString` used in a `jdbcQuery` element:

```
<jdbcQuery datasourceId="SugarCRMDataSource" id="p1cases">
       <fieldList>
               ...
               <field id="status" type="java.lang.String"/>
       </fieldList>
       <filterString>status != 'closed'</filterString>
       <query>...</query>

</jdbcQuery>
```

8.1.11 Representing Calculated Fields in XML

Calculated fields are defined as `field` elements with an additional XML attribute, `dataSetExpression`, that holds the calculation for the field.

8.1.11.1 field

`field` is a child of the `fieldList` element. The location of the `fieldList` element for calculated fields depends on the location of the fields used in the calculation:

- Calculated fields where all the columns in the calculation come from a single table appear twice in the design file:
  - Under the `jdbcTable` element for the join tree that contains the table.
  - Under the `jdbcTable` or `jdbcQuery` element for that table.
- Calculated fields that use columns from different tables appear only once, under the `jdbcTable` element for the join tree contains both tables. Remember, it is not possible to create calculated fields from columns in different join trees.
- Constant fields appear under `resources` as a `null` element.

  Constant fields are calculated fields that do not reference any column names. Constant fields are defined as `field` elements in a `fieldList` under the `null` element. Because constant fields are not dependent on any column values, they may be used in any join tree or data island. They can also be used in other calculated fields and pre-filters.

8.1.11.1.1 XML Attributes

Attribute	Type	Description
dataSetExpression	String	(Required) Expression that calculates a value. This value is either based on other columns or is a constant value. For a description of the syntax for the expression, including how to reference columns, see **“The DomEL Syntax” on page 297**. For information on using JasperReports Server attributes in calculated fields, see **8.1.13, “Using JasperReports Server Attributes in Design Files,” on page 294**.
id	String	(Required) Identifier for the calculated field. The format of the id depends on how the calculated field appears in the design file: 1. If all columns used in the expression are from the same table, and the table appears in a join, there are two field elements for the calculated field, with different ids: • When the field appears under the jdbcTable or jdbcQuery element for the table, the id is a simple column name field_name. • When the field appears under the jdbcTable element for the join tree that uses the table, the id has the form table_ID.field_name. 2. If the expression references columns from different tables, the field appears only in the join tree that contains those tables and the id has the form jointree_ID.field_name. 3. If the expression is a constant value, the field appears in the null element under resources and the id is constant_fields_level.field_name. The id XML attribute can contain alphanumeric characters along with any combination of the following: @#$^`_~? It cannot start with a digit.
type	String	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 **8.1.7.4.2, “Supported Types,” on page 270**.

8.1.11.1.2 Example

The following example shows the XML for a calculated expression that only references one table, 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"
               schemaAlias="public" datasourceTableName="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"
               schemaAlias="public" datasourceTableName="anything">
 <fieldList>
  ...
  <field dataSetExpression="concat( accounts.billing_address_city, ', ',
      accounts.billing_address_state )" id="accounts.city_and_state"
      type="java.lang.String"/>
  ...
 </fieldlist>
</jdbcTable>
```

8.1.11.2 null

The `null` element is a container for constant fields. `null` is a child of the **8.1.6.1, “resources,” on page 266** element.

8.1.11.2.1 Child Elements

Element Name	Description
<**fieldList**>	(Required) A container for the `field` elements for constant fields. The `null` element can contain only one `fieldlist` element.

8.1.11.3 fieldList

As a child of `null`, `fieldList` is a container for one or more `field` elements that represent constant fields.

8.1.11.3.1 Child Elements

Element Name	Description
<field>	(Required) A constant field. Constant fields may be used in any join tree or data island. They can also be used in other calculated fields and pre-filters.

8.1.11.3.2 Example

The following example shows the XML for two constant calculated fields.

```
<resources>
  <null id="constant_fields_level" datasourceId="FoodmartDataSourceJNDI">
    <fieldList>
      <field id="CF_Constant" dataSetExpression="1 + 1 + 1" type="java.math.BigInteger" />
      <field id="CF_Constant_2" dataSetExpression="12" type="java.math.BigInteger" />
    </fieldList>
  </null>
...
</resources>
```

8.1.12 Representing Sets and Items in XML

The hierarchy of data islands, sets, and items are defined through the `dataIslands` and `itemGroups` elements and their children. These two elements, which are defined together, are direct children of the top-level `schema` element.

8.1.12.1 Presentation Hierarchy

The following hierarchy is used to represent data islands, sets, and items. This hierarchy is at the top level, directly under the `schema` element.

```
<schema>
       <dataIslands> (0...1)
               <itemGroup> (1...n)
       <itemGroups> (1...n)
               <itemGroup> (0...n)
                       <itemGroups> (0...n)
                               <itemGroup> (0...n)
                                       <itemGroups> (0...n)
                               <items> (0...n)
                                       <item> (1...n)

                                        ...
                       <items> (0...n)
                               <item> (1...n)
               <items> (0...n)
                       <item> (1...n)
       <items> (0...n)
               <item> (1...n)

...
</schema>
```

The `itemGroups` and `items` elements are recursive elements that are equivalent to the sets and items on the Data Presentation 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 **“Properties” on page 236**.

8.1.12.2 dataIslands

The `dataIslands` element lists all the data islands on the Data Presentation tab. A schema can have at most one `dataIslands` element. If there are no data islands in the presentation, the `dataIslands` element is not required.

Each data island in the Domain must be represented by an `itemGroup` child of the `dataIslands` element. `dataIslands` is a child of `schema`.

8.1.12.2.1 Child Elements

Element Name	Description
<**itemGroup**>	(Required) As a child of `dataIslands`, each `itemGroup` represents a data island.

8.1.12.3 itemGroup

As a child element of `dataIslands`, `itemGroup` represents a data island. Each `itemGroup` in the `dataIslands` element corresponds to an `itemGroups` element that describes the hierarchy of sets and items in the data island. There can be at most one data island for each join tree or unjoined table.

`itemGroup` is also used to represent a set in the Domain presentation. See **8.1.12.5, “itemGroup,” on page 288** for more information.

8.1.12.3.1 XML Attributes

Attribute	Type	Description
`id`	String	(Required) Unique identifier for the data island in the Domain design file. The `id` XML attribute can contain alphanumeric characters along with any combination of the following: @#$^`_~? It cannot start with a digit. The `id` attribute is used by other elements in the presentation representation to identify the data island, via the `referenceId`. In XML files exported from Domain Designer, the `id` attribute corresponds to the data island's **ID** property on the Data Presentation tab.
`label`	String	The data island's name, visible to users of the Domain. If the label is missing, the Ad Hoc Editor displays the `id`. Corresponds to **Label** on the Data Presentation tab.
`description`	String	A description of the set, visible to users as a tooltip on the set name in the Ad Hoc Editor. Corresponds to **Description** on the Data Presentation tab.
`labelId`	String	The internationalization key for the label in the Domain’s locale bundles. Corresponds to **Label Key** on the Data Presentation tab. Can contain alphanumeric characters along with any of the following: .`~_
`descriptionId`	String	The internationalization key for the description in the Domain’s locale bundles. Corresponds to **Descriptions Key** on the Data Presentation tab. Can contain alphanumeric characters along with any of the following: .`~_
`resourceId`	String	(Required) `resourceId` of the `jdbcTable` or `jdbcQuery` element for the join tree or unjoined table that corresponds to the data island. jdbcQuery is only used in the case when your data island comes from a derived table that isn't joined to anybody.

8.1.12.4 itemGroups

itemGroups – A container for itemGroup elements and/or items elements. Each instance of itemGroups corresponds to a data island defined as an itemGroup in dataIslands.

All descendents of a specific itemGroups element must refer to the same join tree or unjoined table. This means that the resourceId attribute of all descendents of an itemGroups element must reference the id attribute of the same jdbcTable or jdbcQuery.

You can upload a Domain with an empty itemGroups element, but for the corresponding data island to be used, it must contain at least one itemGroup or items element.

A recursive relationship between itemGroups and itemGroup is used to show sets and subsets. The top-level element in a set hierarchy is always an itemGroups element. Every itemGroups element must contain at least one items or itemGroup element.

8.1.12.4.1 Child Elements

Element Name	Description
<**itemGroup**>	A set in the data presentation.
<**items**>	A container for item elements.

8.1.12.5 itemGroup

As a child of the itemGroups element or itemGroup element, itemGroup represents a set or subset. itemGroup contains itemGroups elements and/or items elements. Set properties are represented as attributes.

A recursive relationship between itemGroups and itemGroup is used to show sets and subsets. The top-level element in a set hierarchy is always an itemGroups element. Every itemGroups element must contain at least one items or itemGroup element.

itemGroup is also used to represent a data island in dataIslands. See **8.1.12.3, “itemGroup,” on page 287** for more information.

8.1.12.5.1 Child Elements

The hierarchy of itemGroups and items elements defines the hierarchy of sets and items in the data island.

Element Name	Description
<**itemGroups**>	A container for itemGroup elements, which represent sets.
<**items**>	A container for item elements, which represent items.

8.1.12.5.2 XML Attributes

The attributes of itemGroup specify the properties of the set or data island. See **7.3.7.5, “Properties,” on page 236** for more information.

Attribute	Type	Description
id	String	(Required) The unique identifier of the set among all set and item IDs. The id XML attribute can contain alphanumeric characters along with any combination of the following: @#$^`_~? It cannot start with a digit. The id attribute is used by other elements in the presentation representation to identify the set. In XML files exported from Domain Designer, the id attribute corresponds to the set's **ID** property on the data presentation tab.
label	String	The set's name, visible to users of the Domain. If the label is missing, the Ad Hoc Editor displays the id. Corresponds to **Label** on the Data Presentation tab.
description	String	A description of the set, visible to users as a tooltip on the set name in the Ad Hoc Editor. Corresponds to **Description** on the Data Presentation tab.
labelId	String	The internationalization key for the label in the Domain's locale bundles. Corresponds to **Label Key** on the Data Presentation tab. If blank, the default key is used. Can contain alphanumeric characters along with any of the following: .`~_
descriptionId	String	The internationalization key for the description in the Domain's locale bundles. Corresponds to **Descriptions Key** on the Data Presentation tab. If blank, the default key is used. Can contain alphanumeric characters along with any of the following: .`~_
resourceId	String	(Required) id of the join tree or unjoined table that contains this set, as specified in dataIslands. The resourceId attribute must be the same for all item and itemGroup elements that descend from the same itemGroups element.

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 **"Localizing Domains" on page 302**.

8.1.12.6 items

The items element is a container for item elements, which represent items on the Data Presentation tab. items can be a child of itemGroup or schema.

- As a child of schema, items is a container for items in a single data island that do not belong to any a set. All item elements in an items element must belong to the same data island. There is one items element in schema for each data island that has items that are not in a set.
- As a child of itemGroup, items is a container for the items in the set represented by itemGroup.

8.1.12.6.1 Child Elements

Element Name	Description
<**item**>	An item in the data presentation.

8.1.12.7 item

The item element represents an item in the data presentation of the Domain. An item is the representation of a database field or a calculated field along with the display name and formatting properties defined in the Domain.

8.1.12.7.1 Attributes

The attributes of item specify the properties of the item. See **7.3.7.5, "Properties," on page 236** for more information.

Attribute	Type	Description
id	String	(Required) The unique identifier of the item among all set and item IDs. The id XML attribute can contain alphanumeric characters along with any combination of the following: @#$^`_~? It cannot start with a digit. By default, the Domain Designer uses the name of the field as the id. If there are multiple instances of the field in the data island, then the Domain Designer appends a number to the name, such as field2.
label	String	The item's name, visible to users. If the label is missing, the Ad Hoc Editor displays the id.
description	String	An optional description of the item, visible as a tooltip on the item name in the Ad Hoc Editor.
labelId	String	The internationalization key for the label in the Domain's locale bundles. Can contain alphanumeric characters along with any of the following: .`~_
descriptionId	String	The internationalization key for the description in the Domain's locale bundles. Can contain alphanumeric characters along with any of the following: .`~_

Attribute	Type	Description
resourceId	String	(Required) A reference to the column on which the item is based. This defines the connection between what the user sees and the corresponding data in the data source. The precise format depends on the location of the column in the data structure: • When the item refers to a column in an unjoined table, resourceId has the form table_id.field_ID. • When the item refers to a column in a join tree, resourceId has the form jointree_id.table_ID.field_name. In the Domain Designer, table_id and jointree_id can be set using the **Rename...** option from the context menu on the Joins tab. The table_id or jointree_id component of the resourceId attribute must be the same for all item and itemGroup elements that descend from the same itemGroups element.
dimensionOrMeasure	Field or Measure	Corresponds to the **Field or measure** setting in the user interface. Its possible values are Dimension (equivalent to field) or Measure. This attribute is optional and necessary only when overriding the default behavior, for example to make a numeric item explicitly not a measure. By default, all numeric fields are treated as measures in the Ad Hoc Editor.
defaultMask	See table	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 **Table 8-1 on page 292** for the data formats available for each type.
defaultAgg	See table.	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. See **Table 8-2 on page 292** for the possible summary functions based on the column type. The **Appearance** column shows the equivalent setting in the Data Presentation tab.

Labels and descriptions may contain any characters, but the ID property value of both itemGroup and item elements must be alphanumeric.

Table 8-1 Data Formats for Aggregation by Type

Field Type	Attribute Value	Appearance

Integer	`#,##0`
	`0`
	`$#,##0;($#,##0)`
	`#,##0;(#,##0)`
		`-1,234`
		`-1234`
		`($1,234)`
		`(1234)`

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
All others	*Not allowed*	

Table 8-2 Default Summary Functions By Type

Attribute Value	Appearance

`Max`
`Min`
`Average`
`Sum`
`CountDistinct`
`CountAll`
	Maximum
	Minimum
	Average
	Sum
	Distinct Count
	Count All

8.1.12.8 Examples

For example, suppose you have a join tree, JoinTree_1, with the following hierarchy:

- A set labeled Accounts
- A subset of Accounts, with the label Account Address
- A single element, Account Creator, that is in the join tree, but is not in any set.

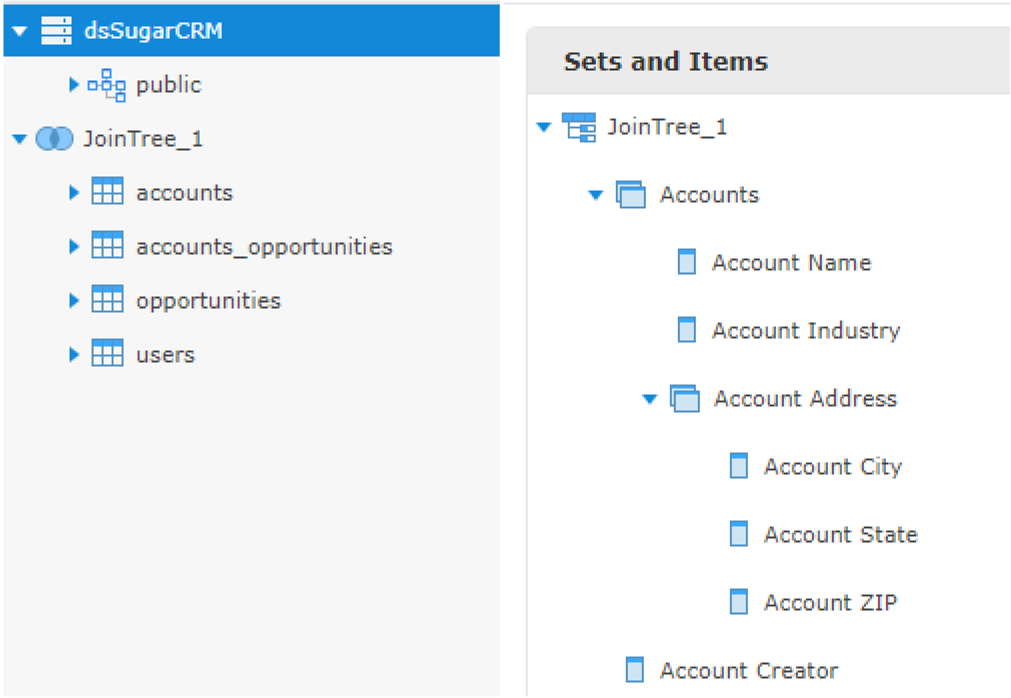


Figure 8-1 Example of a Join Tree Hierarchy

The XML for this join tree might look like this:

```
<dataIslands>
  <itemGroup id="JoinTree_1" resourceId="JoinTree_1" />
</dataIslands>
<itemGroups>
  <itemGroup id="AccountsSet" label="Accounts" resourceId="JoinTree_1">
    <itemGroups>
      <itemGroup id="AddressSubset" label="Account Address" resourceId="JoinTree_1">
        <items>
          <item id="billing_address_city" label="Account City" resourceId="JoinTree_1.ac-
counts.billing_address_city" />
          <item id="billing_address_state" label="Account State" resourceId="JoinTree_1.ac-
counts.billing_address_state" />
          <item id="billing_address_postalcode" label="Account ZIP" resourceId="JoinTree_1.ac-
counts.billing_address_postalcode" />
        </items>
      </itemGroup>
    </itemGroups>
    <items>
      <item id="name" label="Account Name" resourceId="JoinTree_1.accounts.name" />
      <item id="industry" label="Account Industry" resourceId="JoinTree_1.accounts.industry" />
    </items>
  </itemGroup>
</itemGroups>
<items>
    <item id="created_by" label="Account Creator" resourceId="JoinTree_1.accounts.created_by" />
</items>
```

8.1.13 Using JasperReports Server Attributes in Design Files

An JasperReports Server attribute is a name-value pair associated with a user, organization, or server. To define or modify JasperReports Server attributes, see the section on defining attributes in the *JasperReports Server Administrator Guide*.

You can use attributes in the Domain design file. When you use an attribute, the server derives the value at run time, based on the user, organization, or server. Attributes are supported in DomEl expressions in the following locations:

- Calculated field calculations
- Join expressions
- Pre-filter expressions

They can also be used in other locations in the Domain design, including:

- Schema entry keys
- Derived table queries

If you want to use a JasperReports Server attribute in a data source, you must specify the attribute when you create the data source, not in the Domain design file. See the *JasperReports Server Administrator Guide* for information about using attributes in data sources.

8.1.13.1 Attribute Syntax

When referring to an attribute, you can specify the attribute categorically or hierarchically:

- Categorical reference – If you specify a category for the attribute value, the server attempts to find that particular value of the attribute. You can specify these attribute categories:
    - User – In the attributes defined on the logged-in user.
    - Tenant – In the attributes defined on the organization of the logged-in user.
    - Server – In the attributes defined at the server-level.
- Hierarchical reference - If you don't specify a category for the attribute, the filter searches attributes hierarchically and uses the value for the first attribute it finds with the given name. This search starts with the logged-in user, then proceeds to the user's organization and parent organization, and finally to the server level.

For join expressions, calculated fields, and pre-filters, the DomEL syntax is used. See **8.2, "The DomEL Syntax," on page 297** for more information.

- `attribute('attributeName', 'server')` for a server-level attribute.
- `attribute('attributeName', 'tenant')` for an organization-level attribute.
- `attribute('attributeName', 'user')` for a user-level attribute.

If no level is specified, the server will search for the attribute hierarchically, starting at the `'user'` level:

- `attribute('attributeName')`

For schema names and derived tables, the attribute must be escaped with curly brackets ({}):

- `{attribute('attributeName'), 'server'}` for a server-level attribute.
- `{attribute('attributeName', 'tenant')}` for an organization-level attribute.
- `{attribute('attributeName', 'user')}` for a user-level attribute.
- `{attribute('attributeName')}` to search for an attribute hierarchically.

For more information on attributes, see the section "Managing Attributes" in the *JasperReports Server Administrator Guide*. For information about using JasperReports Server attributes in Domain security files, see the *JasperReports Server Security Guide* and *Jaspersoft OLAP Ultimate Guide*.

8.1.13.2 Examples

To use a JasperReports Server attribute as the name of a schema in the data source, use the attribute syntax in the `string` element for the `entry` for that schema. You can use a constant or an attribute for the XML `key` attribute of `entry`.

```
<jdbcDataSource id="FoodmartDataSourceJNDI">
  <schemaMap>
    <entry key="defaultSchema">
      <string>public</string>
    </entry>
    <entry key="_attribute_schemaAttribute_">
      <string>{attribute('schemaAttribute')}</string>
    </entry>
  </schemaMap>
</jdbcDataSource>
```

For pre-filters, JasperReports Server attributes can appear in the expression for the `filterString` element in the `jdbcTble` or `jdbcQuery` element:

```
<jdbcTable id="region" datasourceId="FoodmartDataSourceJNDI" schemaAlias="public" data-
sourceTableName="region">
  <fieldList>
    <field id="region_id" type="java.lang.Integer" />
    <field id="sales_city" type="java.lang.String" />
    <field id="sales_country" type="java.lang.String" />
  </fieldList>
  <filterString>sales_country == attribute('attributeCountry')</filterString>
</jdbcTable>
```

For calculated fields, JasperReports Server attributes can appear in `dataSetExpression` in the `field` element:

```
<field id="Sample_Calculated_Field" dataSetExpression="salary_paid &gt; attribute('numericAttribute')"
type="java.lang.Boolean" />
```

For derived tables, JasperReports Server attributes can appear in the `query` element in `jdbcQuery`. For example, if you have defined an attribute with the name of the table you want called `tableAttribute`, a derived table that uses that attribute might look like this:

```
<jdbcQuery id="inventory_fact_1998_2" datasourceId="dsFoodMart">
      <fieldList>
        <field id="warehouse_cost" type="java.math.BigDecimal" />
        <field id="warehouse_profit" type="java.math.BigDecimal" />
        <field id="warehouse_sales" type="java.math.BigDecimal" />
      </fieldList>
      <query>select

   {attribute('tableAttribute')}.warehouse_sales as warehouse_sales,
   {attribute('tableAttribute')}.warehouse_cost as warehouse_cost,

  ( {attribute('tableAttribute')}.warehouse_sales -  {attribute('tableAttribute')}.warehouse_cost) as
warehouse_profit

from
  {attribute('tableAttribute')}</query>
    </jdbcQuery>
```

You can use attributes in the secondary joins in a complex join `expression`. However, you can't use attributes in the first join in the expression:

```
<joinInfo alias="account" referenceId="account" />
<joinList>
    <join expr="account.account_id == expense_fact.account_id and
       account.account_description == attribute('tableAttribute')"
       left="account" right="expense_fact" type="inner" weight="1" />
</joinList>
<joinOptions />
```

8.1.13.3 Attribute Casting Functions

A JasperReports Server attribute value is always a string. You can use the following functions to cast it to other types in DomEl expressions, 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';

8.1.13.4 Things to Remember When Using Attributes

Note the following when using attributes:

- The JasperReports Server attribute value must match the expected data type. For example, if you are creating a calculated field of type Integer, the attribute value must be an Integer.
- Attribute collections are not supported in the main Domain design. The JasperReports Server attribute must be a single value. Attribute collections can be used in Domain security, however.
- If you are using JasperReports Server attributes for the name of a schema, make sure that any dependent tables and columns in the Domain will be available for every possible value of the attribute. See **7.6.1, "Changing the Data Source," on page 249** for more information.

If you use an attribute in your data source definition, you must ensure that referenced schemas, tables, and columns will be available. Data source attributes are defined outside the Domain Designer. See the section on defining attributes in the *JasperReports Server Administrator Guide* for more information.

- If a specified attribute does not exist, any item that uses it will fail. For example, if you specify the attribute `Country` for a schema name, and a `Country` attribute does not exist anywhere on the server, you will be unable to load the schema. This is similar to referring to column that does not exist.
- If an attribute exists but is not defined (empty) for a particular user, the behavior depends on where the attribute is used. An empty attribute can occur, for example, if you have set up a `Country` attribute, but the current user has no value defined for `Country`.
    - For pre-filters, an empty attribute is interpreted based on the field's data type, as described below.

  - For a field of type String, an empty attribute is interpreted as an empty string: `""`
  - For a Boolean field, and empty attribute is interpreted as `FALSE`.
  - For numeric and date fields, an error will occur.
- For derived table queries and custom joins, the attribute will fail in most cases. However, in some cases, JasperReports Server will consider the resulting expression to be valid, with potentially unexpected results, so be cautious when using attributes in this part of your design.
- For schema names, the default schema is used when the attribute is empty. The default schema must be defined in the XML Domain design file. See **8.1.5.5.1, "Default Schema," on page 264** for more information.

8.2 The DomEL Syntax

A DomEL expression is a shorthand way of writing a complex query. Many 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 `IN` clause for custom joins
- Calculated fields
- Filter expressions in Domains and Domain topics (equivalent to `where` clauses)
- JasperReports Server attribute values (see **8.1.13, "Using JasperReports Server Attributes in Design Files," on page 294**)
- Row-level security (see **8.4, "The Domain Security File," on page 308**)

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.

8.2.1 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'

Simple Type	Description	Example of Constant
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)`

8.2.2 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 `id` attributes defined 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.
Join expression	`table_id.field_name`	Within a join expression, tables are given alias names that must be used.
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.

Appears In	Field Reference	Explanation
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 **Label** entered with double quotes (`"`). Item labels are created on the Data Presentation 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.

8.2.3 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`	
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.

Operator	Syntax	Description
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.

8.2.4 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 **7.4, "Derived Tables and Calculated Fields," on page 241** 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.

8.2.5 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.

8.2.6 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')
```

8.2.7 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
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.
Pre-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).

8.3 Localizing Domains

You can configure Domains to provide localized text for the labels and descriptions of presentation elements (data islands, sets, and items). Text you localize appears in Ad Hoc views and reports built from your Domains, based on the locale specified in the user's operating environment.

8.3.1 Properties File Format

To localize Domain labels and descriptions, you must create a properties file for each locale you want to support. You can also create an optional "default" file that is used as a fallback whenever a message is missing in the properties file for the current locale.

Domain localization in JasperReports Server is implemented using Java locales and resource bundles. A single properties file is sometimes called a "locale bundle."

Each .properties file is a text file containing key-value pairs for a locale:

- The keys, when present, are the same across all the .properties files for the supported locales. A file can omit some of the keys.
- By default, the keys are based on IDs in the presentation, but you can create your own keys if you prefer.
- The values are the labels and descriptions in the language of the target locale.
- File names are of the form <any_name>_<locale>.properties, where:
  - <any_name> is arbitrary and the same for all files.
  - <locale> is any Java-compliant locale identifier, for example `fr` or `fr_CA`.
  - You can optionally include a default .properties file, which is used for the default locale or when a key is missing or empty in one of the other .properties files. The default properties file name is <any_name>.properties. It does not have a language extension.

8.3.2 Keys and Values

Inside the properties files, each entry is of the form:

```
key=value
```

8.3.2.1 Key Names

You can use the default key names generated by JasperReports Server or you can assign your own names to the label and description keys. Key names must satisfy the following rules:

- Each key name must be unique among all key names in the Domain.
- Key names can only use characters from the ISO Latin-1 set, digits, and underscores (_).

In the Domain Designer, you can define key names using the **Label Key** and **Descriptions Key** properties on the Data Presentation tab. In the XML design file, these correspond to the `labelId` and `descriptionId` attributes on the `itemGroup` or `item` element.

If no name has been assigned to the label key or description key, JasperReports Server expects key names in a hierarchical format, for example:

```
<dataIslandID>.LABEL
<dataIslandID>.DESCR
<dataIslandID>.<setID1>...<setIDn>.LABEL
<dataIslandID>.<setID1>...<setIDn>.DESCR
<dataIslandID>.<setID1>...<setIDn>.<item>.LABEL
<dataIslandID>.<setID1>...<setIDn>.<item>.DESCR
```

8.3.2.2 Values

Values are strings that specify the text to use for the locale associated with the file. Values must satisfy the following rules:

- Everything after the = symbol is part of the translated text.
- The properties files use the ISO-8859-1 (Latin-1) encoding that supports most, but not all, accented characters.
- For characters that are not in ISO-8859-1, use Unicode escape sequences. For example, the œ character has the Unicode escape sequence `\u0153`.

You can define values in several places. JasperReports Server looks for values in the following order, and displays the first non-empty value it finds:

1. In the properties file for the current user's locale.
2. In the default properties file.
3. In the **Label** or **Descriptions** property on the Data Presentation tab in the Domain Designer. In the XML design file, these correspond to the `label` and `description` attributes on the `itemGroup` or `item` element.
4. If no value is found in any of the previous locations, then:
    - For a label, JasperReports Server uses the **ID** from the Data Presentation tab, which is the same as the value of the `id` property in the XML file.
    - For a description, the result is left blank.

8.3.3 Examples

8.3.3.1 Example of File Naming for Locales

The following table shows an example relationship between file names, key names, and values in two files, a default file in English and a file of French translations.

Language	Locale	File Name	Sample Contents

English	default	Labels.properties	`JOINTREE_1.SET1.STORE_CITY.LABEL=City`
			`JOINTREE_1.SET1.STORE_CITY.DESCR=Store city`

French	fr	Labels_fr.properties	`JOINTREE_1.SET1.STORE_CITY.LABEL=Ville`
			`JOINTREE_1.SET1.STORE_CITY.DESCR=Ville du magasin`

8.3.3.2 Example of Empty Properties File

The following example shows part of a typical properties file created from the Domain Designer, with internationalization keys generated automatically. Sets and items in the file appear in the same order as on the Data Presentation tab of the Domain Designer. If you have already entered values for the Label Key or Descriptions Key for an item, the value is filled in.

```
JOINTREE_1.ACCOUNTS.LABEL=
JOINTREE_1.ACCOUNTS.DESCR=
JOINTREE_1.ACCOUNTS.NAME.LABEL=
JOINTREE_1.ACCOUNTS.NAME.DESCR=
JOINTREE_1.ACCOUNTS.ACCOUNT_TYPE.LABEL=
JOINTREE_1.ACCOUNTS.ACCOUNT_TYPE.DESCR=
JOINTREE_1.ACCOUNTS.INDUSTRY.LABEL=
JOINTREE_1.ACCOUNTS.INDUSTRY.DESCR=
...
```

8.3.3.3 Example of a Localized File

The following example shows the French translation for the properties file fragment 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.

```
JOINTREE_1.ACCOUNTS.LABEL=Comptes
JOINTREE_1.ACCOUNTS.DESCR=Détails des comptes clients.
JOINTREE_1.ACCOUNTS.NAME.LABEL=Client
JOINTREE_1.ACCOUNTS.NAME.DESCR=Nom du client
JOINTREE_1.ACCOUNTS.ACCOUNT_TYPE.LABEL=Type
JOINTREE_1.ACCOUNTS.ACCOUNT_TYPE.DESCR=Type du compte client.
JOINTREE_1.ACCOUNTS.INDUSTRY.LABEL=Industrie
JOINTREE_1.ACCOUNTS.INDUSTRY.DESCR=Industrie d\u2019activité primaire.
...
```

8.3.4 Creating and Maintaining Translations

Creating and editing translation files for Domains can be a complicated process that often happens incrementally. The best strategy for you depends on your deployment. The following sections describe a common workflow that is useful for most situations.

You can create and edit the properties file in any text editor. Some editors provide features for editing properties files. There are also Java translation editors that manage all of the properties files together, allowing you to translate in parallel and find missing keys and missing translations.

8.3.4.1 Creating Properties Files

1. If you know you will be translating, create your labels and descriptions in a properties file instead of in the Domain Designer. This makes it easier to create and edit additional properties files in the future. You can do this even if you are not translating at this time. See **8.3.2.2, "Values," on page 303** for more information on evaluation order of values.

   If you have no current need for translation, you can create your labels and descriptions within the Domain Designer. If you later need to translate, you can create key names and properties file as you need them.

2. Decide whether you want to explicitly assign keys for each element, or if you want to use generated keys.
3. If you are using generated keys, give the data islands and sets in your presentation meaningful **ID**s. It is easier to work with keys such as `ACCOUNTS_MASTER.ACCOUNTS.LABEL` rather than `JOINTREE_1.SET1.ACCOUNTS.LABEL`.
4. Once you have finalized the naming of the keys or **ID**s, use the **download the template** link on the Options tab in the Domain Designer to generate a file with the keys you want. See **8.3.5, "Exporting a Properties File From the Domain Designer," on page 306** for more information.
5. Using the downloaded file as a starting point, create a file with blank values as a template for your properties files. If you did not explicitly defined any labels or definitions the downloaded file is automatically blank. If you have explicitly defined values, you need to remove them to create a file with blank values.
6. Create a master file with the strings for your base locale. Set this as the default properties file.

   Make sure to create a default properties file. If every file has a locale code in the filename, the values will not appear for users in unsupported locales.

   If you prefer, you can define labels and descriptions in the Domain Designer, which display in unsupported locales. However, this makes it harder to create and maintain a blank version of the properties file.

7. For each additional locale you want to support, copy the blank file and change the `_<locale>` designator to create a basis for the translated file. For information about file naming, see **8.3.1, "Properties File Format," on page 302**.
8. Use specialized software or a translation service to create the Java localization properties files for each locale you want to support.
9. Use the Options tab in the Domain Designer to attach the properties files to the Domain. See **7.3.8, "The Options Tab," on page 237** for more information.

8.3.4.2 Maintaining Properties Files

Once you have the files you will need to update them as keys are added or changed, or as you add locales. Make sure to save a blank properties file or the default locale bundle as a template for future translations.

To add or change keys:

1. Make sure to save a blank properties file or the default locale bundle as a template for future translations.

2. If you change the Domain design, export the locale bundle again and compare it to the template to find all new keys.
3. If you have defined descriptions and keys in the Domain design, delete the exported values to create a new blank template.
4. Translate the new keys and add them to your properties files for your supported locales.
5. Use the Options tab in the Domain Designer to attach the properties files to the Domain. See **7.3.8, “The Options Tab,” on page 237** for more information.

To add or change a locale:

1. Make sure to save a blank properties file or the default properties file as a template for future translations.
2. Copy the blank properties file and change or add the `_<locale>` designator to create a properties file for the new locale. For information about file naming, see **8.3.1, “Properties File Format,” on page 302**.
3. If you have defined descriptions and keys in the Domain design, delete the exported values to create a new blank template.
4. Translate the new keys and add them to your properties files for your supported locales.
5. Use specialized software or a translation service to create the Java localization properties files for each locale you want to support.
6. Use the Options tab in the Domain Designer to attach the properties files to the Domain. See **7.3.8, “The Options Tab,” on page 237** for more information.

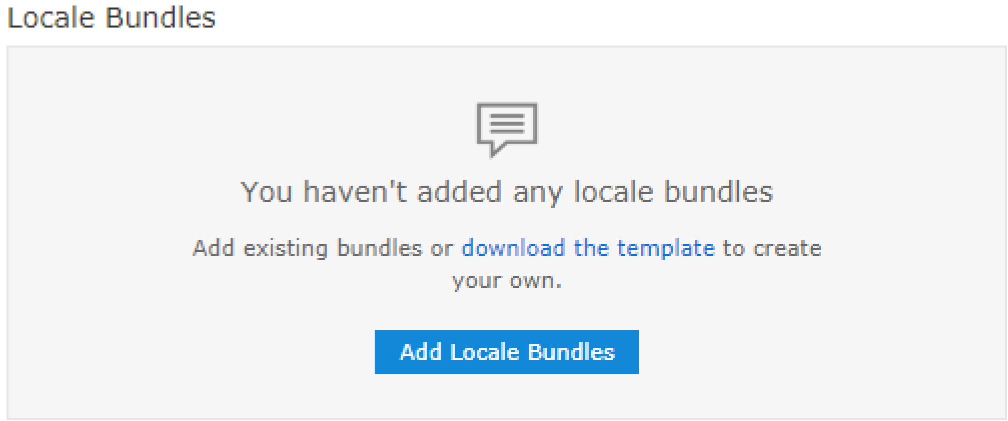
For additional information about locales and properties files (also called locale bundles) in JasperReports Server, see the Localization chapter in the *JasperReports Server Administrator Guide*.

8.3.5 Exporting a Properties File From the Domain Designer

In the Domain Designer, the name of the internationalization keys are defined by the **Label Key** and **Descriptions Key** properties on each set and item on the Data Presentation tab. By default, these properties are blank. You can 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 (_).

You can export a properties file from the Domain Designer.

1. Click **download the template** on the Options tab of the Domain Designer.



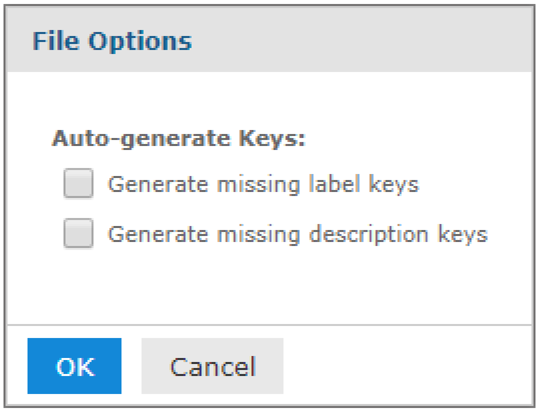
Locale Bundles

You haven't added any locale bundles

Add existing bundles or download the template to create your own.

Figure 8-2 Template Download Link on Options Tab

2. In the File Options dialog box, select which keys to export:
   - **Generate missing label keys** – Automatically generates label key names for any elements where **Label Key** is not defined.
   - **Generate missing description keys** – Automatically generates description key names for any elements where **Description Key** is not defined.
   - When both options are deselected, keys are not generated automatically. Only keys explicitly defined as **Label Key** or **Descriptions Key** are exported.



File Options

Auto-generate Keys:

□ Generate missing label keys

□ Generate missing description keys

Figure 8-3 File Options for Downloading a .properties File

3. Click **OK**.

   The bundle is exported with a generated file name such as "domain xxx.properties". You can change the name to match the name you want to use for your properties files.

   The file contains the following:
   - All explicitly defined keys.
   - Optional generated keys, if selected. For the format of generated keys, see **8.3.2.1, "Key Names," on page 302**.
   - Any values that are explicitly defined as the **Label** or **Description** field for an element on the Presentation tab, or the `labelId` or `descriptionId` attributes on an `itemGroup` or `item` element in the XML design file.

   The exported file is a Java properties file in the proper format containing the selected keys. If you did not set any label or descriptions in the Domain Designer or XML design file, then the file contains blank values and is ready for translation.

If the design file does not load properly in the Domain Designer, you are not 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.

8.3.6 UTF-8 Support In Domains

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.

IDs have the following character restrictions:

- All UTF-8 characters are allowed in IDs except for the following symbols, which 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.
- 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.

8.4 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. In Domains, columns display the items in the Domain; rows display the values of each item. A user can see results only where they have both column- and row-level access. When a user is designing a report in the Ad Hoc Editor, they see only the columns to which they have access. When the report runs, portions to which the user has no access are blank.

User access for a Domain is defined in a single security file. The file is attached to the Domain as a resource, as described in **"The Options Tab" on page 237**. The security file references the `ids` of tables, columns, sets, and items in the Domain design file. 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 **Understanding Domain Design Files**.

If you modify the Domain, you should also export the design file and update the security file with any IDs that have changed. Then attach the updated security file to the Domain using the **Replace Security File** option on the Options tab of the Domain Designer.

For the structure and syntax of the security file, see "Domain and Security Recommendations" in the *JasperReports Server Security Guide*.

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.

Amazon Web Services (AWS)

Cloud platform, used to provide and host a family of services, such as RDS, S3, and EC2.

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.

CloudFormation (CF)

Amazon Web Services CloudFormation gives developers and systems administrators an easy way to create and manage a collection of related AWS resources, provisioning, and updating them in an orderly and predictable fashion.

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.

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.

JasperReports IO

An HTTP-based reporting service for JasperReports Library that provides a REST API for running, exporting, and interacting with reports and a JavaScript API for embedding reports and their input controls into your web pages and web applications.

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 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.

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

Depending on the context:

- In JasperReports Server, the repository is 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.
- In JasperReports IO, the repository is where all the resources needed to create and run reports are stored. The repository can be stored in a directory on the host computer or in an S3 bucket hosted by Amazon Web Services. Users access the repository through a file browser on the host machine or through the AWS console.

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.

S3 Bucket

Cloud storage system for Amazon Web Services. JasperReports IO can use an S3 bucket to store files for its repository.

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.

Stack

A collection of Amazon Web Services resources you create and delete as a single unit.

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