



JASPERSOFT OLAP USER GUIDE

RELEASE 5.6

<http://www.jaspersoft.com>

Copyright © 2014 JasperSoft Corporation. All rights reserved. Printed in the U.S.A. JasperSoft, the JasperSoft logo, JasperSoft iReport Designer, JasperReports Library, JasperReports Server, JasperSoft OLAP, JasperSoft Studio, and JasperSoft ETL are trademarks and/or registered trademarks of JasperSoft Corporation in the United States and in jurisdictions throughout the world. All other company and product names are or may be trade names or trademarks of their respective owners.

This is version 0514-JSP56-19 of the *JasperSoft OLAP User Guide*.

TABLE OF CONTENTS

Chapter 1 Introduction to Jaspersoft OLAP	5
1.1 Overview of Jaspersoft's OLAP Tools	5
1.2 Jaspersoft OLAP Information Resources	6
1.3 External Information Resources	6
Chapter 2 Working with OLAP Views	9
2.1 OLAP Tool Bar	10
2.1.1 Edit Display Options	11
2.1.2 Save Buttons	13
2.1.3 Drilling into a Dimension Member	13
2.1.4 Row and Column Display Options	14
2.2 Navigation Table	14
2.3 Drill-through Table	15
2.4 Analyzing Data in an OLAP View	16
2.4.1 Opening an OLAP View	16
2.4.2 Sorting the Display	18
2.4.3 Drilling Through to Details	19
2.4.4 Displaying Charts	20
2.4.5 Exporting Output	21
2.4.6 Showing the MDX Query	21
2.4.7 Saving an OLAP View	22
Chapter 3 Jaspersoft OLAP Administration	23
3.1 Overview for OLAP Administrators	23
3.2 Administering OLAP Views	24
3.2.1 Overview of an OLAP View	25
3.2.2 Creating an OLAP View with a Mondrian Connection	25
3.2.3 Creating an OLAP View with an XML/A Connection	33
3.2.4 Editing an OLAP View	35
3.3 Working with OLAP Settings and Caches	37
3.3.1 Changing OLAP Settings	38
3.3.2 Flushing the OLAP Cache	46
3.3.3 Configuring OLAP Cache Reuse	47
3.3.4 Configuring the XML/A Cache	48

3.4 Working with OLAP Objects in the Repository	50
3.4.1 Working with Data Sources	50
3.4.2 Working with OLAP Schemas	51
3.4.3 Working with Mondrian Connections	53
3.4.4 Working with XML/A Connections	65
3.4.5 XML/A Against JasperReports Servers Hosting Multiple Organizations	70
3.4.6 Working with XML/A Sources	70
3.4.7 Working with Access Grant Definitions	73
3.5 Performance Tuning	78
3.6 Troubleshooting Jaspersoft OLAP	79
3.6.1 Logging	79
3.6.2 Performance	80
3.6.3 Drill-through Behavior for Dimensions with Parent-child Hierarchies	80
3.6.4 XML/A-based Sample Views and Reports Fail	81
3.6.5 404 Error When WebLogic Hosts Jaspersoft OLAP	81
Glossary	83
Index	93

CHAPTER 1 INTRODUCTION TO JASPERSOFT OLAP

JasperReports Server builds on JasperReports 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 providing 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.

The heart of the 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 business intelligence users, Jaspersoft offers Jaspersoft OLAP, which runs on the server. This optional component is described in this user guide.

This chapter includes:

- **Overview of Jaspersoft's OLAP Tools**
- **Jaspersoft OLAP Information Resources**
- **External Information Resources**

1.1 Overview of Jaspersoft's OLAP Tools

Jaspersoft offers a number of tools to help you define multidimensional cubes and explore their data. For end-users, the following tools provide access to OLAP data:

- **Ad Hoc views.** Created with the Ad Hoc Editor, these views can be based on OLAP client connections. 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. You can save an Ad Hoc view as a report in order to edit it in the interactive report viewer, schedule it, or add it to a dashboard.
- **OLAP-based reports.** Reports created in the Ad Hoc Editor, iReport, and Jaspersoft Studio can read data from OLAP data sources. These reports tend to be more static than views based on the same data sources,

but are excellent when data exploration, such as slice or drill-through, yields results you need to distribute widely.

- Jaspersoft OLAP views. 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.
- Jaspersoft ODBO Connect. This separate client user interface enables Microsoft Excel Pivot Tables to access Jaspersoft OLAP and other OLAP servers that support the XML for Analysis (XML/A) Simple Object Access Protocol (SOAP) protocol. It lets users familiar with Excel leverage those skills to explore their OLAP cubes and create pivot tables based on them. Jaspersoft ODBO Connect is described in its own user guide, which is installed with the product.

In addition to these end-user tools, the Jaspersoft OLAP workbench helps data analysts define the measures and dimensions of a cube based on the data in their transactional database. The workbench is described in its own user guide, which is installed with the product.

1.2 Jaspersoft OLAP Information Resources

In addition to this guide and the others installed with JasperReports Server, Jaspersoft provides several sources of information to help extend your knowledge:

- Our Ultimate Guides document advanced features and configuration. They also include best practice recommendations and numerous examples. The guides are available as downloadable PDFs from the documentation link on the [community website](#).
- Our free [Business Intelligence Tutorials](#) let you learn at your own pace, and cover topics for developers, system administrators, business users, and data integration users. The tutorials are available online from Professional Services section of our [website](#).
- Our free samples, which are installed with JasperReports, iReport, Jaspersoft Studio, and JasperReports Server, are documented online:
 - [JasperReports Samples Reference](#)

Click the Docs link on our [community website](#) to find more documentation about all our products.

- Our [community website](#) hosts open source projects, associated source code, tools for bug tracking, version control, and forums for community discussion. You'll find development and implementation advice, a secure development environment for community-driven projects, and community feedback.



The OLAP connections you define in Jaspersoft OLAP can be used to explore data in Ad Hoc views, which retain the dimension and measure metadata defined for your cubes. For more information, refer to the *JasperReports Server User Guide*.

1.3 External Information Resources

As analytic practices permeate the business sector, reference books about OLAP and MDX proliferate. While many focus on Microsoft SQL Server, the language concepts and syntax apply more broadly. The following resources can help you understand analysis:

- *Mondrian Technical Guide*: <http://mondrian.pentaho.com/documentation>
- Jaspersoft OLAP Workbench:
<http://sourceforge.net/projects/jasperserver/files/JasperServer/JasperServer%204.0.0/>

- Historical overview of analysis: <http://en.wikipedia.org/wiki/analysis>
- JPivot SourceForge project: <http://jpivot.sourceforge.net/>
- William Pearson. *MDX at First Glance: Introduction to SQL Server MDX Essentials*.
<http://www.databasejournal.com/features/mssql/article.php/1495511>
- Microsoft MDX reference: <http://msdn2.microsoft.com/en-us/library/ms145506.aspx>
- Wikipedia MDX overview and links: http://.wikipedia.org/wiki/Multidimensional_Expressions
- A collection of other MDX article links: <http://sqlblog.com/blogs/mosha/archive/2005/10/11/mdx-functions-in-analysis-services-2005.aspx>

In addition to these resources, Jaspersoft encourages you to review the *Jaspersoft OLAP Ultimate Guide*.

CHAPTER 2 WORKING WITH OLAP VIEWS

Jaspersoft OLAP is an On Line Analytical Processing (OLAP) application. OLAP applications help users analyze and understand complex data. Jaspersoft OLAP Community Project uses JPivot for analytical processing with an improved user interface.



The procedures in this document assume that you installed JasperReports Server with its sample data. The schemas and views it describes are available only if the samples are installed.

An OLAP view consists of a client connection and an MDX query. A client connection is either a Mondrian connection or an XML/A connection; it provides the data that appears in the view. Jaspersoft OLAP contains a number of sample OLAP views. This guide assumes you are logged in as a user with an administrator role (that is, that you have access to the view and its data).

The following describe Jaspersoft OLAP's analysis capabilities:

- **OLAP Tool Bar**
- **Navigation Table**
- **Drill-through Table**
- **Analyzing Data in an OLAP View**









2.1 OLAP Tool Bar







The tool bar on left side of your view provides access to many OLAP operations and features.



Figure 2-1 OLAP Tool Bar

Table 2-1 OLAP Tool Bar Icons

Icon	Name	Description
	Zoom on Drill	Toggles (that is, turns on or off) the zoom in/out hyperlinks for hierarchy members. See "Edit Display Options" on page 11
	Sort Across Hierarchy	Toggles between sort across and sort within hierarchy. See "Sort Options" on page 12.
	Hide Empty Rows/Columns	Hides or reveals rows or columns that do not have relevant fact data. See "Edit Display Options" on page 11.
	Swap Axes	Changes the orientation of the table by switching the columns and rows.
	Edit Display Options	Allows users to configure the cube options, drill-through options, and sort options. See "Edit Display Options" on page 11.
	Show Chart	Displays a chart of the navigation table data.
	Edit Chart Options	Defines various charting options.
	Change Data Cube	Changes an OLAP view and defines dimension filters.

Icon	Name	Description
	Show MDX Query	Changes the navigation table by editing the MDX query that generates the view. This feature is intended for advanced users familiar with MDX and the data structures underlying the view.
	Export to Excel	Prompts you to view or save the current navigation table in Microsoft Excel format.
	Export to PDF	Prompts you to view or save the current navigation table in Adobe Acrobat PDF format.
	Edit Output Options	Defines various output options.
	Save View	Saves this OLAP view. Changes you've made since you opened the view are saved to the repository. If you don't have permission to save the view in its current location, the Save View As dialog prompts you to select a new location.
	Save View As	Saves this OLAP view under a new name and location. Changes you've made since opening the view are saved to the repository in the location you select. Note that you cannot use the Save View As button to overwrite an existing view (even if you have sufficient permissions).

2.1.1 Edit Display Options

The Display Options dialog lets you control the content and appearance of the information in your view, such as cube options, drill-through options, and sort options, which are described in the following sections.

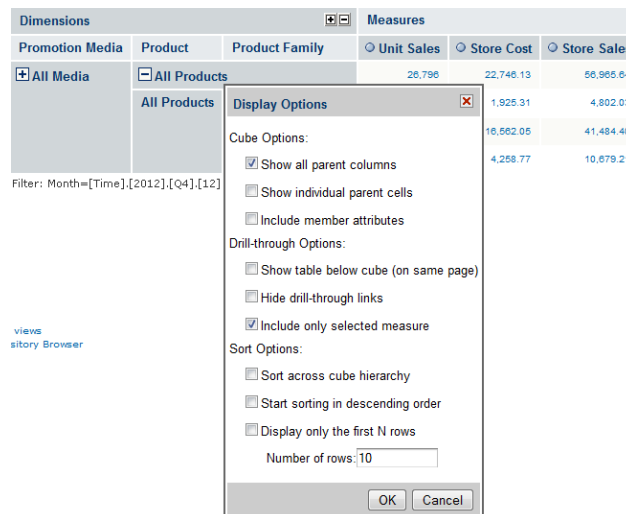
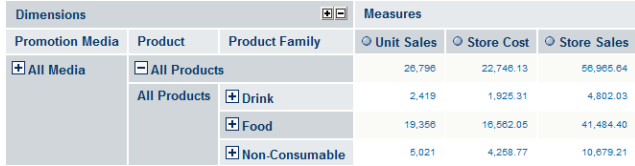
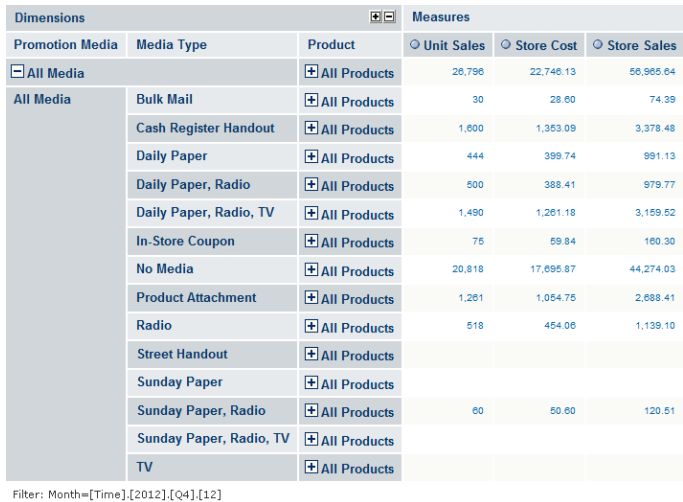


Figure 2-2 Display options Dialog


2.1.1.1 Cube Options

Show all parent columns	Displays the column headings of a given hierarchy. The following navigation table shows Product and Product Family as parent column headings.
	
Show individual parent cells	Displays each parent member of a given hierarchy. The following navigation table displays all parent cells for Promotion Media and Product dimensions.
	
Include member attributes	Displays the member properties of the displayed hierarchy members.

2.1.1.2 Drill-through Operations

Show table below cube (on same page)	Displays the drill-through table below the navigation table. By default, the drill-through table appears in a separate browser window.
Hide drill-through links	Removes the hyperlinks from the fact data in measures.
Include only selected measure	Limits the display to only the selected measure in the drill-through table.

2.1.1.3 Sort Options

The option to sort across a cube's hierarchy is also available in the form of a toolbar button . In either case, it changes the behavior of sorting across or within dimension hierarchies. In the following example, Sort Across

Hierarchy is selected, and the Unit Sales measure is sorted in descending order across the Product hierarchy.

For more information, see the [Jaspersoft OLAP Ultimate Guide](#)

Dimensions			Measures		
Promotion Media	Product	Product Family	Unit Sales	Store Cost	Store Sales
All Media	All Products		26,796	22,746.13	56,955.64
	All Products	Food	19,356	16,562.05	41,484.40
		Non-Consumable	5,021	4,258.77	10,679.21
		Drink	2,419	1,925.31	4,802.03



Filter: Month=[Time].[2012].[Q4].[12]

Figure 2-3 Sorting Across Hierarchy

The Display Options dialog also provides the following options:


Start sorting in descending order	Toggles the sort behavior between ascending and descending.
Display only the first N rows	Limits the number of rows displayed after sorting.

2.1.2 Save Buttons

	Save View	Saves this OLAP view. Changes you've made since you opened the view are saved to the repository. If you don't have permission to save the view in its current location, the Save View As dialog prompts you to select a new location.
	Save View As	Saves this OLAP view under a new name and location. Changes you've made since opening the view are saved to the repository in the location you select. Note that you cannot use the Save View As button to overwrite an existing view (even if you have sufficient permissions).

For more information, see [“Saving an OLAP View” on page 22](#).

2.1.3 Drilling into a Dimension Member


The Zoom on Drill button  allows you to use the zoom in and out hyperlinks for hierarchy members. In the following example, Zoom on Drill is selected. All Media and All Products are hyperlinks, and in this state the user can click them to view the members of their hierarchies.

Dimensions		Measures		
Promotion Media	Product	Unit Sales	Store Cost	Store Sales
 All Media	 All Products	26,796	22,748.13	56,965.64

Filter: Month=[Time].[2012].[Q4].[12]

Figure 2-4 Zoom on Drill toggled on

2.1.4 Row and Column Display Options

The Hide Empty Rows/Columns button  allows you to hide or reveal rows or columns that do not have relevant fact data. The following example includes empty rows for Promotion Media (Street Handout; Sunday Paper; and Sunday Paper, Radio, TV).




 All Media	All Products	 Drink	2,419	1,926.31	4,802.03
No Media	All Products	 Drink	1,906	1,497.47	3,723.05
Cash Register Handout	 All Products		1,600	1,353.09	3,378.48
Daily Paper, Radio, TV	 All Products		1,490	1,261.18	3,159.52
Product Attachment	 All Products		1,261	1,054.75	2,688.41
Radio	 All Products		518	454.06	1,139.10
Daily Paper, Radio	 All Products		500	388.41	979.77
Daily Paper	 All Products		444	399.74	991.13
In-Store Coupon	 All Products		75	59.84	160.30
Sunday Paper, Radio	 All Products		60	50.60	120.51
Bulk Mail	 All Products		30	28.60	74.39
Street Handout	 All Products				
Sunday Paper	 All Products				
Sunday Paper, Radio, TV	 All Products				
	All Products	 Drink			
		 Food			







Figure 2-5 Showing Empty Rows

2.2 Navigation Table

The navigation table appears at the top of the OLAP view (Figure 2-7 on page 17). It shows the data that is retrieved by the current MDX query, which appear in both the main view and in drill-through tables.

Table 2-2 Navigation Table Icons and Options


Icon	Name	Description
	Expand Position	Expands rows at a specific hierarchy member.
	Collapse Position	Collapses rows at a specific hierarchy member.


Icon	Name	Description
	Expand/Collapse Member	Synchronizes the expansion or contraction of rows across all hierarchy members when they are clicked.
	Zoom In/Out	Click hyperlinked hierarchy members to replace the current table with a sub-table that depicts the selected member. This option is only available when Zoom on Drill is enabled.
	Zoom Out All	Restores the navigation table to its initial view after having zoomed. This option is only available when you're in Zoom on Drill mode.
	Natural Order	Located next to measure labels, indicates that the navigation table is sorted according to the order of hierarchy members. Click it to change the sort order.
	Ascending	Located next to measure labels, indicates that the navigation table is sorted according to their numeric value, from smallest to largest. Click it to change the sort order.
	Descending	Located next to measure labels, indicates that the navigation table is sorted according to their numeric value, from largest to smallest. Click it to change the sort order.
	Expand All	Located near the top-left corner of the navigation table, expands all of the currently displayed members (all those that display the plus sign) to the next level of detail in the hierarchies. This can be selected repeatedly to expand all levels of detail. This option is only available when Zoom on Drill is not active. This operation is limited by the memory available to the application server that hosts JasperReports Server. It stops expanding members when this limit is reached.
	Collapse All	Located in the top-left corner of the navigation table, collapses the navigation table to its initial view.

2.3 Drill-through Table

The drill-through table displays supporting details for the selected roll-up value in the navigation table (described in “[Navigation Table](#)” on page 14). The following sections describe the available functionality.

Table 2-3 Drill-Through Table Icons and Options

Icon	Name	Description
	Edit Properties	Displays at the top-left corner of the drill-through table. Selects the columns to display or hide in the drill-through table. The up and down arrows move the columns and specify the number of rows to display per page.

Icon	Name	Description
	Output as CSV	Displayed at the top-left corner of the drill-through table. Prompts you to view or save the current drill-through table in comma-separated values format.
	Expand/Collapse Member	Synchronizes the expansion or contraction of rows across all hierarchy members when they are clicked.
Page 1/14 ►►	First, Previous, Next, Last	Click the arrows to navigate the pages of data.
Goto Page <input type="text" value="1"/> ⌵	Goto Page	Enter the number of the page you want to view and press return to display the page.
Rows/page <input type="text" value="10"/> ⌵	Rows/page	Set the number of rows to display.

2.4 Analyzing Data in an OLAP View

The purpose of data analysis is to uncover relationships and trends in the data. The analysis should give you new insights into

the situation that the data describes. To structure the analysis, you should ask questions like these:

- How did my organization perform this year as compared to last year? Which parts and personnel of the organization did better and which did not?
- For a consumer business, what is my most and least profitable product/customer/salesperson/office/store? Which factor in my data trend in the same direction as the most profitable, which factors trend in the opposite direction, and which are neutral? How do those factors trend for the least profitable?
- For a hospital, for instance, which patients are staying longer than is typical for their diagnosis? What symptoms and other diagnoses do the long-staying patients have?

The questions you can answer with Jaspersoft OLAP depend on:

- The available data.
- The structure of the data in terms of OLAP; that is, the structure of your cubes, dimensions, and measures.
- The starting OLAP view defines which cube you want to analyze and the metrics relevant to a particular need.

OLAP views give business users a starting point for analysis that can then be sliced and diced to answer detailed questions. For a particular OLAP data set, there are usually a number of OLAP views defined as convenient entry points.

As an example, let's answer a specific question: "What is the quarterly sales dollar amount for the snack foods category in 2012 for stores in California?"

2.4.1 Opening an OLAP View

To open an OLAP view:

1. Click **View > Repository**.

The repository appears.

2. Scroll through the repository to select an OLAP view or, type the name (or partial name) of the view you want to see in the search field at the top of the page.

For example, enter `Foodmart`.

The repository reappears, displaying the objects that match your text.

The screenshot shows the JasperReports Repository interface. On the left is a 'Filters' sidebar with categories like 'All available', 'All types', 'Any time', and 'Any schedule'. The main area is titled 'Repository' with a search bar containing 'foodmart'. Below the search bar is a table of results, sorted by Name. The table has columns: Name, Description, Type, Created Date, and Modified Date. The 'Foodmart Sample Analysis View' is highlighted in blue.

Name	Description	Type	Created Date	Modified Date
Foodmart	Foodmart Mondrian Analysis Connection	Mondrian Connection	Today	8/17/2011
foodmart data for crosstab	foodmart data for crosstab	Report	Today	May 1
Foodmart Data Source	Foodmart Data Source	JDBC Data Source	Today	May 1
Foodmart Data Source JNDI	Foodmart Data Source JNDI	JNDI Data Source	Today	8/17/2011
Foodmart Grant	ANALYSIS Foodmart Grant	File	Today	May 1
Foodmart Grant without top level	ANALYSIS Foodmart Grant with no top levels allowed	File	Today	8/17/2011
Foodmart Mondrian Sales Report	Report with Mondrian Datasource	Report	Today	May 1
Foodmart Mondrian XMLA definition	Foodmart Mondrian XMLA definition	Mondrian XMLA	Today	May 1
Foodmart Multi Variable Grant	ANALYSIS Foodmart Multi Variable Grant	File	Today	May 1
Foodmart Multi Variable Grant: top level	ANALYSIS Foodmart Multi Variable Grant. Has top level	File	Today	May 1
Foodmart Multi Variable Grant without top level	ANALYSIS Foodmart Multi Variable Grant. Has no top level	File	Today	May 1
Foodmart Sample Analysis View	Sample Analysis View: 2006 Q1 Foodmart Unit Sales	OLAP View	Today	May 1
FoodmartSchema	Foodmart Analysis Schema	File	Today	8/17/2011
Foodmart Schema Uppercase	Foodmart Analysis Schema Uppercase	File	Today	May 1
Foodmart Variable Grant	ANALYSIS Foodmart Variable Grant	File	Today	May 1
Foodmart XML/A Connection	Foodmart XML/A Connection	XML/A Connection	Today	May 1
NewFoodmart_JNDI_Datasource	JNDI Datasource	JNDI Data Source	Today	8/18/2011
NewFoodmart_JNDI_Datasource	JNDI Datasource	JNDI Data Source	Today	8/18/2011
NewFoodmart_Mondrian_Connection	Mondrian Connection	Mondrian Connection	Today	8/17/2011
NewFoodmart_Mondrian_Connection	Mondrian Connection	Mondrian Connection	Today	8/18/2011

Figure 2-6 Search Results in the Repository

3. To display an OLAP view, right-click it and select **Run**. For example, right-click the **Foodmart Sample Analysis View** and click **Run**.

JasperReports Server displays the view.

The screenshot shows the 'Foodmart Sample Analysis View' in JasperReports Server. On the left is a vertical toolbar with icons for various actions. The main area displays a table with dimensions and measures. The dimensions are 'Promotion Media' and 'Product'. The measures are 'Unit Sales', 'Store Cost', and 'Store Sales'. The table shows data for 'All Media' and 'All Products'. Below the table is a filter: 'Month=[Time].[2012].[Q4].[12]'.

Dimensions		Measures		
Promotion Media	Product	Unit Sales	Store Cost	Store Sales
All Media	All Products	26,796	22,746.13	56,965.64

Filter: Month=[Time].[2012].[Q4].[12]

[Back to OLAP views](#)
[Back to Repository Browser](#)









Figure 2-7 Foodmart Sample Analysis View

4. Click the tool bar buttons and values in the navigation table to explore the data.

2.4.2 Sorting the Display

Sorting allows you to display an ordered view of data.

To sort the display:

1. Open your Analysis View.
2. Click .
3. Click the **Move to Rows** icon  next to the filter you want to use to create a row, then expand it as needed.
4. Click next to **Promotion Media** and **Product** to remove them from the view.
5. Next to the Time filter, click the **Move to Columns** icon  to make Time a column.
6. Click **Time** to open a tree displaying the dimension members.
7. Expand Time by clicking  next to it, and select **2012**, then click **OK**.
8. Click **Measures** in the column section, and deselect **Unit Sales** and **Store Cost**.
9. Click **OK**, then click **OK** again.
10. Click to activate **Zoom on Drill**. When active, its tool bar button looks like this .
11. Click to zoom into USA.
12. Clear **Zoom on Drill** by clicking .
13. Click the navigation table's **Expand All Members** button ..
14. Click the **Edit Display Options** button  and make sure **Show all parent columns**, **Sort across cube hierarchy**, and **Start sorting in descending order** are selected, then click **OK**.
15. Click the navigation table's **Expand All Members** button again.
16. Click the **Sort** button  next to 2012.

The navigation table displays the top stores for the year ordered by dollar sales amount.

Dimensions					Measures						
					Store Sales						
					Time						
					2012	2012					
Store					Q1			Q2			
Store Country					1	2	3	4	5	6	
Store State											
Store City											
Store Name											
All Stores	USA	WA			263,793.22	19,499.70	21,728.80	22,054.36	20,926.37	19,884.51	21,685.76
		CA			159,167.84	11,737.29	12,654.65	11,783.26	13,605.89	11,787.43	13,003.43
		OR			142,277.07	14,302.70	9,675.34	16,192.25	8,346.99	12,784.35	10,642.54
		OR	Salem		87,218.28	9,893.50	4,962.84	11,152.75	4,535.16	7,326.09	5,563.65
			Salem	Store 13	87,218.28	9,893.50	4,962.84	11,152.75	4,535.16	7,326.09	5,563.65
		WA	Tacoma		74,843.96	6,083.25	5,760.48	6,022.02	5,376.22	5,054.03	6,630.91
			Tacoma	Store 17	74,843.96	6,083.25	5,760.48	6,022.02	5,376.22	5,054.03	6,630.91
		OR	Portland		55,058.79	4,419.20	4,712.50	5,039.50	3,810.83	5,468.26	5,078.89
			Portland	Store 11	55,058.79	4,419.20	4,712.50	5,039.50	3,810.83	5,468.26	5,078.89
		CA	Los Angeles		54,545.28	4,167.72	5,570.17	3,999.08	3,917.49	3,396.00	4,370.95
			Los Angeles	Store 7	54,545.28	4,167.72	5,570.17	3,999.08	3,917.49	3,396.00	4,370.95
			San Diego		54,431.14	4,722.87	3,854.69	4,720.27	4,296.07	4,345.29	4,335.92
			San Diego	Store 24	54,431.14	4,722.87	3,854.69	4,720.27	4,296.07	4,345.29	4,335.92
		WA	Bremerton		52,896.30	3,437.91	4,530.33	4,517.53	4,235.60	3,700.04	5,080.67
			Bremerton	Store 3	52,896.30	3,437.91	4,530.33	4,517.53	4,235.60	3,700.04	5,080.67
			Seattle		52,644.07	4,037.00	4,154.51	4,569.13	4,444.06	4,557.58	3,531.22
			Seattle	Store 15	52,644.07	4,037.00	4,154.51	4,569.13	4,444.06	4,557.58	3,531.22
			Spokane		49,634.46	3,244.88	4,606.84	3,778.39	4,109.22	3,809.01	3,717.19
			Spokane	Store 16	49,634.46	3,244.88	4,606.84	3,778.39	4,109.22	3,809.01	3,717.19
		CA	Beverly Hills		45,750.24	2,629.09	2,836.37	2,738.43	5,065.00	3,667.80	3,864.35

Figure 2-8 Sorting the Top West Coast Stores in Sales

2.4.3 Drilling Through to Details

The drill-through operation displays detailed transaction information for a given aggregated value.

To work with source data:

- Using the table created in Sorting the Display, click the number \$87,218.28 next to Store 13.


The drill-through table appears in a new page.

This is the default behavior. You can also display the drill-through table in the same page as the navigation table.

The drill-through table shows the underlying data in the database that is used to generate the summarized information in the navigation table. This is useful for validation of results. Drill-through data can also reveal interesting trends or anomalies. For example, you might identify a particular demographic that tends to make larger purchases.

As shown in the following figure, Carbonated Beverages are among the items making up the Store Sales.

Foodmart Sample Analysis View

 **Drill Through Table for [Store_Sales = 87,218.28]**

Store_Name	Store_Sqft	Store_Type	Year	Quarter	Month	Week	Day	Product_Family	Product_Department	Product_Category
Store 13	27694	Deluxe Supermarket	2012	Q1	1	3	7	Drink	Alcoholic Beverages	Beer and Wine
Store 13	27694	Deluxe Supermarket	2012	Q1	1	3	7	Drink	Alcoholic Beverages	Beer and Wine
Store 13	27694	Deluxe Supermarket	2012	Q1	1	3	7	Drink	Alcoholic Beverages	Beer and Wine
Store 13	27694	Deluxe Supermarket	2012	Q1	1	3	7	Drink	Alcoholic Beverages	Beer and Wine
Store 13	27694	Deluxe Supermarket	2012	Q1	1	3	7	Drink	Alcoholic Beverages	Beer and Wine
Store 13	27694	Deluxe Supermarket	2012	Q1	1	3	7	Drink	Beverages	Carbonated Beverages
Store 13	27694	Deluxe Supermarket	2012	Q1	1	3	7	Drink	Beverages	Carbonated Beverages
Store 13	27694	Deluxe Supermarket	2012	Q1	1	3	7	Drink	Beverages	Drinks
Store 13	27694	Deluxe Supermarket	2012	Q1	1	3	7	Drink	Beverages	Drinks
Store 13	27694	Deluxe Supermarket	2012	Q1	1	3	7	Drink	Beverages	Drinks






Page 1/1,335   Goto Page  Rows/page 





Figure 2-9 Drill-through Table Showing High-value Items

Note that you can export the current set of source data to an Excel spreadsheet by clicking at the top of the drill-through table. You can change the columns that are displayed and their sort order by clicking . Navigate through the paged data and control the number of rows per page using the controls at the bottom of the drill-through table.

2.4.4 Displaying Charts

Charts can provide more dramatic visual impact. Create a simple view to display as a chart.

To view a chart:

1. From the Repository, open the Foodmart Sample Analysis View.
2. Click .
3. Click the **Move to Rows** icon next to the Store filter to create a row, then click **Store**.
4. Expand All Stores, then USA, and select **CA**, **OR**, and **WA**. Click **OK**.
5. Click the filter icon  next to **Promotion Media** and **Production** in the Rows section to remove them from the row.
6. Click **OK**.
7. Click **Collapse All** .
8. Click **Show Chart** .

A default chart output appears.



Default chart for Product Sales Across Store Locations


2.4.5 Exporting Output

OLAP views can be exported to Excel and PDF formats:

- Click **Output as Excel** to export the OLAP view to Excel.
- Click **Print as PDF** to export the OLAP view to a PDF.

2.4.6 Showing the MDX Query

The MDX Query Editor contains the MDX query that retrieves the contents of the navigation table. As you change the content of the navigation table, the MDX query is automatically updated. You can also change the contents of the navigation table by changing the MDX in the editor.

1. Click  to open the MDX Query Editor.

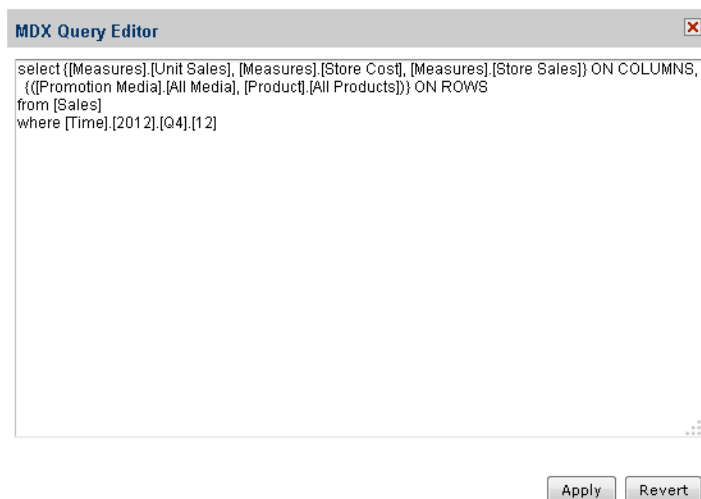


Figure 2-10 MDX Query Editor

An MDX query consists of data sets, query scope, and filter specifications:

- A **SELECT** statement determines the data sets that will populate the columns (x-axis) and rows (y-axis) of the navigation table. The **SELECT** statement includes the measures to use as columns and rows. The query in this example specifies data sets in terms of:
 - [Measures].[Unit Sales], [Measures].[Store Cost], [Measures].[Store Sales] as columns; [Promotion Media].[All Media] and [Product].[All Products] as rows.
 - The **FROM** clause specifies the cube that is queried. You can query only one cube at a time.
 - The **WHERE** clause uses dimensions to constrain the data sets retrieved by the query, that is, the clause specifies the filters that screen the data the query returns. In the example, [TIME].[2012] is the filter.
- 2. Click **Apply** to update the navigation table in the OLAP view. The system validates the query and updates the navigation table.
- 3. Click **Revert** to discard all changes.

For details about the MDX query language, see Microsoft's [Multidimensional Expressions Reference](#).

2.4.7 Saving an OLAP View

After making changes to the navigation table, save the OLAP view as described in [Save Buttons](#). With sufficient permissions, you can either overwrite the existing view or save the view with a new name. When you save, you are prompted for a name, location, and an optional description. If a view with the same name exists in this location, you are prompted to overwrite the existing file. When you save a view with a new name, JasperReports Server displays the new view, rather than the view you initially edited.



Jaspersoft OLAP only saves the drill-through table if it is displayed on the same page as the navigation table. For more information, refer to Jaspersoft OLAP Ultimate Guide.

CHAPTER 3 JASPERSOFT OLAP ADMINISTRATION

Jaspersoft OLAP administration involves repository management and performance tuning. For information about more general administrative tasks, such as access control, see the *JasperReports Server Administrator Guide*.



The **Manage** menu, discussed in some of the following sections, only appears if you have an administrative role, such as `ROLE_ADMINISTRATOR` or `ROLE_SUPERUSER`. In commercial editions, the **Manage > Server Settings** menu item is only available to the superuser account.

This chapter includes:

- **Overview for OLAP Administrators**
- **Administering OLAP Views**
- **Working with OLAP Settings and Caches**
- **Working with OLAP Objects in the Repository**
- **Performance Tuning**
- **Troubleshooting Jaspersoft OLAP**



JasperReports Server's themes can control the look and feel of your OLAP views and interface, such as changing the colors and borders used in the navigation table. For more information, refer to the *JasperReports Server Administrator Guide*.

3.1 Overview for OLAP Administrators

Jaspersoft OLAP lets you analyze data organized into a hierarchical dimensional model, which has cubes and dimensions as its primary entities. In a relational database implementation of online analytical processing (OLAP), the entities reside in relational tables. This is sometimes referred to as Relational OLAP (ROLAP).

Jaspersoft OLAP is based on an open source ROLAP server called Mondrian. JPivot, an open source web-based OLAP user interface, enables users to display and navigate Mondrian's results. Jaspersoft OLAP includes an improved JPivot user interface (as compared to JPivot).

The de facto standard for OLAP query languages is MDX; many analysis applications use it. In a distributed computing environment, XML for Analysis (XML/A) is the standard for accessing OLAP data sources remotely. XML/A uses a web services architecture and transmits MDX queries using the SOAP protocol.



This overview assumes that you rely exclusively on JasperReports Server's built-in OLAP engine. If you rely on a different OLAP engine, such as Microsoft SQL Server Analytic Services, refer to the associated documentation for instructions on maintenance.

To implement and maintain Jaspersoft OLAP:

1. Store cube data in a relational database and use an OLAP schema file to define the cubes. Note that using existing transactional databases may be inefficient with large amounts of data. To improve performance, use an extract, transform, and load (ETL) process: extract information from one or more data sources, integrate and transform the data, and load the result into the separate cube database.
2. Identify facts or measures (the values to summarize) and dimensions (divisions of the measures – for example, dividing time into weeks, months, and years) in the cube database.
3. Define an OLAP schema, mapping logical facts and dimensions onto the physical database. The Jaspersoft OLAP engine uses the schema to interpret the database and perform OLAP queries. The Jaspersoft OLAP Schema Workbench enables you to develop, validate, and test schemas against the database. The [Mondrian Technical Guide](#) gives details of schemas and their options.
4. Create an OLAP client connection that includes your schema and a database connection. Optionally, define data level security for the cube with access grants that limit users to only the data they are allowed to see, based on their roles and user profile information. The repository stores the information you provide to define an OLAP client connection.
5. Define entry points for analysis of the cube data, known as OLAP views. OLAP views enable end users to interact with cubes without having to know query languages, database connections, and other technical details. An OLAP view is an MDX query that is run against the cube as a starting point for interactive analysis. OLAP views appear as objects that users can select through the web interface. You can use the Jaspersoft OLAP Schema Workbench to develop and test MDX queries.
6. If you have defined access grants, test them by using the **Log in As** function to impersonate users with different roles.
7. Plan for updates to data in the cube, usually on a regular basis (for example, nightly or weekly). After an update, flush (empty) the Jaspersoft OLAP data cache so that subsequent queries get the results of new data. For more information, see **“Flushing the OLAP Cache” on page 46**.
8. Tune for performance. As a database administrator, analyze the SQL queries that Jaspersoft OLAP issues against the cube database. Jaspersoft OLAP has built-in tools that enable you to track queries (for information, refer to **“Performance Tuning” on page 78**). Tune the cube database accordingly with tools specific to the database type (for example, indexes and data striping). With large data volumes, you can create pre-calculated tables during the data load (ETL) process to aggregate measure values. Jaspersoft OLAP can use the aggregate tables to replace calculations and thus improve query performance.
9. Design your environment for scalability and availability. Jaspersoft OLAP can run on a single machine. However, for large numbers of users, large data volumes, and high availability, you can use a multi-host environment with load balancers and some machines dedicated to either OLAP user interface or OLAP server duties. You can use XML/A to distribute processing. For more information, refer to **“Working with XML/A Connections” on page 65** and **“Working with XML/A Sources” on page 70**.

The detailed procedures, beginning in section **“Creating an OLAP View with a Mondrian Connection” on page 25**, step you through the process of setting up all components of an OLAP view.

3.2 Administering OLAP Views

This section describes basic administration of OLAP views, including:

- **Overview of an OLAP View**
- **Creating an OLAP View with a Mondrian Connection**
- **Creating an OLAP View with an XML/A Connection**
- **Editing an OLAP View**

3.2.1 Overview of an OLAP View

An OLAP view is a collection of multidimensional data that is based on an OLAP client connection and an MDX query. It is the entry point to analysis operations, such as slice and dice, navigate, and drill-through. End users open these views from the repository once administrators create them. Creating a view entails identifying the elements that allow Jaspersoft OLAP to retrieve and secure the data.

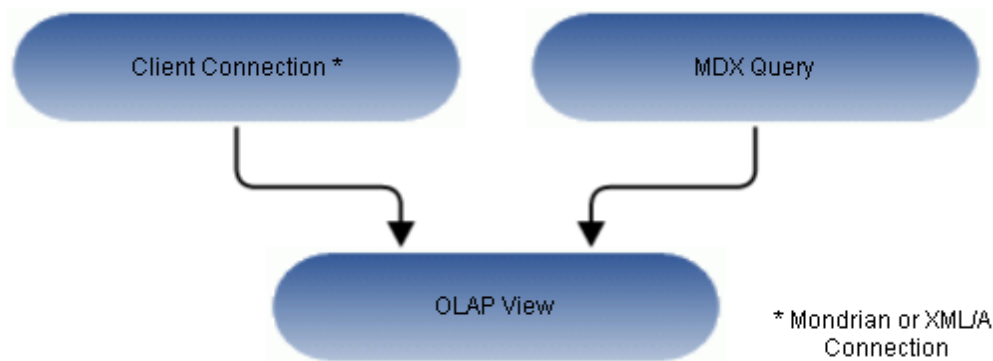


Figure 3-1 Anatomy of an OLAP View

For more information about client connections, refer to **“Editing a Mondrian Connection” on page 60** and **“Editing an XML/A Connection” on page 69**.

3.2.2 Creating an OLAP View with a Mondrian Connection

An OLAP view can retrieve data from a Mondrian connection. For more information on Mondrian connections, refer to **“Editing a Mondrian Connection” on page 60**.

To create an OLAP view with a local Mondrian connection:

1. Click **View > Repository**.
The repository page appears.
2. In the **Folders** panel, navigate to **Organization > Organization > Analysis Components > Analysis Views**.
3. Right-click the folder and select **Add Resource > OLAP View**.
The Name the View page appears and prompts you to provide a name for the new view.

Figure 3-2 Name the View Page

4. Enter a name and description for the new view. The **Resource ID** field is auto-generated when you type in the **Name** field. After it is saved, it can't be changed.
5. Click **Next**.

The Locate Mondrian Client Connection Source page appears and prompts you to select or create a local Mondrian connection.

Figure 3-3 Locate Mondrian Client Connection Source Page

6. Click either:
 - **Define a Mondrian Client Connection in the next step.**
 - **Select a Mondrian Client Connection from the repository.**

Then click **Browse**, navigate to the connection you want, and click **Select**.

7. Click **Next**.

If you chose to define a new Mondrian connection, the Set Connection Type and Properties page appears and prompts you to define a connection.

The screenshot shows a web form titled "Add OLAP Client Connection" with a sub-header "Set Connection Type and Properties". Below the sub-header is a instruction: "First select the type of connection you want to add, then enter the required property values." The form contains a "Type: Connection Type" dropdown menu with "Mondrian" selected. Below this are three text input fields: "Name (required):" with the value "Foodmart1", "Resource ID (required):" with the value "Foodmart1", and "Description:" which is empty. Below these is a "Select folder from repository to save" section with a text input field containing "/analysis/schemas" and a "Browse..." button. At the bottom of the form are three buttons: "Previous", "Next", and "Cancel".

Figure 3-4 Set Connection Type and Properties Page

8. To change the type of the connection, select a connection type from the **Type** drop-down and complete the fields. Otherwise, enter the requested information. For details see **“Creating a Mondrian Connection” on page 54**.
9. To chose a location for the connection, click **Browse**, navigate to a folder, and click **Select**.
10. Click **Next**.

The Locate OLAP Schema page appears and prompts you to upload an OLAP schema or select one from the repository.

The screenshot shows a web form titled "Add OLAP Schema Resource" with a sub-header "Locate OLAP Schema". Below the sub-header are two radio button options. The first option is "Upload a Local File" with a text input field and a "Browse..." button. The second option is "Select a resource from the Repository" with a text input field and a "Browse..." button. At the bottom of the form are three buttons: "Previous", "Next", and "Cancel".

Figure 3-5 Locate OLAP Schema Page

11. Click either:

- **Upload a Local File** to select a file from your local computer.
Then click **Browse**, navigate to select the file you want, and click **Select**.
- **Select a resource from the Repository** to select an existing schema.
Then click **Browse**, navigate to select the schema, and click **Select**.

12. Click **Next**.

The OLAP Schema Resource page appears.

Figure 3-6 OLAP Schema Resource Page

If you chose to upload a new file, the fields are editable. Enter the requested information. For details, refer to **“Working with OLAP Schemas” on page 51**.

13. Click **Next**.

The Locate Data Source page appears and prompts you to create or select a data source.

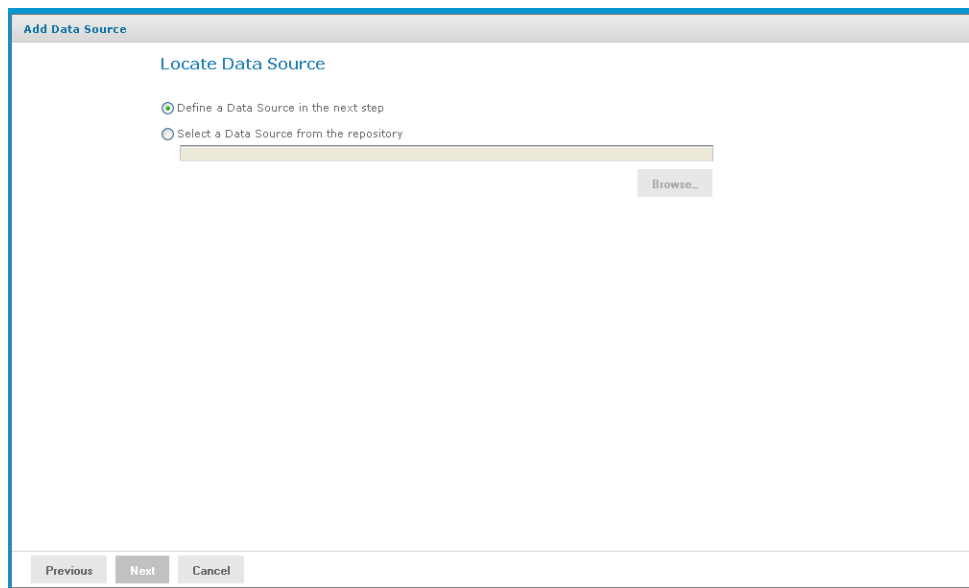


Figure 3-7 Locate Data Source Page

14. Click either:

- **Define a Data Source in the next step** to add a data source.
- **Select a Data Source from the repository** to select a data source from the repository.

Then click **Browse**, navigate to the data source you want to use, and click **Select**. Click **Next** and skip to [step 18](#).

15. Click **Next**.

If you chose to create a new data source, the Set Data Source Type and Properties page appears.

Add Data Source

Set Data Source Type and Properties

First, select the type of data source you wish to add, then enter the required property values.

Type: JDBC Data Source

Name (required):

Resource ID (required):

Description:

Driver (required):

Hint: org.postgresql.Driver

URL (required):

Hint: jdbc:postgresql://localhost:5432/mydb

User Name (required):

Password:

Time Zone:
Use database setting.
Hint: Do not change the time zone setting unless you know the database timestamp data is incorrect.

Save Location:

Figure 3-8 Set Data Source Type and Properties Page

16. Enter the requested information and test the connection. For details, refer to **“Working with Data Sources” on page 50** and to the *JasperReports Server Administrator Guide*.
17. When the test succeeds, click **Next**.

If you use a commercial edition of the server, the Locate Access Grant Definition page appears, prompting you to set the properties for the resource.

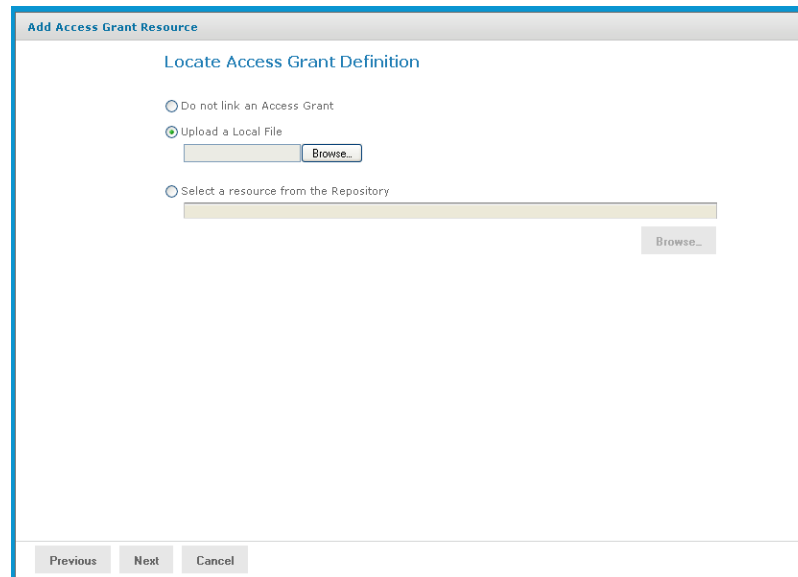


Figure 3-9 Locate Access Grant Definition Page

18. Click one of the following:

- **Do not link an Access Grant** if you don't need to apply data security.
Then click **Next** and skip to step [step 22](#).
- **Upload a Local File** to select a file from your local computer.
Then click **Browse**, navigate to select the file you want, and click **Select**.
- **Select a resource from the Repository** to select an existing schema.
Then click **Browse**, navigate to select the schema, and click **Select**.

In our case, we don't need to secure the data in the view, so won't specify an access grant schema. The next steps show you how to add a file if it's needed.

19. Click **Next**.

The Access Grant Resource page appears.

The screenshot shows the 'Access Grant Definition Details' window. At the top, it says 'Access Grant Resource' and 'Set the properties for the resource.' Below this, a box labeled 'Type: Access Grant Schema' contains the following fields:

- Selected Resource:** FoodmartAccessGrant.xml
- Name (required):** Foodmart Access Grant
- Resource ID (required):** Foodmart_Access_Grant
- Description:** (empty text area)
- Save Location:** /analysis/schemas

At the bottom right of the form is a 'Browse...' button. At the bottom of the window are three buttons: 'Previous', 'Submit', and 'Cancel'.

Figure 3-10 Access Grant Resource Page

20. If you chose to upload a new file from your computer, the fields are editable. Enter the requested information. For details, refer to [“Uploading an Access Grant Schema” on page 77](#). If you chose a file from the repository, the fields aren’t editable.
21. Click **Next**.

The Define the Query page appears and prompts you for an MDX query string.

The screenshot shows the 'Add Query' window. It has a title bar 'Add Query' and a main heading 'Define the Query'. Below the heading, it says 'Select a language and enter the query'. The form contains:

- Query Language:** MDX (selected from a dropdown menu)
- Query String:** (empty text area)

At the bottom of the window are three buttons: 'Previous', 'Submit', and 'Cancel'.

Figure 3-11 Define the Query Page

22. Enter an MDX query. For example, type:

```
select {[Measures].[Unit Sales], [Measures].[Store Cost], [Measures].[Store Sales]}
on columns, {[Promotion Media].[All Media], [Product].[All Products]]} ON rows from
Sales where ([Time].[2012].[Q4].[12])
```

To learn more about writing MDX queries, refer to the reference material listed in **“External Information Resources” on page 6**.

23. Click **Submit**.

If the view passes validation, it is added to the repository. If you receive an error, it is likely that the problem is a typo in your query. Carefully review the query to ensure that it is valid.

24. When you have a valid OLAP view, clicking **Submit** adds it to the repository.

If the view passes validation, it is added to the repository.

3.2.3 Creating an OLAP View with an XML/A Connection

An OLAP view can retrieve data from an XML/A connection. An XML/A connection is a connection to a remote Mondrian client connection. For more information on XML/A connections, refer to **“Working with XML/A Connections” on page 65** and **“Working with XML/A Sources” on page 70**.

To create an OLAP view with an XML/A connection:

1. Click **View > Repository**.

The repository appears.

2. In the Folder panel, navigate to **Organization > Organization > Analysis Components > Analysis Views**.
3. Right-click the **Analysis Views** folder and select **Add Resource > OLAP View** from the context menu.

The Name the View page appears and prompts you to enter the basic details about the new view.

Add OLAP View

Name the View

Provide a name for the OLAP View.

Name (required):

Resource ID (required):

Description:

Figure 3-12 Name the View Page

4. Enter a name and a description of the view and click **Next**.

The Locate Mondrian Connection page appears.

5. In the **Connection Type** drop-down, select XML/A Connection.
6. Click either:
 - **Define a XML/A Client Connection in the next step** to add a new connection.
 - **Select a XML/A Client Connection from the Repository** to select a data source from the repository.

Click **Browse**, navigate to the location where you want to add the file, and click **Select**. Click **Next** and skip to [step 9](#).

7. If you chose to create a new client connection, the Set Connection Type and Properties page appears and prompts you to define the connection. Enter the requested information. For details, refer to [“Working with XML/A Connections” on page 65](#).

Add OLAP Client Connection

Set Connection Type and Properties

First select the type of connection you want to add, then enter the required property values.

Type: Connection Type XML/A

Name (required):

Resource ID (required):

Description:

Catalog (required):

Hint: Foodmart

Data Source (required):

Hint: Provider=Mondrian;DataSource=Foodmart

URI (required):

Hint: http://localhost:8080/jasperserver-pro/xmla

User Name:

Password:

Select folder from repository to save

Figure 3-13 Set Connection Type and Properties - XML/A Page



Your XML/A provider may be another JasperReports Server instance hosting Mondrian connections. For more information, refer to sections [“Working with XML/A Connections” on page 65](#) and [“Working with XML/A Sources” on page 70](#).

8. Click **Next**.

The Define the Query page appears and prompts you for a query string.

Figure 3-14 Define the Query Page

9. In the **Query String** field, enter the MDX query. For example, type:

```
select {[Measures].[Unit Sales], [Measures].[Store Cost], [Measures].[Store Sales]}
on columns, {[Promotion Media].[All Media], [Product].[All Products]]} ON rows from
Sales where ([Time].[2012].[Q4].[12])
```

To learn more about writing MDX queries, refer to the reference material listed in “[External Information Resources](#)” on page 6.

10. Click **Submit**.

If the view passes validation, it is added to the repository. If you receive an error, it is likely that the problem is a typo in your query. Carefully review the query to ensure that it is valid.

11. When you have a valid OLAP view, clicking **Submit** adds it to the repository.

If the view passes validation, it is added to the repository.

3.2.4 Editing an OLAP View

To change the naming, connection, or MDX query in an OLAP view:

1. In the **Search** field in the repository, enter the name (or partial name) of the OLAP view you want to edit, and click the **Search** icon.

For example, enter `food`.

The repository displays the objects that match the text you enter.

2. Right-click a view and click **Edit**. In this example, we’re editing the Foodmart Sample Analysis View.

The Name the View page appears with the fields populated.

Edit OLAP View

Name the View

Provide a name for the OLAP View.

Name (required):
Foodmart Sample Analysis View

Resource ID (read-only):
Foodmart_sample

Description:
Sample Analysis View: 2006 Q1 Foodmart Unit Sales

Previous Next Cancel

Figure 3-15 Name the View Page

3. Make your changes to the fields as necessary and click **Next**.

The page that appears depends on the type of client connection defined in the view. For example, if the view specifies a Mondrian connection, the Locate a Mondrian Connection Source page appears.

Add Mondrian Client Connection

Locate Mondrian Client Connection Source

Connection Type: Mondrian

☐ Define a Mondrian Client Connection in the next step

☒ Select a Mondrian Client Connection from the repository

Browse...

Previous Next Cancel

Figure 3-16 Locate Mondrian Client Connection Source Page

4. Depending on the type of connection specified, enter values as necessary. Click each field you want to change and enter new values. For details, refer to [“Creating an OLAP View with a Mondrian Connection” on page 25](#) and [“Creating an OLAP View with an XML/A Connection” on page 33](#).
5. Click **Next**.

The Define a Query page appears with the query language set to MDX.

Figure 3-17 Define the Query Page

6. Change the query or enter a new one, if necessary.



You can also edit a view's MDX query by modifying the navigation table and saving the view. To learn more about writing MDX queries, refer to the reference material listed in **“External Information Resources” on page 6**.

7. Click **Submit**.

If the view passes validation, it is added to the repository. If you receive an error, it is likely that the problem is a typo in your query. Carefully review the query to ensure that it is valid.

8. When you have a valid OLAP view, clicking **Submit** adds it to the repository.

3.3 Working with OLAP Settings and Caches



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.

This section describes system configuration and maintenance options, including options that control the OLAP engine's behavior and the various caches that can store the multi-dimensional data that populate your views and reports.

This section includes:

- **Changing OLAP Settings**
- **Flushing the OLAP Cache**
- **Configuring OLAP Cache Reuse**
- **Configuring the XML/A Cache**

3.3.1 Changing OLAP Settings

Various configurable properties control the OLAP engine's behavior. In most cases, you can use the default values for these properties. However, if you want to adjust the performance, you may need to change them.



Jaspersoft recommends that you carefully review the effects of the changes you make to the performance tuning variables, and to test such changes before using them in a production environment.

The OLAP settings page requires you to have administrative permissions:

- In JasperReports Server commercial editions, you must be logged in using the superuser account.
- In JasperReports Server Community Project, you must be logged in as a user with the ROLE_ADMINISTRATOR role. Also not that, in this edition, the options are not editable. You must make your changes in a configuration file on the server's host, instead.

To change the OLAP settings:

1. Click **Manage > Server Settings**.
2. Click **OLAP Settings**.

The OLAP Settings page appears.

OLAP Settings Flush OLAP Cache

General Behavior

Performance Profiling Enabled
☒ ji-profiling.enabled Change Cancel

Disable OLAP Memory Caching
☐ mondrian.rolap.star.disableCaching Change Cancel

If disabled, Mondrian will throw an exception during perform a drill through
☒ mondrian.drillthrough.enable Change Cancel

Generate Formatted SQL Traces
☐ mondrian.rolap.generate.formatted.sql Change Cancel

Query Limit
 mondrian.query.limit Change Cancel

Result Limit (number of rows)
 mondrian.result.limit Change Cancel

Maximum number of MDX query threads per Mondrian server instance
 mondrian.rolap.maxQueryThreads Change Cancel

Defines the interval value between polling operations performed by the RolapConnection shepherd thread
 mondrian.rolap.shepherdThreadPollingInterval Change Cancel

Maximum number of passes allowable while evaluating an MDX expression
 mondrian.rolap.evaluate.MaxEvalDepth Change Cancel

Class name of ExpCompiler to use
 mondrian.calc.ExpCompiler.class Change Cancel

MDX identifiers are case-sensitive
☐ mondrian.rolap.case.sensitive Change Cancel

If > 0, number of cells limit that are batched together when building segments
 mondrian.rolap.cellBatchSize Change Cancel

Sibling members are ordered according to their ordinal expression
☐ mondrian.rolap.compareSiblingsByOrderKey Change Cancel

If > 0, Maximum query time (number of seconds)
 mondrian.rolap.queryTimeout Change Cancel

Number of members read when processing high cardinality dimension elements
 mondrian.result.highCardChunkSize Change Cancel

Figure 3-18 OLAP Settings Page, Commercial Editions

Each property is listed with its underlying name (as it appears in the underlying OLAP engine), as well as a more descriptive label. The properties are described in the table below.

3. If your edition of the server includes it, click the **Performance Profiling Enabled** check box to generate performance reports and views, and click **Change**.

The page displays a message indicating that the setting was updated.

4. Locate and analyze the performance reports and views. The reports are found in the repository at /performance/reports. The views are found at /performance/views.
5. Review the available properties described below to determine if they can be changed to improve performance.
6. Adjust any options as needed, click **Change**, and run the performance views and reports to understand the impact your changes made.
7. Adjust the settings as necessary.

Test your views and adjust properties as your findings dictate.



To log the SQL queries Jaspersoft OLAP sends to the database, set the **Generate Formatted SQL Traces** option. This ensures that Jaspersoft OLAP writes the SQL queries it executes to the log file you specify.

The following table lists the properties' meanings.

Table 3-1 OLAP Settings

Property	Notes
General Behavior	
Performance Profiling Enabled	When enabled, performance profiling data is generated and recorded. The availability of this setting is controlled by your license.
Disable OLAP Memory Caching	Turns off caching completely. Disabling caching can have a very noticeable negative performance impact.
If disabled, Jaspersoft OLAP returns an exception when users drill through	If disabled, Jaspersoft OLAP returns an exception if you attempts to drill-through. This can be useful when the underlying data contains confidential information or is so voluminous that drill-through performance would be unacceptable.
Generate Formatted SQL Traces	When tracing is enabled, Jaspersoft OLAP formats SQL queries in the trace output with line breaks, which makes them easier to read.
Query Limit	The maximum number of concurrent queries allowed.
Result Limit (number of rows)	When set to a number greater than 0, result sets are limited to the specified number of rows.
Maximum number of MDX query threads per Jaspersoft OLAP instance	For each Jaspersoft OLAP instance in your deployment, the maximum number of concurrent threads that can be used for MDX queries. For more information about running multiple instances in a single deployment, refer to “Performance Tuning” on page 78 .

Property	Notes
Interval between polling operations performed by the RolapConnection shepherd thread	Specifies the interval between polling operations performed by the RolapConnection shepherd thread. This controls query timeouts and cancellation, so a small value (a few milliseconds) is usually best. When this is set to a value higher than the value defined for the If > 0, Maximum query time (number of seconds) setting, the timeout isn't enforced as expected.
Maximum number of passes allowable while evaluating a MDX expression	When evaluating an MDX query, the maximum number of passes allowed. Jaspersoft OLAP returns an error when this threshold is exceeded; for example, the error may occur during complex calculations.
Class name of ExpCompiler to use	If entered, this must be a Java class name that is an implementation of the mondrian.calc. ExpCompiler interface. Refer to the Mondrian Javadoc for more information.
MDX identifiers are case-sensitive	Specifies whether the MDX parser considers the case of identifiers.
If > 0, the number of cells that are batched together when building segments	When set to a number greater than zero, defines a limit on the number of cells that can be batched together when building segments.
Sibling members are ordered according to their ordinal expression	Specifies whether siblings at the same level of a dimension are compared according to the order key value retrieved from their ordinal expression. By default, ordinal expressions are only used for ORDER BY, and Jaspersoft OLAP ignores the actual values. When this property is enabled, Jaspersoft OLAP can correctly order members when native filtering is used. Note that this setting requires that the RDBMS provides non-null instances of <code>java.lang.Comparable</code> that yield the correct ordering when calling their <code>Comparable.compareTo</code> method.
If > 0, Maximum query time (number of seconds)	When set to a value greater than zero, Jaspersoft OLAP times out if the query takes longer than the specified number of seconds. If a query exceeds the limit, Jaspersoft OLAP returns an error. For more information, refer to the Query Limit and Result Limit entries in this table.
Number of elements read when processing high cardinality dimension elements	This property determines how many members Mondrian reads in one block from the database. Setting large values for this property increases performance but can overload memory. Values should be prime with <code>mondrian.result.limit</code> . For more information, refer to the configuration guide .
Sparse Segment Density Threshold	Performance tuning variable. This property only applies when <code>SparseSegmentCountThreshold</code> is enabled. It determines whether to use a sparse or a dense representation when collections of cell values are stored in memory.

Property	Notes
Sparse Segment Count Threshold (number of cell values)	<p>Performance tuning variable. this property only applies when SparseSegmentDensityThreshold is set. It determines whether a sparse or a dense representation is used when collections of cell values are stored in memory.</p> <p>When storing collections of cell values, Jaspersoft OLAP can use either a sparse or a dense representation. This is determined by the possible and actual number of values: density is calculated as actual / possible.</p> <p>Whenever possible, Jaspersoft OLAP uses a sparse representation - countThreshold * actual > densityThreshold.</p> <p>For example, for the default values (countThreshold = 1000, SparseSegmentDensityThreshold = 0.5), Jaspersoft OLAP uses a dense representation for:</p> <ul style="list-style-type: none"> • 1000 possible, 0 actual, or • 2000 possible, 500 actual, or • 3000 possible, 1000 actual <p>If there are fewer actual values or more possible values, Jaspersoft OLAP uses a sparse representation.</p>
During schema load, invalid members are ignored and will be treated as a null	When enabled, Jaspersoft OLAP ignores invalid members during schema load; they are treated as null members if they are referenced in a later query.
During query validation, invalid members are ignored and will be treated as a null	When enabled, Jaspersoft OLAP ignores invalid members during query validation; invalid members are ignored and are treated as null members.
Defines how a null Member is represented in the result output	Specifies how Jaspersoft OLAP should represent a null member in the result output.
If > 0, the maximum number of iterations allowed when evaluating an aggregate	<p>When set to a number greater than 0, the maximum number of iterations allowed when evaluating an aggregate. When set to 0, iterations are unlimited.</p> <p>If a query exceeds the limit, Jaspersoft OLAP returns an error that specifies this property's value. For more information, refer to the Query Limit and Result Limit entries in this table.</p>
If > 0, crossjoin result limit beyond which the optimizer will be applied (number of rows)	<p>When set to a number greater than 0, specifies a threshold for a crossjoin input list's size; if it exceeds this value, and the axis has the NON EMPTY qualifier, Jaspersoft OLAP uses the non-empty optimizer.</p> <p>When this property is set to 0, Jaspersoft OLAP applies the non-empty optimizer to all crossjoin input lists. To ensure that the optimizer is never applied to crossjoin input lists, set this value to the <code>Integer.MAX_VALUE</code>.</p>
Enable the in-memory rollup of segment data	When enabled (the default setting), segment data are rolled up in-memory.

Property	Notes
If there are unrelated dimensions to a measure in context during aggregation, the measure is ignored in the evaluation context	<p>When dimensions unrelated to a measure are detected during aggregation, the measure is ignored in the evaluation context. Note that this property can only affects measures whose CubeUsage's <code>IgnoreUnrelatedDimensions</code> is false.</p> <p>For more information, refer to the configuration guide.</p>
Do elements of a dimension (levels, hierarchies, members) need to be prefixed with dimension name in the MDX query	<p>Determines if elements of dimension (levels, hierarchies, and members) must be prefixed with the dimension name in MDX queries.</p> <p>This property determines whether certain queries succeed or fail based on the way the dimension elements are defined in the MDX query.</p> <p>When this property is enabled, this query fails:</p> <pre>select {[Omaha]} on columns from cust</pre> <p>When this property is disabled, that same query succeeds.</p> <p>With the property enabled, the correct query is:</p> <pre>select {[Nebraska].[Omaha]} on columns from cust</pre> <p>If your schema is very large or complex, Jaspersoft recommends enabling this setting, as processing such schemas can be very resource-intensive.</p> <p>For more information, refer to the configuration guide.</p>
Division by null or zero produces NULL	<p>By default, when Jaspersoft OLAP attempts to divide by zero or null, it evaluates to Infinity, which is correct for some analysis providers. When this property is enabled, Jaspersoft OLAP instead evaluates such calculations to null, which can be helpful in certain circumstances.</p>
Comma-separated list of classes to use to get statistics about the number of rows in a table	<p>Comma-separated list of classes to use to get statistics about the number of rows in a table or the number of distinct values in a column.</p>
Solve Order evaluation behavior	<p>Specifies how to evaluate the <code>SOLVE_ORDER</code>:</p> <ul style="list-style-type: none"> • Absolute. Any specified scope is ignored and the <code>SOLVE_ORDER</code> value is absolute. For example, Jaspersoft OLAP gives precedence to a query-defined calculated member with a <code>SOLVE_ORDER</code> of 1 over a cube defined value of 2. • Scoped. Jaspersoft OLAP first resolves cube-calculated members, then resolves session-scoped members, and finally resolves query-defined calculations. Jaspersoft OLAP only applies the <code>SOLVE_ORDER</code> value within the scope in which it was defined.
Aggregate Settings	
Enable Aggregate Tables	<p>When enabled, Jaspersoft OLAP identifies tables in the database that are aggregates, and uses those aggregate values.</p> <p>For more information, refer to the configuration guide.</p>

Property	Notes
Choose Aggregate Table By Volume	<p>Performance tuning variable for aggregates. This property only applies when aggregate tables are used. Consider using this property when optimizing for tables with many large columns. this property determines whether aggregate tables are ordered by volume or row count.</p> <p>When this property is enabled, Jaspersoft OLAP uses the aggregate table with the smallest volume (number of rows multiplied by number of columns); when it is disabled, Jaspersoft OLAP uses the aggregate table with the fewest rows.</p>
Optimize predicates	<p>Determines whether Jaspersoft OLAP optimizes predicates. When this property is enabled, Jaspersoft OLAP optimizes certain predicates.</p> <p>When it is disabled, predicates are only optimized when all of a dimension's members are included. In this case, Jaspersoft OLAP only retrieves the data specified in the query.</p> <p>For more information, refer to the configuration guide.</p>
Rule file for aggregate table identification	<p>Specifies a file that defines aggregate table recognition rules. This file can either reside in the application server or the file system. Typically, you can accept the default.</p>
AggRule element's tag value	<p>The AggRule element's tag value. Typically, you can accept the default.</p>
SQL to log for aggregate table creation to support MDX with aggregates	<p>Determines whether Jaspersoft OLAP prints the SQL code generated for aggregate tables.</p> <p>When this property is enabled, Jaspersoft OLAP processes each aggregate request and prints both the lost and collapsed dimension <code>create</code> and <code>insert</code> SQL statements. When used in conjunction with the CmdRunner, it lets you automatically create aggregate table generation SQL.</p>
Factory class for determining the tables and columns of a data source	<p>If entered, this must be a Java class name that is an implementation of the <code>mondrian.rolap.aggmatcher.JdbcSchema.Factory</code> interface. The default implementation is <code>mondrian.rolap.aggmatcher.JdbcSchema.StdFactory</code>. Refer to the Mondrian Javadoc for more information.</p>
Cache and SQL Generation Settings	
Use a cache for the results of frequently evaluated expressions	<p>Determines whether Jaspersoft OLAP caches the results of frequently-evaluated expressions. Enabling this property can reduce the number of unnecessary computations when processing a large amount of data.</p>
Cache RolapCubeMember objects	<p>Determines whether Jaspersoft OLAP caches RolapCubeMember objects, which each associate a member of a shared hierarchy with a cube that uses it.</p> <p>Disable this property if you plan to use the member cache control.</p>

Property	Notes
Defines which SegmentCache implementation to use	Defines the SegmentCache implementation to use. Specify the value as a fully qualified class name, such as <code>org.example.SegmentCacheImpl</code> (where <code>SegmentCacheImpl</code> is an implementation of <code>mondrian.spi.SegmentCache</code>).
Maximum number of threads per Jaspersoft OLAP instance used to perform operations on external caches	The maximum number of threads per Jaspersoft OLAP instance that can be used to perform operations on external caches. For more information about running multiple instances in a single deployment, refer to “Performance Tuning” on page 78 .
Maximum number of threads per Jaspersoft OLAP instance used to run SQL queries when populating segments	The maximum number of threads per Jaspersoft OLAP instance that can be used to run SQL queries when populating segments. For more information about running multiple instances in a single deployment, refer to “Performance Tuning” on page 78 .
Determines whether the data from segments is cached locally	Determines whether data from segments is cached locally. To create custom caches, implement the SegmentCache SPI (Schema Processing Interface).
Maximum number of constraints in a single 'IN' SQL clause	The maximum number of constraints in a single SQL IN clause. This value varies with your RDBMS type and configuration. Jaspersoft recommends these values: <ul style="list-style-type: none"> • DB2: 2,500 • MySQL: 10,000 • Oracle: 1,000 • Postgres: 10,000 • Other: 10,000
Some NON EMPTY CrossJoin MDX statements will be computed in the database	When enabled, Jaspersoft OLAP pushes down to the database processing related to some NON EMPTY <code>CrossJoin</code> MDX statements.
Some TopCount MDX statements will be computed in the database	When enabled, Jaspersoft OLAP pushes down to the database processing related to some <code>TopCount</code> MDX statements.
Some Filter() MDX statements will be computed in the database	When enabled, Jaspersoft OLAP pushes down to the database processing related to some <code>Filter()</code> MDX statements.
Some NON EMPTY MDX statements will be computed in the database	When enabled, Jaspersoft OLAP pushes down to the database processing related to some NON EMPTY MDX set operations (Such as <code>member.children</code> , <code>level.members</code> and <code>member.descendants</code>).
Expand non native sub-expressions of a native expression into MemberLists	When enabled, Jaspersoft OLAP expands non-native sub-expressions of a native expression into <code>MemberLists</code> .

Property	Notes
If enabled, some queries against high-level members of snowflake dimensions are more expensive	When enabled, some queries against members of high levels of snowflake dimensions are more expensive. When disabled, and some rows in an outer snowflake table aren't referenced by a row in an inner snowflake table, some queries return members that have no children.
Alerting action in case native evaluation of a function is enabled but not supported in a particular query	Specifies the error level (OFF, WARN, or ERROR) to use when a function's native evaluation is enabled but isn't supported in the context of a particular query's usage. Errors are only returned in the case of a NonEmptyCrossJoin. Alerts are only raised when there is a possibility that native evaluation would improve matters.
Some rollup queries will be combined using group by grouping sets if the SQL dialect allows it	When enabled, Jaspersoft OLAP combines some rollup queries using <code>group by grouping sets</code> . Note that this property only applies to data stored in Oracle, DB2, or Teradata.
XML/A Settings	
XML/A Maximum Drill Through Rows	Limits the number of rows returned from an XML/A drill-through operation.
First row in the result of an XML/A drill-through request will be filled with the total count of rows	If this property is enabled, the first row returned for an XML/A drill-through request shows the total count of rows in the underlying database.
Interval between refreshes to the list of XML/A catalogs	Determines the frequency of refreshes to the list of XML/A catalogs in a remote server. Values can be suffixed with units of measures for time, such as: <ul style="list-style-type: none"> • <code>s</code> to specify seconds • <code>ms</code> to specify milliseconds If no unit of measure is specified, Jaspersoft OLAP assumes the unit is milliseconds.
Memory Monitoring Settings	
Use Java Memory monitoring to avoid out of memory	When enabled, Jaspersoft OLAP uses the Java's memory monitoring capability. In this case, when memory is exhausted, Jaspersoft OLAP returns a <code>MemoryLimitExceededException</code> exception rather than an out of memory error.
Threshold to limit memory used (percent of total memory)	Specifies the percent of memory usage that should trigger a notification to Jaspersoft OLAP that memory is low.

Property	Notes
The number of previous execution instances Jaspersoft OLAP keeps in its history	Defines the number of previous execution instances the Jaspersoft OLAP keeps in its history so that it can send the events that happen thereafter. Setting this property too high can cause the Java Virtual Machine to run out of memory. Setting it too low can prevent some events from reaching the monitor's listeners.
Class name of factory to manage memory	If entered, this must be a Java class name that is an implementation of the <code>mondrian.util.MemoryMonitor</code> interface. Such a class (<code>mondrian.util.NotificationMemoryMonitor</code>). Refer to the Mondrian Javadoc for more information.

These properties are stored in the repository (at /properties) as a list of values called `GlobalPropertiesList`. This list is populated the first time you alter one of the properties. Jaspersoft recommends that you allow JasperReports Server to manage this list of values for you automatically; therefore, only change the values using the **Manage > Server Settings** menu option.

For more information about these properties, refer to the [configuration guide](#). Use the name of each property (shown below each descriptive label) when searching this reference.

Many other properties can be set by editing configuration files, but some of them do not affect Jaspersoft OLAP because they are controlled in the repository user interface. For example, you do not have to provide a `connectString` property, or to specify the `jdbcDrivers` for Mondrian, because Jaspersoft OLAP automatically generates these properties when the `DataSource` for an `AnalysisClientConnection` is selected.



To log the SQL queries Jaspersoft OLAP sends to the database, set the **Generate Formatted SQL Traces** option. This ensures that Jaspersoft OLAP writes the SQL queries it executes to the log file you specify.

3.3.2 Flushing the OLAP Cache

This option clears the in-memory cache that Mondrian builds; caching is used to improve query performance. Flushing the cache is not usually necessary except when underlying data in the database changes. For example, after running an ETL process, the cache is out of date; it may not match the data in the database.

Such data changes are not reflected in OLAP views until the application server is restarted or until the cache is flushed.

To flush the cache, click **Manage > Server Settings**, click **OLAP Settings**, and click **Flush OLAP Cache** near the top of the page.

The OLAP cache is also automatically flushed when an existing Mondrian connection or one of its components (such as the schema or data source) is changed in the repository.

For more information on caching, see the *Jaspersoft OLAP Ultimate Guide* and the *JasperReports Server Administrator Guide*.

3.3.3 Configuring OLAP Cache Reuse

JasperReports Server caches your data when a user opens an OLAP view, Ad Hoc view, or OLAP-based report. These data are reused if they are subsequently requested before the cache is flushed. To determine whether a new request can be fulfilled with cached data, the server creates an identifier (by combining several connection attributes of the OLAP schema (such as `catalogUrl`, `connectionKey`, `jdbcUser`, `dataSourceStr` (or `catalogUrl`), and `dataSource`)). The server then compares the identifier to those of datasets in the cache. If it matches one, the data is reused; otherwise, the server assumes that this is a unique request and loads new data into the cache.

Alternatively, you can configure the server to compare schemas' checksums. In this cases, the server calculates the checksum of the schema being loaded and compares it to the checksums of schemas with datasets in the cache. If it matches one, the data are reused; otherwise, the server assumes that this is a unique request and loads new data into the cache.

If you want the server to identify datasets in the OLAP cache using a checksum, you can edit a configuration file:

OLAP Schema Checksum Comparison for Caching	
Configuration File	
...\\WEB-INF\\classes\\mondrian.connect.string.properties	
Property	Description
<code>UseContentChecksum</code>	When this property is set to <code>true</code> , JasperReports Server calculates an OLAP schema's checksum and compares that to checksums of schemas with datasets already in the cache. If it matches one, the data are reused; otherwise, the server assumes that this is a unique request and loads new data into the cache.



Jaspersoft recommends caution if you configure the server to identify datasets by the checksum of the associated OLAP schema. Under very specific circumstances, setting this property to `true` can cause unexpected behavior in which cached data are reused incorrectly. Users might see data associated with a different schema whose data is already in the cache simply because the checksums match. The following scenario presents one such case, though others are certainly possible.

Consider this server configuration:

- The JasperReports Server instance hosts two organizations: Client1 and Client2.
- Each organization has a user responsible for data analysis:
 - Client1's data analyst is Amber.
 - Client2's data analyst is Huan.
- Each organization includes an OLAP-based report that uses identical OLAP schemas:
 - Client1
 - Report: Sales Leads OLAP Report (C1)
 - OLAP Schema: SalesLeadsSchemaC1
 - Client2
 - Report: Sales Leads OLAP Report (C2)
 - OLAP Schema: SalesLeadsSchemaC2

- Each OLAP schema relies on a different database with the same data structures, so that the users in each organization see their own data.

Now consider this use case:

- Amber (in the Client organization) opens the Sales Leads OLAP Report (C1).
Amber sees sales leads generated by Client1. In the cache, this data is identified by the checksum of the SalesLeadsSchemaC1 schema.
- Before the cache is flushed, Huan (in the Client2 organization) opens the Sales Leads OLAP Report (C2).
By default, Huan sees sales leads generated by Client2 (which is the expected result). However, when the `UseContentChecksum` property is set to `true`, the server compares the checksum of the of the SalesLeadsSchema2 with those in the cache. Because this checksum matches the checksum of SalesLeadsSchema1 (the schema loaded by Amber), the server returns the cached data instead of loading data from Client2's data source.



In this scenario, Huan sees another client's sales data.

If your JasperReports Server instance hosts multiple organizations, Jaspersoft recommends that you use the default setting and allow the server to identify OLAP datasets using their connection URL details.

Note that other scenarios can cause similar unexpected behavior when the `UseContentChecksum` property is set to `true`; two checksums could conceivably match for any number of unpredictable reasons.

3.3.4 Configuring the XML/A Cache

When JasperReports Server connects to an XML/A provider to retrieve data that populate views and reports, it relies on a cache to improve performance. When data is first requested from the XML/A provider, it is retrieved and cached. Subsequent requests for the data are then fulfilled from the cache until it is refreshed. You can configure the frequency and behavior of the cache's refresh mechanism by editing a properties file:

XML/A Cache Configuration		
Configuration File		
...\\WEB-INF\\applicationContext-olap-connection.xml		
Property	Value	Description
OLAP4J_CACHE	org.olap4j.d- river.xmla.cache. XmlaOlap4jNamedMemoryCache	Do not change this value.
OLAP4J_CACHE_NAME	org.olap4j.d- river.xmla.cache. XmlaOlap4jNamedMemoryCache	Do not change this value.

XML/A Cache Configuration		
OLAP4J_CACHE_MODE	LFU	<p>Specifies the eviction policy to use when determining what data to evict from the cache. Valid values are:</p> <ul style="list-style-type: none">• LIFO: Last In First Out• FIFO: First In First Out• LFU: Least Frequently Used• MFU: Most Frequently Used
OLAP4J_CACHE_SIZE	Commercial Editions: 10000 Community Project: 1000	<p>The number of cache entries to maintain. The number of entries generated is determined by the number of queries sent to the XML/A provider via SOAP.</p>
OLAP4J_CACHE_TIMEOUT	Commercial Editions: 3600 Community Project: 600	<p>The length of time, expressed in seconds, to keep an entry in the cache. The default is one hour in commercial editions and ten minutes in the community project.</p>

3.4 Working with OLAP Objects in the Repository

OLAP views rely on several other types of object in the repository. This section describes their creation and maintenance, including:

- [Working with Data Sources](#)
- [Working with OLAP Schemas](#)
- [Working with Mondrian Connections](#)
- [Working with XML/A Connections](#)
- [Working with XML/A Sources](#)
- [Working with Access Grant Definitions](#)



An OLAP view references most of these objects indirectly. The same holds true for Ad Hoc views. For example, an OLAP schema is a part of a Mondrian connection; the OLAP view refers to the Mondrian connection which in turn refers to the schema. The following figures can help you understand how the objects relate:

- [“Anatomy of an OLAP View” on page 25](#)
- [“Anatomy of a Mondrian Connection” on page 54](#)
- [“Anatomy of an XML/A Connection” on page 65](#)
- [“Anatomy of an XML/A Source” on page 71](#)

The repository objects described in this section are also used by Ad Hoc views that return OLAP data. Such views are created against OLAP client connections (Mondrian or XML/A) using the Ad Hoc Editor. For more information on Ad Hoc views, refer to the *JasperReports Server User Guide*.

3.4.1 Working with Data Sources

A data source is a resource in the repository that defines how and where to obtain the data displayed by reports or views. Typically, it includes the location of the data and the details you need to access it, such as a user name and password. In the case of Jaspersoft OLAP, data sources are exposed through Mondrian connections.

For detailed information about data sources, refer to the *JasperReports Server Administrator Guide*.

To edit a data source:

1. In the **Search** field, enter the name (or partial name) of the object you want to edit, and click the Search icon. For example, enter `food`.

The search results display objects that match the text you entered.

2. Right-click the data source and click **Edit** from the context-menu.

The Set Data Source Type and Properties page appears.

Edit Data Source

Set Data Source Type and Properties

First, select the type of data source you wish to add, then enter the required property values.

Type:

Name (required):

Resource ID (read-only): FoodmartDataSource

Description:

Driver (required):
Hint: org.postgresql.Driver

URL (required):
Hint: jdbc:postgresql://localhost:5432/mydb

User Name (required):

Password:

Time Zone:
Hint: Do not change the time zone setting unless you know the database timestamp data is incorrect.

Save Location:

Figure 3-19 Set Data Source Type and Properties



The options displayed depend on the type of data source selected.

3. Change the values as necessary and click **Test Connection** to ensure the connection is valid.

A banner at the top of the page displays a message: **Connection successful** or **Connection failed**.

4. If the connection test fails, check the values you entered, or check that the data is available from the service you specify, and test the connection again.
5. When the test succeeds, click **Submit**.

The data source is saved to the repository.

3.4.2 Working with OLAP Schemas

An OLAP schema is a metadata definition of a multidimensional database. Use the Jaspersoft OLAP workbench to create OLAP schemas, and then upload them to the JasperReports Server repository. For more information on the workbench, refer to **“Overview of Jaspersoft’s OLAP Tools” on page 5**. OLAP schemas are stored in the repository as XML file resources.

This section describes how to work with the OLAP schema through the web UI (including **“Uploading an OLAP Schema” on page 52** and **“Editing an OLAP Schema” on page 52**).

3.4.2.1 Uploading an OLAP Schema

Upload an OLAP schema to the repository so that it can be accessed by more than one Mondrian connection. Doing this before creating a view simplifies the procedure for defining Mondrian connections and OLAP views.

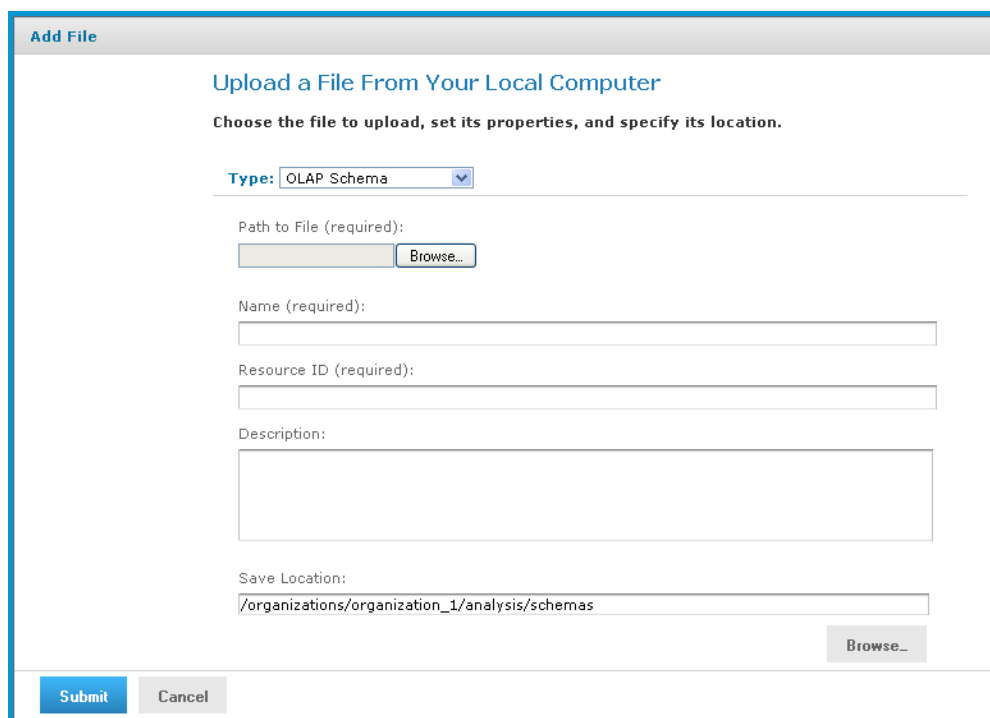
To upload a schema :

1. Click **View > Repository**.

The repository appears.

2. In the Folder panel, navigate to **Analysis Components > Analysis Schemas**.
3. Right-click the folder and navigate to **Add Resource > File > OLAP Schema**.

The Upload a File From Your Local Computer page appears and prompts you to select a file and set its properties.



The screenshot shows a web-based form titled "Add File" with a subtitle "Upload a File From Your Local Computer". Below the subtitle is the instruction "Choose the file to upload, set its properties, and specify its location." The form contains the following fields and controls:

- Type:** A dropdown menu currently set to "OLAP Schema".
- Path to File (required):** A text input field followed by a "Browse..." button.
- Name (required):** A text input field.
- Resource ID (required):** A text input field.
- Description:** A larger text input area.
- Save Location:** A text input field containing the path "/organizations/organization_1/analysis/schemas", followed by a "Browse..." button.
- At the bottom are two buttons: "Submit" and "Cancel".

Figure 3-20 Upload a File From Your Local Computer - OLAP Schema

4. Under **Path to File**, click **Choose File** and locate the OLAP schema you want to add.
5. Enter a name and description for the schema.

The **Resource ID** is auto-generated as you type in the **Name** field. You can change the ID if necessary.

6. Next to the **Save Location** field, click **Browse** and navigate to the location in the repository where you want the file to reside.
7. Click **Submit**.

The new file appears in the repository.

3.4.2.2 Editing an OLAP Schema

You can change the schema name, the schema file, and location of an OLAP schema.

To change an OLAP schema's naming and file source:

1. In the **Search** field, enter the name (or partial name) of the schema you want to edit, and click the Search icon. For example, enter `sugar`.

The search results appear, displaying objects that match the text you entered.

2. Select the schema, right-click, and click **Edit**.

The Upload a File From Your Local Computer page appears and prompts you to the values.



You cannot change the **Type** or **Resource ID** fields.

Figure 3-21 Upload a File From Your Local Computer - OLAP Schema

3. To upload a new file, next to the **Path to File** field, click **Browse** and navigate to and select the file you want to upload.
4. Enter changes to the **Name** and **Description** fields as necessary.
5. Under the **Save Location** field, click **Browse** and navigate to the location where you want to store the file, and click **Select**.
6. Click **Submit**.

The edited schema appears in the repository.

3.4.3 Working with Mondrian Connections



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 Mondrian connection describes how to present your transactional data as a multidimensional cube for analysis.

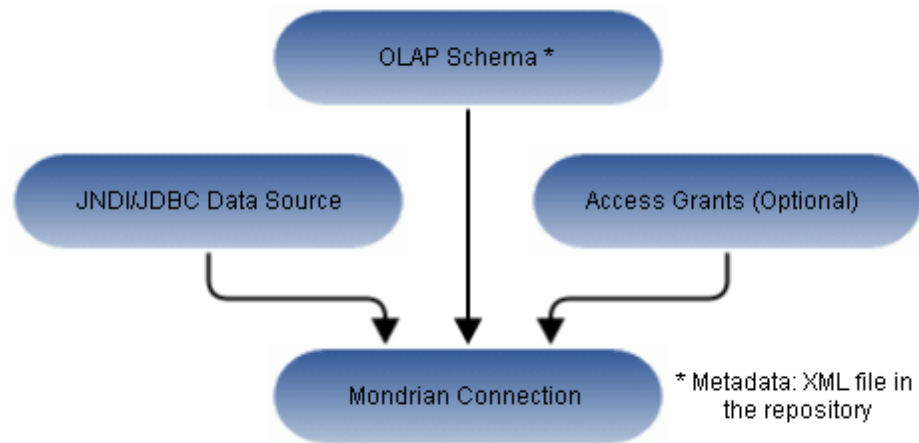


Figure 3-22 Anatomy of a Mondrian Connection

If you use JasperReports Server Community Project, you cannot include access grants in your Mondrian connection. Data-level security is only supported in commercial editions of the server.

3.4.3.1 Creating a Mondrian Connection

To create a Mondrian connection:

1. Click **View > Repository**.
The repository page appears.
2. In the **Folders** panel, navigate to **Organization > Organization > Analysis Components > Analysis Connections**.
3. Right-click the folder and select **Add Resource > OLAP Client Connection**.
The Set Connection Type and Properties page appears and prompts you to define a connection.

Add OLAP Client Connection

Set Connection Type and Properties

First select the type of connection you want to add, then enter the required property values.

Type: Connection Type Mondrian

Name (required):

Resource ID (required):

Description:

Select folder from repository to save
 Browse...

Previous Next Cancel

Figure 3-23 Set Connection Type and Properties Page

By default, the server prompts you to create a Mondrian connection. If you want to create an XML/A connection, refer to [“Creating an XML/A Connection” on page 66](#).

4. Enter a name and description for the new connection. The **Resource ID** field is auto-generated when you type in the **Name** field. After it is saved, it can't be changed.
5. To change the location of the connection, click **Browse**, navigate to a folder, and click **Select**.
6. Click **Next**.

The Locate OLAP Schema page appears and prompts you to upload an OLAP schema or select one from the repository.

Add OLAP Schema Resource

Locate OLAP Schema

☐ Upload a Local File Browse...

☒ Select a resource from the Repository Browse...

Previous Next Cancel

Figure 3-24 Locate OLAP Schema Page

7. Click either:
 - **Upload a Local File** to select a file from your local computer.
Then click **Choose File**, navigate to select the file, and click **Select**.
 - **Select a resource from the Repository** to select an existing schema.
Then click **Browse**, navigate to select the file, and click **Select**.
8. Click **Next**.

The OLAP Schema Resource page appears.

Figure 3-25 OLAP Schema Resource Page

If you chose to upload a new file from your computer, the fields are editable. Enter the requested information. For details, refer to [“Uploading an OLAP Schema” on page 52](#). If you chose a file from the repository, the fields aren’t editable.

9. Click **Next**.

The Locate Data Source page appears and prompts you to create or select a data source.

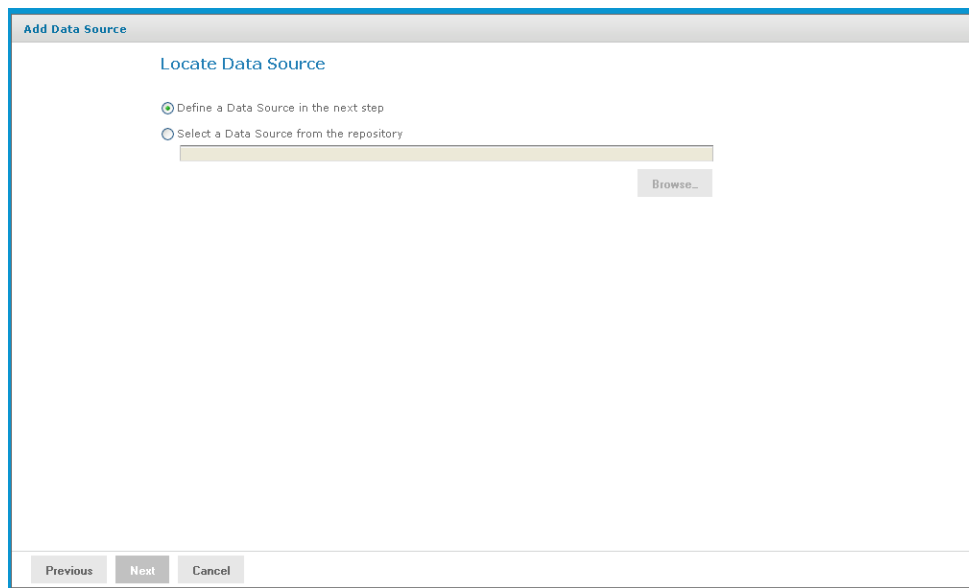


Figure 3-26 Locate Data Source Page

10. Click either:

- **Define a Data Source in the next step** to add a data source.
- **Select a Data Source from the repository** to select a data source from the repository.

Then click **Browse**, navigate to select the file, and click **Select**.

11. Click **Next**.

The Set Data Source Type and Properties page appears.

Add Data Source

Set Data Source Type and Properties

First, select the type of data source you wish to add, then enter the required property values.

Type:

Name (required):

Resource ID (required):

Description:

Driver (required):
Hint: org.postgresql.Driver

URL (required):
Hint: jdbc:postgresql://localhost:5432/mydb

User Name (required):

Password:

Time Zone:
Hint: Do not change the time zone setting unless you know the database timestamp data is incorrect.

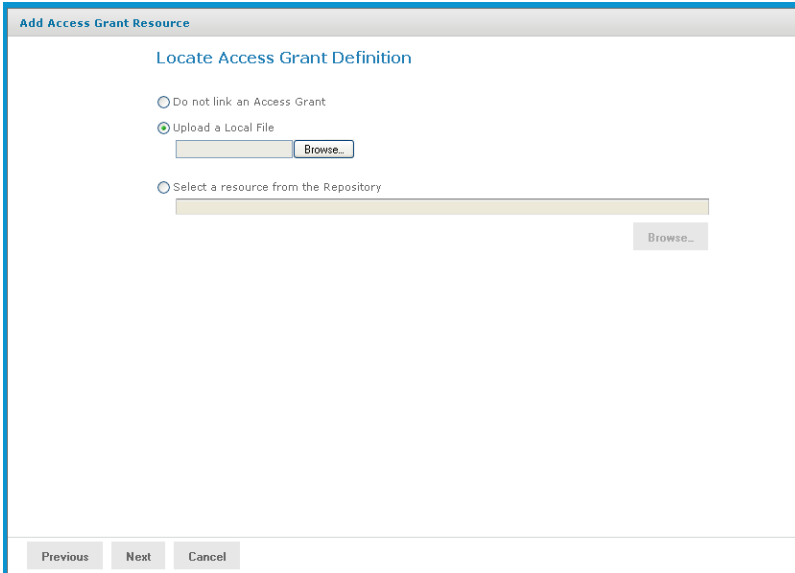
Save Location:

Figure 3-27 Set Data Source Type and Properties Page

If you chose to define a new data source, the fields are editable. Enter the requested information. For details, refer to **“Working with Data Sources” on page 50**. If you chose a data source from the repository, the fields aren’t editable.

12. Click **Next**.

The Locate Access Grant Definition page appears and prompts you to set the properties for the resource.



Add Access Grant Resource

Locate Access Grant Definition

☐ Do not link an Access Grant

☒ Upload a Local File

Browse...

☐ Select a resource from the Repository

Browse...

Previous **Next** **Cancel**

Figure 3-28 Locate Access Grant Definition Page

13. Click one of the following:

- **Do not link an Access Grant** if you don't need to apply data security.
Then skip to step [step 16](#).
- **Upload a Local File** to select a file from your local computer.
Then click **Browse**, navigate to select the file you want, and click **Select**.
- **Select a resource from the Repository** to select an existing schema.
Then click **Browse**, navigate to select the schema, and click **Select**.

14. Click **Next**.

If you chose to secure the view, the Access Grant Resource page appears.

Figure 3-29 Access Grant Resource Page

15. If you chose to upload a new AGXML file, the fields are editable. Enter the requested information. For details, refer to **“Uploading an Access Grant Schema” on page 77**. If you chose an access grant file from the repository, the fields aren’t editable.

16. Click **Next**.

The Mondrian connection is added to the repository. Views can now reference this connection to expose data to your users. For information on creating OLAP views, refer to **“Administering OLAP Views” on page 24**. For information of creating Ad Hoc views, refer to *JasperReports Server User Guide*.

3.4.3.2 Editing a Mondrian Connection

You can change the connection name, schema, data source, and access grant definition in a Mondrian connection.

To edit a Mondrian connection:

1. In the **Search** field, enter the name (or partial name) of the Mondrian connection you want to edit, and click the Search icon. For example, enter `foodmart`.

The repository appears and displays objects that match the text you entered.

2. Right-click the Mondrian connection you want to edit and click **Edit**.

The Set Connection Type and Properties page appears with the fields populated.

Add OLAP Client Connection

Set Connection Type and Properties

First select the type of connection you want to add, then enter the required property values.

Type: Connection Type **Mondrian**

Name (required):

Resource ID (required):

Description:

Select folder from repository to save
 Browse...

Previous **Next** **Cancel**

Figure 3-30 Set Connection Type and Properties Page

3. Change values as necessary and click **Next**.
 The Locate OLAP Schema page appears.

Add OLAP Schema Resource

Locate OLAP Schema

☐ Upload a Local File
 Browse...

☒ Select a resource from the Repository
 Browse...

Previous **Next** **Cancel**

Figure 3-31 Locate OLAP Schema Page

You can either accept the existing file or replace it. If you replace the file, you can either upload a new file or select one from the repository.

4. To accept the existing file, click **Next**.
5. To replace the file, either:
 - Click **Upload a Local File** and click **Browse** to upload a new schema from your local computer.
 - Click **Select a resource from the Repository**, click **Browse**, and navigate the repository to the schema you want to use. Then click **Select**.
6. Click **Next**.

The OLAP Schema Resource page appears.

Figure 3-32 OLAP Schema Resource

7. If you chose to upload a new file, enter a name and description for it.
If you accepted the existing file or selected one from the repository, the fields aren't editable.
8. Click **Next**.

The Locate Data Source page appears.

Figure 3-33 Locate the Data Source

You can either accept the existing data source or replace it. If you replace it, you can either define a new data source or select one from the repository.

9. To accept the existing data source, click **Next**.
10. To replace the data source, either:
 - Click **Define a Data Source in the next step**.
 - Click **Select a Data Source from the Repository**, click **Browse**, and navigate the repository to locate the data source you want to use. Then click **Select**.
11. Click **Next**.

If you accepted the existing data source, or if you selected a data source from the repository, that connection is used. Clicking **Next** displays the Locate Access Grant page.

If you chose to define a new data source, the Set Data Source Type and Properties page appears.

Figure 3-34 Set Data Source Type and Properties Page

12. Enter the requested information, For details on defining data sources, refer to **“Working with Data Sources” on page 50** and to the *JasperReports Server Administrator Guide*.



Test the new data source to ensure it works properly.

13. Click **Submit**.

The Locate Access Grant Definition page appears.

Figure 3-35 Locate Access Grant Definition Page

14. Click one of the following options:

- **Do not link an Access Grant.** Click **Next** and skip to [step 17](#).
- **Upload a Local File.** Click **Browse** to select a different local file.
- **Select a resource from the Repository.** Click **Browse** to select a different file in the repository.

15. Click **Next**.

If you chose to secure the data, the Access Grant Resource page appears.

Figure 3-36 Access Grant Resource Page

16. If you upload a new AGXML file, enter the requested information. For details, refer to **“Uploading an Access Grant Schema” on page 77**. If you selected a resource from the repository, the fields aren’t editable.
17. Click **Submit**.

The updated connection appears in the repository.

3.4.4 Working with XML/A Connections

An XML/A connection describes how to connect to a remote XML/A provider, such as an XML/A source in another instance of JasperReports Server. In this case, the XML/A source points to a Mondrian connection in the remote server.

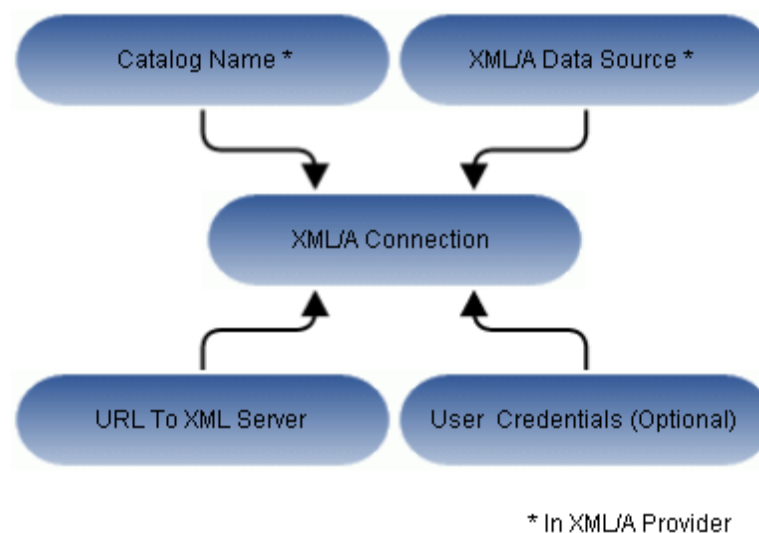


Figure 3-37 Anatomy of an XML/A Connection

JasperReports Server can act as either an XML/A provider or as an XML/A client. In the former case, remote servers retrieve data from an OLAP data source exposed by an XML/A definition in the JasperReports Server repository. In the latter case, JasperReports Server connects to a remote XML/A provider to retrieve data that populate reports and views. Typically, the server accesses a remote instance of JasperReports Server to form a distributed environment (**“Performance Tuning” on page 78**), but the server can also connect to other types of XML/A provider, such as Microsoft SQL Server Analytic Services (SSAS). Because JasperReports Server uses OLAP4J (<http://www.olap4j.org/>), it may also be able to connect to other types of XML/A provider, though such configurations aren’t currently certified. In this release, only JasperReports Server and Microsoft SQL Server Analytic Services are certified as XML/A providers.

The remote server must also be configured for XML/A. For instructions regarding Jaspersoft OLAP, see **“Jaspersoft OLAP Administration” on page 23** and **“Creating an XML/A Source” on page 72**. If you’re connecting to Microsoft SQL Server Analytic Services, at a high level, you must:

- Configure Internet Information Services (IIS) to provide connectivity to Microsoft SQL Server Analytic Services.
- Deploy and configure msmdpump.dll, which is an IIS extension that provides data connectivity via HTTP.
- Configure security in Microsoft SQL Server Analytic Services according to your needs.

- Test Microsoft SQL Server Analytic Services using an XML/A client tool, such as waRehouse EXplorer (REX; <http://sourceforge.net/projects/whex>), which is an open source Java client for browsing multidimensional data sources. Testing with a third-party tool can help you troubleshoot your Microsoft SQL Server Analytic Services configuration.

For more information about Microsoft SQL Server Analytic Services, refer to the associated documentation.

3.4.4.1 Creating an XML/A Connection

When creating an XML/A connection, the type of server providing the data determines the values you must specify. This section generally assumes you're connecting to a Mondrian connection stored in a remote JasperReports Server, but also provides some details about connecting to Microsoft SQL Server Analytic Services.

To create an XML/A connection:

1. Click **View > Repository**.

The repository page appears.

2. In the **Folders** panel, navigate to **Organization > Organization > Analysis Components > Analysis Connections**.

3. Right-click the folder and select **Add Resource > OLAP Client Connection**.

The Set Connection Type and Properties page appears and prompts you to define a connection.

4. In the **Connection Type** drop-down, select XML/A Connection.

The page refreshes and prompts you to define the XML/A connection.

5. Enter general details, such as the name, label, and description of the connection.
6. Enter the details, such as the catalog, data source, and URI, that define the XML/A source you want to connect to:
 - a. Catalog: the name of the schema that defines the data cube.
 - b. Data Source:

- If you are connecting to JasperReports Server, enter full connection string. For example:
`Provider=Mondrian;DataSource=JRS`

Note that, in previous releases, the `DataSource` portion of the connection string was the catalog name; in the current release, it is always `JRS`.

- If you are connecting to Microsoft SQL Server Analytic Services and the connection will be used by OLAP views and reports created in iReport or Jaspersoft Studio, enter the full connection string. For example: `Provider=MSOLAP.4;Data Source=172.16.254.1;Catalog=AdventureWorks`
- If you are connecting to Microsoft SQL Server Analytic Services and the connection will be used by Ad Hoc views and their reports, use the Microsoft SQL Server's instance name. For example:
`Win-MyHost`



When connecting to Microsoft SQL Server Analytic Service, the form of the data source depends on the way you plan to use this XML/A connection:

- If you plan to use the XML/A connection to create Ad Hoc views, use the Microsoft SQL Server's instance name. This is typically the name of the computer hosting Microsoft SQL Server. For example, if your Microsoft SQL Server instance is installed on `Win-MyHost`, the data source is: `Win-MyHost`.
- If you plan to use the XML/A connection to create OLAP views, use the full connect string. For example, if your Microsoft SQL Server instance is installed on a computer with the IP address `172.16.254.1`, and your catalog is named `AdventureWorks`, the data source is:
`Provider=MSOLAP.4;Data Source=172.16.254.1;Catalog=AdventureWorks`

c. URI (Uniform Resource Identifier): the identifier of the XML/A provider; typically a computer name or URL.

7. Enter credentials (the user name and password) that Jaspersoft OLAP can pass to the remote XML/A provider to log in. If this user's password changes, the connection fails. You can leave the **User Name** and **Password** fields blank, so the logged in user's credentials are passed to the remote server when the connection is accessed.

If the name of the user includes a backslash (\), you must escape the character by placing a backslash in front of it. For example, consider the case when the user name includes a domain, such as `domain\username`; this is represented in the **User Name** field as `domain\\username`.



The credentials you define for an XML/A connection are transmitted to the XML/A provider as clear-text. Because of the security risk inherent in this approach, Jaspersoft recommends that you always specify a user name and password when defining an XML/A connection in order to prevent your users' passwords from being transmitted. This user should have restricted rights in the remote XML/A provider. For more information, see section [“XML/A Security” on page 71](#).



Your XML/A provider may be another JasperReports Server instance where local Mondrian connections have been defined. For more information, refer to section [“Working with XML/A Sources” on page 70](#).

Add OLAP Client Connection

Set Connection Type and Properties

First select the type of connection you want to add, then enter the required property values.

Connection Type: **XML/A**

Name (required):

Resource ID (required):

Description:

Catalog (required):

Hint: Foodmart

Data Source (required):

Hint: Provider=Mondrian;DataSource=JRS

URI (required):

Hint: http://localhost:8080/jasperserver-pro/xmla

User Name:

Password:

Select folder from repository to save
/analysis/connections

Figure 3-38 Set Connection Type and Properties - XML/A Page

8. Click **Test Connection**.

Jaspersoft OLAP attempts to connect to the remote server:

- If it can connect, a message indicating success appears.
- If the connection fails, a message indicating the type of problem appears. For example, the message might indicate that a catalog with the specified name was not found in the data source; re-enter the catalog name and test the connection again. If a data source with the specified name isn't found, the message may indicate that no data source was found; examine your remote server's data sources, update the connection's details, and click **Test Connection** again.

9. Click the **Show Details** link to learn more about the problem.

10. When the test succeeds, click **Submit**.

11. Click **Submit**.

The new XML/A connection appears in the repository.



If you specify an instance of JasperReports Server as your XML/A provider (in the **URI** field), and it hosts more than one organization, specify the organization name in the **User Name** field, separated from the account name with the pipe character (|). For example, to connect as a user named `joeuser` in an organization named `organization_1`, specify `joeuser|organization_1` in the **User Name** field.

If you are logged in as superuser, you cannot use the Ad Hoc Editor to access data exposed through an XML/A connection. Instead, Jaspersoft recommends that you login as `jasperadmin` or a non-administrative user when creating Ad Hoc views from XML/A connections.

3.4.4.2 Editing an XML/A Connection



In previous releases, XML/A connections that pointed to remote JasperReports Server instances had to specify slightly different information than what is required in this release. If you have recently upgraded from a version prior to 5.6, you must edit these XML/A connections' **Data Source** field.

For example, in previous versions, the Foodmart XML/A connection specified:

```
Provider=Mondrian;DataSource=Foodmart
```

During upgrade, this connection must be changed to:

```
Provider=Mondrian;DataSource=JRS
```

To edit an XML/A connection's naming and properties:

1. In the **Search** field, enter the name (or partial name) of the connection you want to edit, and click the Search icon. For example, enter `sugar`.

The search results appear, displaying objects that match the text you entered.

2. Right-click the XML/A connection you want to change and click **Edit**.

The Set Connection Type and Properties page appears.

Figure 3-39 Set Connection Type and Properties

3. Make changes as necessary.
4. Click **Test Connection**.

Jaspersoft OLAP attempts to connect to the remote server:

- If it can connect, a message indicating success appears.

- If the connection fails, a message indicating the type of problem appears. For example, the message might indicate that a catalog with the specified name was not found in the data source; re-enter the catalog name and test the connection again. If a data source with the specified name isn't found, the message may indicate that no data source was found; examine your remote server's data sources, update the connection's details, and click **Test Connection** again.
- 5. Click the **Show Details** link to learn more about the problem.
- 6. When the test succeeds, click **Submit**.

The edited XML/A Connection appears in the repository.

3.4.5 XML/A Against JasperReports Servers Hosting Multiple Organizations

When your JasperReports Server instance hosts multiple organizations, clients that retrieve data from it using XML/A must specify an organization. JasperReports Server looks for the requested XML/A source in that organization. If it doesn't find it in that organization, it then looks in for the XML/A source in Public folders. The server passes data from the first XML/A source it finds that matches the specified ID. Since resources in Public folders are shared with all users regardless of their organization, Jaspersoft recommends caution in putting XML/A data sources in Public folders. When done thoughtfully, this configuration can ensure that XML/A requests are always fulfilled, but the approach necessarily exposes the data to a wider group of users than keeping the XML/A source in a more tightly-secured location.



When acting as an XML/A server, a server instance that hosts multiple organizations expects the user's organization ID to be passed in with the credentials. Since superuser is not associated with any organization, you can't pass the organization ID with the credentials. Because of this limitation, Jaspersoft strongly recommends that you specify a user associated with an organization (and pass the organization ID along with the user name) when connecting via XML/A. For more information, see [“Creating an XML/A Connection” on page 66](#) and [“XML/A Security” on page 71](#).

3.4.6 Working with XML/A Sources

This section describes the XML/A sources that XML/A connection access to retrieve data, and includes:

- [Overview of XML/A Sources](#)
- [XML/A Security](#)
- [Creating an XML/A Source](#)
- [Editing an XML/A Source](#)

3.4.6.1 Overview of XML/A Sources

An XML/A connection points to Mondrian connection (exposed by XML/A source) in a remote XML/A provider, allowing you to separate your front end processing from back end processing when analyzing data in views and reports.

Jaspersoft OLAP can run as a XML/A server, allowing OLAP queries to be run from remote clients and query results returned to those clients over the XML/A web services protocol. The OLAP data for XML/A comes from Mondrian connections referenced by XML/A sources.

Jaspersoft OLAP XML/A connections provide a client view to XML/A servers, including Jaspersoft OLAP. You may want to run separate Jaspersoft OLAP servers, splitting the Jaspersoft OLAP XML/A clients from the XML/A servers. This client/server configuration allows load balancing, around-the-clock availability, and fail

over in Jaspersoft OLAP environments. Other clients that can connect to Jaspersoft OLAP when it is run as a XML/A server include Excel Pivot Tables with the Jaspersoft ODBO Driver.

In order to allow Jaspersoft OLAP to run as a XML/A server, XML/A sources need to be defined in the repository. The list of XML/A sources a given client can connect to can be controlled by repository permissions, as XML/A clients must authenticate via user name and password, and their access to the repository can be limited by permissions. In commercial editions of Jaspersoft OLAP, OLAP data level security can be used to filter OLAP query results based on the logged in user making the XML/A requests. See **“Uploading an Access Grant Schema” on page 77**.

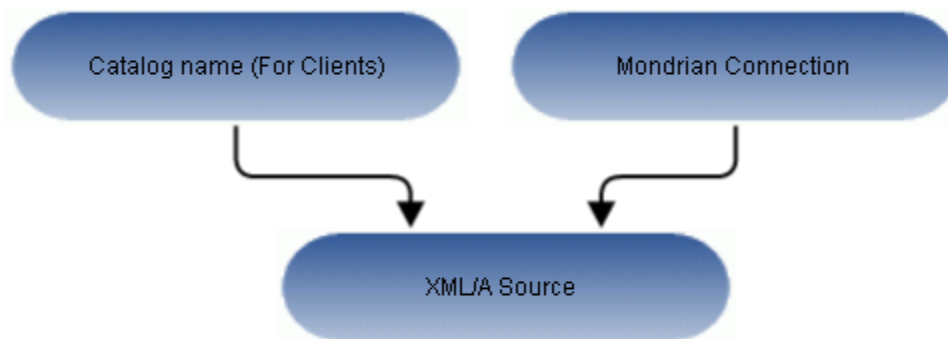


Figure 3-40 Anatomy of an XML/A Source



When your JasperReports Server instance hosts multiple organizations, an XML/A source must be created in the same organizations as the Mondrian connection it points to. If the Mondrian connection is in the Public folder, the XML/A source must also be in the Public folder.

3.4.6.2 XML/A Security

The default configuration uses HTTP Basic authentication to challenge requests for the /xmla path. If the client doesn't have a valid JasperReports Server user name and password in its XML/A connection source, the connection will fail, unless the user name and password are left blank; in this case, the credentials of the logged in user are passed by the client application to the remote server.

Put another way, when creating an XML/A connection, you can either specify a user name and password for all users to share, or you can leave user name and password blank, so that the connection passes the current user's name and password to the server.



With HTTP Basic authentication, clear-text passwords are transmitted in the header of an HTTP request unless you have configured JasperReports Server to use encrypted passwords. For more information, refer to the *JasperReports Server Administrator Guide*.



Regardless of the authentication method you use, clear-text passwords are also transmitted in the body of the XML/A request. Because of the security risk inherent in this approach, Jaspersoft recommends that you always specify a user name and password when defining an XML/A connection in order to prevent your users' passwords from being transmitted. Do not use the superuser account. For more information, see section **“Working with XML/A Connections” on page 65**.

3.4.6.3 Creating an XML/A Source

An XML/A source provides access to a single catalog (database schema) referenced by a Mondrian connection in a local instance of JasperReports Server. It defines a particular Mondrian connection in the repository that answers OLAP requests. The XML/A source is referenced by remote clients (such as an XML/A connection in a remote instance of JasperReports Server). The catalog name you specify uniquely defines the data that an XML/A client can retrieve from the source.

An XML/A source is also sometimes called an XML/A definition.

To add an XML/A source:

1. Click **View > Repository**.
2. In the Folder panel, navigate to **Organization > Analysis Components > xml/a**.
3. Right-click the folder and select **Add Resource > Mondrian XML/A Source** from the context-menu.

The Set Mondrian XML/A Source Properties page appears and prompts you to enter basic information.

Figure 3-41 Set Mondrian XML/A Source Properties Page

4. Enter XML/A source information. For example:

Name	SugarCrmMondrianXmlaSource
Resource ID	Sugar CRM Mondrian XML/A Source
Description	Sugar CRM Mondrian XML/A Source
Catalog	The name of the database that contains the data to analyze.
Mondrian Connection Reference	The path and name of the connection this XML/A source references. If the XML/A provider is JasperReports Server, its value is the connection's URI.

5. Click **Submit** to save the XML/A source.

Once you create an XML/A source in and XML/A provider such as Jaspersoft OLAP, you must create an XML/A connection in your client application that points to it. For more information, see **“Performance Tuning” on page 78**.

3.4.6.4 Editing an XML/A Source

You can change an XML/A source name, connection properties, and location in the repository.

To edit an XML/A source:

1. In the **Search** field, enter the name (or partial name) of the schema you want to edit, and click the Search icon.

For example, enter `food`.

The search results appear, displaying objects that match the text you entered.

2. Right-click an XML/A source, and click **Edit** on the context-menu.

The Set Mondrian XML/A Source Properties page appears.

Figure 3-42 Set Mondrian XML/A Source Properties Page

3. Click each active field you want to change and enter the new data.
4. Click **Submit**.

The updated source appears in the repository.

3.4.7 Working with Access Grant Definitions



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.

An access grant definition is an XML structure that specifies a user's access rights to different parts of the data defined by an OLAP schema. The access grant definition specifies access rights based on roles. Users with a

given role have the access rights granted to that role. An access grant definition can also refer to a profile attribute table that control access through properties defined for specific users. This allows you to use variable substitution to create simple, flexible access grants.

This section includes:

- **Overview of Data-level Access Using AGXML Schemas**
- **Sample Access Grant Definition**
- **Uploading an Access Grant Schema**
- **Working with Profile Attributes**
- **Best Practices for Designing Access Control**



AGXML depends on Jaspersoft's underlying OLAP engine, and as such only applies to data accessed by a local OLAP client connection (that is, a Mondrian connection). To restrict data accessed via XML/A, define your security in the remote host serving your data; for example, attach an AGXML schema to a Mondrian connection exposed by an XML/A source.

3.4.7.1 Overview of Data-level Access Using AGXML Schemas

An access grant definition depends on elements of the OLAP schema associated with a connection. That OLAP schema must contain cubes with the same name and structure as appear in the access grant definition's CubeGrant elements. If you specify access grants down to the member level, the references to member values in the access grant definition must be in the database defined by the connection.

You create an access grant definition as an XML file with an AGXML file extension. To use it, import it into the repository, or upload it while creating an OLAP client connection, just as you can upload an OLAP schema.

The elements are arranged hierarchically, as shown below. The grant definitions for a role lie within the following nested grant elements: SchemaGrant, CubeGrant, HierarchyGrant, and MemberGrant. SchemaGrant is the outermost element, and MemberGrant is innermost. In general, grants within an element override grants in containing elements.

```
SchemaGrant
  CubeGrant
    HierarchyGrant
      MemberGrant
```

The following table describes the attributes of the grant elements and lists possible attribute values:

Grant Element	Attribute	Attribute Description	Values
SchemaGrant	access	Defines the default access for any object in the schema.	<ul style="list-style-type: none"> • all • none
CubeGrant	access	<p>Defines the default access to hierarchies within the cube specified by its cube attribute.</p> <p>If the schema contains cubes for which no CubeGrant element appears, then the default access defined at the SchemaGrant level applies to those cubes.</p>	<ul style="list-style-type: none"> • all • none

Grant Element	Attribute	Attribute Description	Values
HierarchyGrant	access	Defines the role's access to the hierarchy specified by the <code>hierarchy</code> attribute. If <code>access = custom</code> , MemberGrant sub-elements define the role's access within the hierarchy. If a cube contains hierarchies for which no HierarchyGrant element appears, then the default access defined at the CubeGrant level applies to those hierarchies.	<ul style="list-style-type: none"> all none custom
	topLevel bottomLevel	Defines the segment of the hierarchy that users with the given role can see. They can see everything between and including the endpoints. Nothing in an enclosed MemberGrant element can override the <code>topLevel</code> and <code>bottomLevel</code> attributes.	(varies)
MemberGrant	member	Specifies the top level of the hierarchy defined by the enclosing HierarchyGrant element to which the MemberGrant's <code>access</code> attribute applies. For example, if the member value is <code>[Store].[USA].[CA]</code> , the top level to which the <code>access</code> attribute applies is California in the Store hierarchy.	(varies)
	access	Defines everything including or below the level specified by the member attribute, except that it cannot grant access to anything outside the segment defined by the HierarchyGrant attributes <code>topLevel</code> and <code>bottomLevel</code> .	<ul style="list-style-type: none"> all none

The `topLevel` and `bottomLevel` attributes use a dot notation to specify a level in the hierarchy. In the example in [“Sample Access Grant Definition” on page 76](#), the `topLevel` attribute for the Store hierarchy has the value `[Store].[Store.Country]`. The `[Store]` designates the Store hierarchy. The `[Store.Country]` designates the Country level of the Store hierarchy.

You can use substitution variables in the grant expressions that specify the values of the `topLevel`, `bottomLevel`, and `member` attributes. The example in [“Sample Access Grant Definition” on page 76](#) includes variable substitution.

Jaspersoft OLAP applies grants in the order that you define them. For example, if you grant access to USA and then deny access to Oregon, an affected user cannot see Oregon or any of its children (for example, Portland). But if you deny access to Oregon and then grant access to USA, the user can see Oregon and all of its children.

Granting access to a member also grants access to the levels above it, except that the grant cannot override the HierarchyGrant's `topLevel` attribute. For example, if you deny access to USA and then grant access to California, an affected user can see California and USA, but no other states. If the data includes USA totals, they will be based on data from all states.

You can test access grant definitions by creating users with various roles and logging in as them. For more information on roles, users, and the **Log in As** feature, refer to the *JasperReports Server Administrator Guide*. The *Jaspersoft OLAP Ultimate Guide* also includes a detailed implementation example.



Names of roles, users, hierarchical levels, and profile attributes are all case-sensitive in access grant definitions.

3.4.7.2 Sample Access Grant Definition

The following sample access grant definition (called Foodmart Grant and found in the repository at /analysis/schemas) is used with the Foodmart Mondrian connection, (found in the repository at /analysis/connections):

```
<Roles>
  <Role name="StateManager">
    <SchemaGrant access="none">
      <CubeGrant cube="Sales" access="all">
        <HierarchyGrant hierarchy="[Store]"
          access="custom"
          topLevel="[Store].[Store Country]"
        >
          <MemberGrant member="[Store].[USA].[%{State}]"
            access="all"/>
        </HierarchyGrant>
        <HierarchyGrant hierarchy="[Customers]"
          access="custom"
          topLevel="[Customers].[State Province]"
          bottomLevel="[Customers].[City]"
        >
          <MemberGrant member="[Customers].[USA].[%{State}]"
            access="all"/>
        </HierarchyGrant>
        <HierarchyGrant hierarchy="[Gender]"
          access="none"/>
      </CubeGrant>
    </SchemaGrant>
  </Role>
</Roles>
```

This schema defines access permissions for a role called StateManager. In this example, the SchemaGrant element defines none as the default access for all objects in the schema. Users with the StateManager role cannot access anything, unless a lower level grant overrides the default.

The cube attribute has the value Sales, so the definition is for the Sales cube. The access attribute has the value all. This gives users with this role access to the entire Sales cube. Again, lower level grants can override this one.

HierarchyGrant elements define access to the Store, Customers, and Gender hierarchies. Users with this role cannot see any of the gender hierarchy. The other two HierarchyGrant elements define custom access to the Store and Customers hierarchies.

The member grants in this example use substitution variables like %{State} to represent substitutions from the profile attribute table. These particularize the grant to the specific user associated with the StateManager role. For example, suppose user John has StateManager as one of his roles. Also suppose that the profile attribute table has a row in which the values of the columns attribute and values are State, and CA,OR,WA. If you create a view for John, the line:

```
<MemberGrant member="[Customers].[USA].[%{State}]"
  access="all"/>
```

in the above example is interpreted to mean:

```
<MemberGrant member="[Customers].[USA].[CA,OR,WA]"
```

```
access="all"/>
```

3.4.7.3 Uploading an Access Grant Schema

To upload an access grant schema (AGXML):

1. Click **View > Repository**.

The repository page appears.

2. In the **Folders** panel, navigate to **Organization > Organization > Analysis Components > Analysis Schemas**.
3. Right-click the folder and select **Add Resource > File > Access Grant Schema**.

The Upload a File From Your Local Computer page appears and prompts you to select a file and set its properties.

Figure 3-43 Upload a File From Your Local Computer - Access Grant Schema

4. Under **Path to File**, click **Choose File** and locate the access grant schema you want to add.
5. Enter a name and description for the schema.

The **Resource ID** is auto-generated as you type in the **Name** field. You can change the ID if necessary.

6. To select a different folder to hold the schema, click **Browse** next to the **Save Location** field and navigate to the location in the repository where you want the file to reside.
7. Click **Submit**.

The new file appears in the repository.

3.4.7.4 Working with Profile Attributes

Profile attributes can be defined for each JasperReports Server user. They are used to categorize or tag users in order to define security rules around data. Administrators can view and edit profile attributes on the **Manage > Users** page. For more information on profile attributes, refer to the *JasperReports Server Administrator Guide*. For a detailed example of data-level security, including a complete example of cube security based on profile attributes and roles, refer to the *Jaspersoft OLAP Ultimate Guide*.

3.4.7.5 Best Practices for Designing Access Control

Best practices for designing access control will emerge as you gain experience with this feature. Here are some suggestions to get you started:

- Review the detailed implementation example found in the *Jaspersoft OLAP Ultimate Guide* to understand the entire implementation process.
- Start with a clear description of the requirements before writing XML.
- Determine the different roles played by users of your application, and assign specific roles to each user.
- Create a hierarchy in each cube that corresponds in a one-to-one way with the different levels of access. That is, make a hierarchical level that corresponds to a user role. Defining the access control requirements before building the schema is much easier than adding access control after designing the schema.
- Build your access control definition iteratively. Start with the smallest definition that you can test. When it works properly, add and test additional increments until you have a complete definition.
- Use an XML editor to help you format access grants properly.

3.5 Performance Tuning

For a simple application with a relatively small dataset for which performance is not critical, (for example, a small intranet application or developer environment), the simplest server configuration is to run a single application server that connects to a local database, which contains both the data being analyzed (that is, the operational data store) and the JasperReports Server repository database. However, this configuration doesn't perform under a large load. This section describes steps you can take to improve performance for larger implementations and data sets.

For larger analysis applications, the ROLAP database is often a performance bottleneck. In this case, dedicate a computer to host the ROLAP database, and configure JasperReports Server to access it. The database and its host should be optimized for disk read and write. This separation of the application from the data is sufficient for many implementations.

To improve performance further, separate the Jaspersoft OLAP user interface from other elements. Inside Jaspersoft OLAP, the application that visualizes your results and provides navigation (JPivot) competes for resources with the analysis engine (Mondrian). To separate these processing resources, run two different instances of Jaspersoft OLAP: one that handles JPivot's tasks, and another that handles Mondrian's. The former instance is called the XML/A client; the latter is called the XML/A provider (or XML/A server). In the XML/A provider, define Mondrian connections that point to your operational data store. Then define XML/A connections in the XML/A client that point to Mondrian connections in the XML/A provider.

In this configuration, your users connect to the XML/A client, which sends XML/A requests to the XML/A provider, which in turn connects to the database (using a Mondrian connection) to retrieve the results; it returns the results to the XML/A client. The XML/A client then provides visualization and navigation for your end users.

In this configuration, Mondrian and JPivot don't compete for resources, and performance in large implementations is improved. For more information, see **"Working with XML/A Connections" on page 65** and **"Working with XML/A Sources" on page 70**. More detailed information about performance tuning, including load-balancing multiple Jaspersoft OLAP instances in a single deployment, is also found in the *Jaspersoft OLAP Ultimate Guide*.



When connecting to a remote server using XML/A, the performance may be impaired the first time the data is loaded into the cache. The cache is loaded the first time data is requested from the remote server, and is subsequently refreshed based on your configuration. For details, refer to [“Configuring the XML/A Cache” on page 48](#).

3.6 Troubleshooting Jaspersoft OLAP

This section describes some issues you may encounter once Jaspersoft OLAP is installed and your cubes and views have been built. For troubleshooting instructions regarding installation, refer to the *JasperReports Server Installation Guide*. For troubleshooting instructions regarding JasperReports Server, refer to the *JasperReports Server Administrator Guide*.

This section describes:

- **Logging**
- **Performance**
- **Drill-through Behavior for Dimensions with Parent-child Hierarchies**
- **XML/A-based Sample Views and Reports Fail**
- **404 Error When WebLogic Hosts Jaspersoft OLAP**

3.6.1 Logging

OLAP logging in JasperReports Server falls into two categories:

- General logging of MDX and SQL via log4j
- XML/A logging for instances with multiple organizations
- Ad Hoc view and report logging

This section describes configuration of the first two logging categories; for information about logging for Ad Hoc views and reports, refer to the *JasperReports Server Administrator Guide*.

3.6.1.1 General Logging of MDX and SQL by log4j

Jaspersoft OLAP’s fundamental logging is controlled through log4j settings. log4j’s statement logging occurs in the `mondrian.mdx` and `mondrian.sql` log4j categories. These categories log the statements and how long they ran. The SQL log also records the number of results returned in the result set.

To log your MDX and SQL queries, edit the `/WEB-INF/log4j.properties` file and uncomment these lines:

```
#log4j.logger.mondrian.mdx=debug, jasperanalysis
#log4j.logger.mondrian.sql=debug, jasperanalysis
```

To log all possible Mondrian debugging information, add this line to the `log4j.properties` file:

```
log4j.logger.mondrian=debug
```

Adding this property also ensures that Jaspersoft OLAP logs all SQL and MDX queries.

To log the SQL generated when users drill-through to the underlying transactional data, add this line to the `log4j.properties` file:

```
log4j.logger.jasperanalysis.drillthroughSQL=DEBUG,jasperanalysis
```



You can also use JasperReports Server's audit Domains and reports to log user activity. For more information, refer to the *JasperReports Server Administrator Guide*.

3.6.1.2 XML/A Logging for Instances with Multiple Organizations

For server instances that have multiple organizations, XML/A logging is configured separately in the JasperReports Server web UI. Messages from this logger are written to the default log file is WEB-INF\logs\jasperserver.log. To enable this logger, you must enter the correct classname.

To add a logger to the page from the web interface:

1. Log in as system administrator (superuser by default).
2. Select **Manage > Server Settings** and choose **Log Settings** in the left-hand panel.
3. Scroll to the bottom of the page.
4. Enter the logger's classname in the text field. See the other properties on the page for guidance, for example:
`com.jaspersoft.ji.ja.security.service.MTXmlaServletImpl`
5. Use the drop-down to set the logging level.

The logger setting is persistent even when the server is restarted. However, the logger setting may not appear on the Log Settings page again. For information about adding loggers to this page permanently,

3.6.2 Performance

Jaspersoft offers a number of tools to help you improve the performance of views and reports based on OLAP data:

- Many of the options on the OLAP Settings page can improve performance. See [“Changing OLAP Settings” on page 38](#).
- Caching can significantly improve performance, especially when you can configure it to handle your particular data load gracefully. Caching is discussed in:
 - [“Changing OLAP Settings” on page 38](#)
 - [“Flushing the OLAP Cache” on page 46](#)
 - [“Configuring the XML/A Cache” on page 48](#).
- To help you understand bottlenecks and other performance problems, Jaspersoft offers a number of OLAP views and report that capture information about how the OLAP engine is performing against your views and reports. See [“Performance Tuning” on page 78](#) and the *Jaspersoft OLAP Ultimate Guide*.

3.6.3 Drill-through Behavior for Dimensions with Parent-child Hierarchies

When a user drills through an aggregated value in their OLAP data, JasperReports Server returns measure information for that member and for all of the members below it, with the notable exception of dimensions with parent-child hierarchies. Because of a defect in the underlying OLAP engine, drill-through for dimensions with parent-child hierarchies behaves unexpectedly. The behavior varies, depending on whether you access the data through Ad Hoc views or Jaspersoft OLAP views:

- In the Ad Hoc Editor, drill-through is prevented. When a crosstab includes a dimension with parent-child hierarchy, and a user clicks the drill-through link, the server returns a message indicating that drill-through is disabled.

- In Jaspersoft OLAP, drill-through returns measures for the current member, but not for members below it in the parent-child hierarchy. Since the data that is returned is partial, it isn't reliable.

Because of this issue, Jaspersoft recommends that you avoid using parent-child hierarchies in your dimensions. If you must use parent-child hierarchies, Jaspersoft recommends that you access such dimensions only through Ad Hoc views.

For more information about Ad Hoc views, refer to the *JasperReports Server User Guide*. For more information about the underlying issue, see Mondrian's issue tracker at <http://jira.pentaho.com/browse/MONDRIAN-388>.

3.6.4 XML/A-based Sample Views and Reports Fail

Some of JasperReports Server's sample views and reports demonstrate the ability to retrieve data using the XML/A protocol by loading data from the sample Foodmart and Sugar XML/A connections. Because XML/A client connections have to specify a port number in their URI value, the port numbers are hard-coded to the default for Apache Tomcat (8080). If your application server uses a different port number, these reports and views fail. For example, if WebSphere hosts your (commercial edition) server, and it uses its default port number (9080), the sample views and reports based on XML/A connections fail. Another example where this failure occurs is if you used a non-default port number for Apache Tomcat.

To use these samples, you must edit the XML/A connections and enter the correct port number. For instructions, refer to the *JasperReports Server Installation Guide*.

Note that the XML/A connection samples are contrived, in that they retrieve data from the local server, when XML/A is meant to retrieve data from remote OLAP sources.

3.6.5 404 Error When WebLogic Hosts Jaspersoft OLAP



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.

If Jaspersoft OLAP is hosted by WebLogic, you may encounter 404 errors when changing the data cube in your OLAP views. To fix the issue, you must update your WebLogic configuration and delete files from your file system. For more information, refer to the *JasperReports Server Installation 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 iReport, 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 iReport and Jaspersoft Studio without loss, and can be scheduled and added to dashboards.

Ad Hoc View

A view of data that is based on a Domain, Topic, or OLAP client connection. An Ad Hoc view can be a table, chart, or crosstab and is the entry point to analysis operations such as slice and dice, drill down, and drill through. [Compare OLAP View](#). You can save an Ad Hoc view as a report in order to edit it in the interactive viewer, schedule it, or add it to a dashboard.

Aggregate Function

An aggregate function is one that is computed using a group of values; for example, Sum or Average. Aggregate functions can be used to create calculated fields in Ad Hoc views. Calculated fields containing aggregate functions cannot be used as fields or added to groups in an Ad Hoc view and should not be used as filters. Aggregate functions allow you to set a level, which specifies the scope of the calculation; level values include Current (not available for PercentOf), ColumnGroup, ColumnTotal, RowGroup, RowTotal, Total.

Analysis View

[See OLAP View](#).

Audit Archiving

To prevent audit logs from growing too large to be easily accessed, the installer configures JasperReports Server to move current audit logs to an archive after a certain number of days, and to delete logs in the archive after a certain age. The archive is another table in the JasperReports Server's repository database.

Audit Domains

A Domain that accesses audit data in the repository and lets administrators create Ad Hoc reports of server activity. There is one Domain for current audit logs and one for archived logs.

Audit Logging

When auditing is enabled, audit logging is the active recording of who used JasperReports Server to do what when. The system installer can configure what activities to log, the amount of detail gathered, and when to archive the data. Audit logs are stored in the same private database that JasperReports Server uses to store the repository, but the data is only accessible through the audit Domains.

Auditing

A feature of JasperReports Server Enterprise edition that records all server activity and allows administrators to view the data.

Calculated Field

In an Ad Hoc view or a Domain, a field whose value is calculated from a user-defined formula that may include any number of fields, operators, and constants. For Domains, a calculated field becomes one of the items to which the Domain's security file and locale bundles can apply. There are more functions available for Ad Hoc view calculations than for Domains.

CRM

Customer Relationship Management. The practice of managing every facet of a company's interactions with its clientele. CRM applications help businesses track and support their customers.

CrossJoin

An MDX function that combines two or more dimensions into a single axis (column or row).

Cube

The basis of most OLAP applications, a cube is a data structure that contains three or more dimensions that categorize the cube's quantitative data. When you navigate the data displayed in an OLAP view, you are exploring a cube.

Custom Field

In the Ad Hoc Editor, a field that is created through menu items as a simple function of one or two available fields, including other custom fields. When a custom field becomes too complex or needs to be used in many reports, it is best to define it as a calculated field in a Domain.

Dashboard

A collection of reports, input controls, graphics, labels, and web content displayed in a single, integrated view. Dashboards often present a high level view of your data, but input controls can parametrize the data to display. For example, you can narrow down the data to a specific date range. Embedded web content, such as other web-based applications or maps, make dashboards more interactive and functional.

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.

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 > Ad Hoc Settings**.

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.

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

A dashboard element that displays reports or custom URLs. Frames can be mapped to input controls if their content can accept parameters.

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.

iReport Designer

An open source tool for graphically designing reports that leverage all features of the JasperReports Library. The Jaspersoft iReport Designer 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 iReport, or upload it to JasperReports Server. iReport is implemented in NetBeans.

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.

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.

JasperReports Library

An embeddable, open source, Java API for generating a report, filling it with current data, drawing charts and tables, and exporting to any standard format (HTML, PDF, Excel, CSV, and others). JasperReports processes reports defined in JRXML, an open XML format that allows the report to contain expressions and logic to control report output based on run-time data.

JasperReports Server

A commercial open source, server-based application that calls the JasperReports Library to generate and share reports securely. JasperReports Server authenticates users and lets them upload, run, view, schedule, and send reports from a web browser. Commercial versions provide metadata layers, interactive report and dashboard creation, and enterprise features such as organizations and auditing.

Jaspersoft ETL

A graphical tool for designing and implementing your data extraction, transforming, and loading (ETL) tasks. It provides hundreds of data source connectors to extract data from many relational and non-relational systems. Then, it schedules and performs data aggregation and integration into data marts or data warehouses that you use for reporting.

Jaspersoft OLAP

A relational OLAP server integrated into JasperReports Server that performs data analysis with MDX queries. The product includes query builders and visualization clients that help users explore and make sense of multidimensional data. Jaspersoft OLAP also supports XML/A connections to remote servers.

Jaspersoft Studio

An open source tool for graphically designing reports that leverage all features of the JasperReports Library. Jaspersoft Studio lets you drag and drop fields, charts, and sub-reports onto a canvas, and also define parameters or expressions for each object to create pixel-perfect reports. You can generate the JRXML of the report directly in Jaspersoft Studio, or upload it to JasperReports Server. Jaspersoft Studio is implemented in Eclipse.

JavaBean

A reusable Java component that can be dropped into an application container to provide standard functionality.

JDBC

Java Database Connectivity. A standard interface that Java applications use to access databases.

JNDI

Java Naming and Directory Interface. A standard interface that Java applications use to access naming and directory services.

Join Tree

In Domains, a collection of joined tables from the actual data source. A join is the relational operation that associates the rows of one table with the rows of another table based on a common value in given field of each table. Only the fields in a same join tree or calculated from the fields in a same join tree may appear together in a report.

JPivot

An open source graphical user interface for OLAP operations. For more information, visit <http://jpivot.sourceforge.net/>.

JRXML

An XML file format for saving and sharing reports created for the JasperReports Library and the applications that use it, such as iReport Designer and JasperReports Server. JRXML is an open format that uses the XML standard to define precisely all the structure and configuration of a report.

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 manage 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 helpdesk tickets. Such outliers may indicate a problem (or an important achievement) in your business. The analysis features of Jaspersoft OLAP excel at revealing outliers.

Parameter

Named values that are passed to the engine at report-filling time to control the data returned or the appearance and formatting of the report. A report parameter is defined by its name and type. In JasperReports Server, parameters can be mapped to input controls that users can interact with.

Pivot

To rotate a crosstab such that its row groups become column groups and its column groups become rows. In the

Ad Hoc Editor, pivot a crosstab by clicking  .

Pivot Table

A table with two physical dimensions (for example, X and Y axis) for organizing information containing more than two logical dimensions (for example, PRODUCT, CUSTOMER, TIME, and LOCATION), such that each physical dimension is capable of representing one or more logical dimensions, where the values described by the dimensions are aggregated using a function such as SUM. Pivot tables are used in Jaspersoft OLAP.

Properties

Settings associated with an object. The settings determine certain features of the object, such as its color and label. Properties are normally editable. In Java, properties can be set in files listing objects and their settings.

Report

In casual usage, *report* may refer to:

- A JasperReport. [See JasperReport.](#)
- The main JRXML in a JasperReport.
- The file generated when a JasperReport is scheduled. Such files are also called content resources or output files.
- The file generated when a JasperReport is run and then exported.
- In previous JasperReports Server versions, a report created in the Ad Hoc Editor. [See Ad Hoc Report.](#)

Report Run

An execution of a report, Ad Hoc view, or dashboard, or a view or dashboard designer session, it measures and limits usage of Freemium instances of JasperReports Server. The executions apply to resources no matter how they are run (either in the web interface or through the various APIs, such as REST web services). Users of our Community Project and our full-use commercial licenses are not affected by the limit. For more information, please contact sales@jaspersoft.com.

Repository

The tree structure of folders that contain all saved reports, dashboards, OLAP views, and resources. Users access the repository through the JasperReports Server web interface or through iReport. Applications can access the repository through the web service API. Administrators use the import and export utilities to back up the repository contents.

Resource

In JasperReports Server, anything residing in the repository, such as an image, file, font, data source, Topic, Domain, report element, saved report, report output, dashboard, or OLAP view. Resources also include the folders in the repository. Administrators set user and role-based access permissions on repository resources to establish a security policy.

Role

A security feature of JasperReports Server. Administrators create named roles, assign them to user accounts, and then set access permissions to repository objects based on those roles. Certain roles also determine what functionality and menu options are displayed to users in the JasperReports Server interface.

Schema

A logical model that determines how data is stored. For example, the schema in a relational database is a description of the relationships between tables, views, and indexes. In Jaspersoft OLAP, an OLAP schema is the logical model of the data that appears in an OLAP view; they are uploaded to the repository as resources. For Domains, schemas are represented in XML design files.

Schema Workbench

A graphical tool for easily designing OLAP schemas, data security schemas, and MDX queries. The resulting cube and query definitions can then be used in Jaspersoft OLAP to perform simple but powerful analysis of large quantities of multi-dimensional data stored in standard RDBMS systems.

Set

In Domains and Domain Topics, a named collection of items grouped together for ease of use in the Ad Hoc Editor. A set can be based on the fields in a table or entirely defined by the Domain creator, but all items in a set must originate in the same join tree. The order of items in a set is preserved.

Slice

An OLAP operation for filtering data rows.

SQL

Structured Query Language. A standard language used to access and manipulate data and schemas in a relational database.

System Admin

Also called the system administrator. A user who has unlimited access to manage all organizations, users, roles, repository permissions, and repository objects across the entire JasperReports Server instance. The system admin can create root-level organizations and manage all server settings. The default system admin is the `superuser` account.

Topic

A JRXML file created externally and uploaded to JasperReports Server as a basis for Ad Hoc reports. Topics are created by business analysts to specify a data source and a list of fields with which business users can create reports in the Ad Hoc Editor. Topics are stored in the Ad Hoc Components folder of the repository and displayed when a user launches the Ad Hoc Editor.

Transactional Data

Data that describe measurable aspects of an event, such as a retail transaction, relevant to your business. Transactional data are often stored in relational databases, with one row for each event and a table column or field for each measure.

User

Depending on the context:

- A person who interacts with JasperReports Server through the web interface. There are generally three categories of users: administrators who install and configure JasperReports Server, database experts or business analysts who create data sources and Domains, and business users who create and view reports and dashboards.
- A user account that has an ID and password to enforce authentication. Both people and API calls accessing the server must provide the ID and password of a valid user account. Roles are assigned to user accounts to determine access to objects in the repository.

View

Several meanings pertain to JasperReports Server:

- An Ad Hoc view. [See Ad Hoc View.](#)
- An OLAP view. [See OLAP View.](#)
- A database view. See http://en.wikipedia.org/wiki/View_%28database%29.

Virtual Data Source

A virtual data source allows you to combine data residing in multiple JDBC and/or JNDI data sources into a single data source that can query the combined data. Once you have created a virtual data source, you create Domains that join tables across the data sources to define the relationships between the data sources.

WCF

Web Component Framework. A low-level GUI component of JPivot. For more information, see <http://jpivot.sourceforge.net/wcf/index.html>.

Web Services

A SOAP (Simple Object Access Protocol) API that enables applications to access certain features of JasperReports Server. The features include repository, scheduling and user administration tasks.

XML

eXtensible Markup language. A standard for defining, transferring, and interpreting data for use across any number of XML-enabled applications.

XML/A

XML for Analysis. An XML standard that uses Simple Object Access protocol (SOAP) to access remote data sources. For more information, see <http://www.xmla.org/>

XML/A Connection

A type of OLAP client connection that consists of Simple Object Access Protocol (SOAP) definitions used to access data on a remote server. OLAP client connections populate OLAP views.

INDEX

A

- access control
 - access grant definitions 73
 - AGXML files 73
 - and OLAP schemas 74
 - best practices 78
 - elements 74
- Ad Hoc views 5-6
- administering Jaspersoft OLAP
 - ETL process 24
 - implementing 24
 - maintaining 24
 - OLAP views 25
 - security 71
 - servers 78
- Adobe Acrobat 11
- aggregate settings 42

B

- bibliography 6

C

- cache. See OLAP cache. 46
- connections
 - editing 60, 66, 69
 - Mondrian 25, 53, 65, 73
 - Mondrian and XML/A sources 72
 - XML/A 33, 66, 70, 72-73
 - XML/A, testing 68-69

creating

- Mondrian connections 54
- OLAP views 25
- XML/A connections 66
- XML/A sources 72
- CSV format 16
- CubeGrant 74

D

- data sources
 - editing 50
 - types 51
- displaying OLAP views. See OLAP views. 17
- drill-through behavior when using parent-child relationships 80
- drill-through table options
 - Edit Properties 15
 - Output as CSV 16
 - Page Controls 16

E

- editing
 - data sources 50
 - Mondrian connections 60
 - OLAP views 35
 - XML/A connections 69
 - XML/A sources 73
- Excel 11, 16
- exporting OLAP views 11

F

- flushing the OLAP cache 46

Foodmart Sample Analysis View 17

further reading 6

G

GlobalPropertiesList 46

H

HierarchyGrant 75

J

Jaspersoft OLAP

- administering 71, 78

- logging 38

- prerequisites 5

- tools 5

- Workbench 6

JPivot 23

L

logging 38, 79

M

MDX queries

- defining 32

- displaying 11

- editing 11

- example 33, 35

- logging 79

- query strings 35

MemberGrant 75

memory monitoring settings 45

Microsoft SQL Server Analytic Services 24, 65-67

Mondrian

- debugging information 79

- XML/A sources 72-73

Mondrian connections

- creating 54

- creating OLAP views with 25

- editing 60

- understanding 72

- working with 53, 65

N

navigation options

- Expand/Collapse Member 15-16

- Zoom In/Out 15

- Zoom Out All 15

navigation table 14

O

ODBO Connect 6

ODBO driver 71

OLAP-based reports 6

OLAP cache

- flushing 46

- managing 46

- settings 43

OLAP defined 9

OLAP views

- administering 25

- and themes 23

- compared to Ad Hoc views 6

- connections 25, 33

- creating 25, 33

- displaying 17

- editing 35

- exporting 11

- folder 16

- opening 9

- output 11, 16

- overview 25

- samples 9

- saving 11, 13, 22

organizations 70

output of OLAP views 11, 16

P

parent-child relationships in hierarchies 80

PDF 11

performance 38

pivot tables 6, 71

prerequisites for Jaspersoft OLAP 5

profile attributes 77

properties, engine behavior 38

properties, OLAP

- aggregate settings 42

- Cache and SQL Settings 43

- General Behavior 39

- Memory Monitoring Settings 45

- XML/A settings 45

Public folder 70

Q

queries

- defining 36
- editing 37
- language 23
- limiting 39
- logging 38
- query strings 35
- results 39
- SQL 38
- timing out 40

R

ROLAP 23

S

samples

- Foodmart 17
- OLAP views 9

saving OLAP views. See OLAP views. 22

SchemaGrant 74

schemas, for data security 74, 76

schemas, OLAP

- changing file source 53
- creating 52
- editing 53
- Locate OLAP Schema page 61
- settings 38
- uploading 52

security

- data level 74
- HTTP authentication 71
- XML/A 71

SQL

- logging queries 38, 79
- property settings 43

superuser 70

T

themes 23

tool bar, OLAP

- Change Data Cube 10
- Cube Options 12
- Drill-through Operations 12
- Drill into a Dimension Member 10
- Edit Chart Options 10

Edit Display Option 10-11

Edit Output Operations 11

Export to Excel 11

Export to PDF 11

Hierarchical Ascending and Descending 15

Natural Order 15

overview 10

Show Chart 10

Show Empty Rows & Columns 10

Sort Across Hierarchy 10

Swap Axes 10

tools for analysis 5

troubleshooting Jaspersoft OLAP 79, 81

V

views. See OLAP views. 9

X

XML/A

and organizations 70

and superuser 70

configuration 70

connections 33, 65, 73

creating an XML/A source 73

creating OLAP views with 33

definition 72

editing an XML/A connection 69

editing source properties 73

properties 45

security 71

settings 45

source 72

