

# **JasperAnalysis Community Edition User's Guide**

## **Version 3.5**

**<http://www.jaspersoft.com/>**

© 2009 JasperSoft Corporation. All rights reserved. Printed in the U.S.A. JasperSoft, the JasperSoft logo, JasperAnalysis, JasperServer, JasperETL, JasperReports, iReport, and Jasper4 products 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.

# Table of Contents

<b>1</b>	<b>Introduction to JasperAnalysis Professional.....</b>	<b>4</b>
1.1	Login.....	4
1.2	External Sources of Information.....	6
1.3	Jaspersoft ODBO Connect.....	6
<b>2</b>	<b>Working with Analysis Views.....</b>	<b>7</b>
2.1	Opening an Analysis View .....	7
2.2	Tool Bar.....	8
2.2.1	Drill into a Dimension Member.....	8
2.2.2	Sort Across Hierarchy.....	9
2.2.3	Natural Order .....	9
2.2.4	Hierarchical Ascending and Descending.....	9
2.2.5	Show Empty Rows & Columns .....	9
2.2.6	Swap Axes .....	10
2.2.7	Edit Display Option .....	10
2.2.7.1	Cube Options .....	11
2.2.7.2	Drill-through Operations.....	12
2.2.7.3	Sort Option.....	12
2.2.8	Show Chart .....	12
2.2.9	Edit Chart Options.....	13
2.2.10	Change Data Cube .....	13
2.2.11	Show MDX Query .....	14
2.2.12	Export to Excel.....	15
2.2.13	Export to PDF.....	15
2.2.14	Edit Output Operations .....	16
2.2.15	Save Buttons.....	17
2.3	Navigation Table .....	17
2.3.1	Expand/Collapse Position .....	17
2.3.2	Expand/Collapse Member.....	17
2.3.3	Zoom In/Out .....	18
2.3.4	Show Source Data (Drill-through).....	18
2.3.5	Expand All/Collapse All.....	19
2.3.6	Zoom Out All .....	19
2.4	Drill-through Table.....	20

---

2.4.1	Edit Properties.....	20
2.4.2	Output as CSV .....	20
2.4.3	Page Controls .....	20
2.5	Saving an Analysis View .....	20
<b>3</b>	<b>JasperAnalysis Administration.....</b>	<b>21</b>
3.1	Overview .....	21
3.2	Working with an Analysis Views.....	22
3.2.1	Overview of an Analysis View.....	23
3.2.2	Creating an Analysis View with a Mondrian Connection .....	23
3.2.3	Creating an Analysis View with an XML/A Connection .....	27
3.2.4	Editing an Analysis View.....	30
3.3	Working with Analysis Options.....	31
3.3.1	Viewing Analysis Properties.....	31
3.3.2	Flushing the OLAP Cache .....	37
3.4	Editing a Mondrian Connection .....	37
3.5	Editing an XML/A Connection .....	39
3.6	Editing an Analysis Schema.....	39
3.7	Editing a Data Source .....	40
3.9	XML/A Configuration .....	41
3.9.1	Creating a XML/A Source .....	41
3.9.2	Editing an XML/A Source.....	42
3.10	Technical Considerations.....	43
3.10.1	XML/A Security .....	43
3.10.2	Performance Tuning .....	43
3.11	Troubleshooting .....	44
3.11.1	SugarCRM Sample Analysis View Fails in WebSphere .....	44
3.12	Logging.....	45
<b>4</b>	<b>Glossary .....</b>	<b>46</b>

# 1 Introduction to JasperAnalysis Professional

Jaspersoft 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. Jaspersoft exposes comprehensive public integration interfaces enabling seamless integration with other applications and the ability to easily add custom functionality.

The heart of the Jaspersoft BI Suite is JasperServer. For business intelligence users, Jaspersoft offers JasperAnalysis, which runs on JasperServer and helps you:

- Make well informed business decisions by identifying trends, anomalies, and correlations in your data.
- Quickly and easily access your OLAP (On Line Analytical Processing) data through an intuitive web interface.
- Explore data by drilling, pivoting, filtering, visualizing, and defining alerts based on data values.

To use JasperAnalysis, you must supply analysis schemas in XML format, corresponding to data sources (such as JDBC or JNDI), and MDX queries. The built-in JasperAnalysis user interface assembles the schemas, data source, and MDX queries into interactive analysis views. Note that the application doesn't generate schemas or MDX queries.

This introduction includes the following sections:

- [Login](#)
- [External Sources of Information](#)
- [Jaspersoft ODBO Connect](#)

## 1.1 Login

To login to JasperAnalysis Professional:

1. Enter `http://hostname:8080/jasperserver-pro` in your browser (where `hostname` is the name of the computer where JasperServer is installed).

The Login page appears.



2. Before logging in, you may want to review the links that appear on the login page.
3. To login, enter your user name and password.

If you installed the sample data, you can use any of the user names and passwords shown, as well as your own. The default administrator user is `jasperadmin`; its password is set to `jasperadmin` during installation.

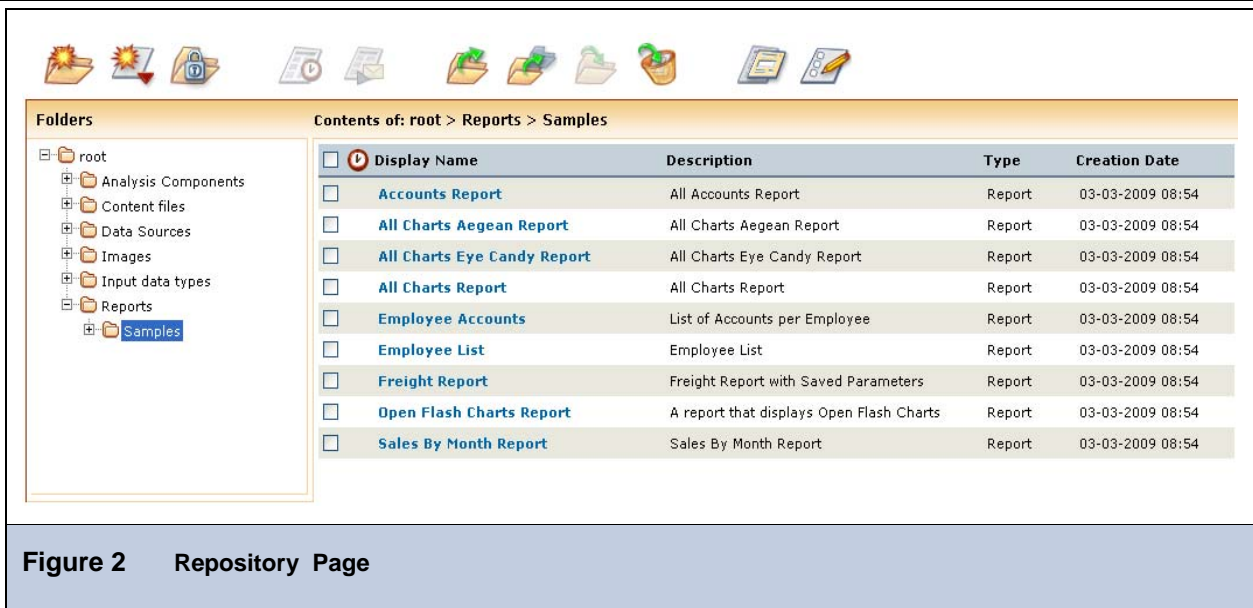
If the **Organization** field appears, enter the name of your organization. If you don't know it, refer to your administrator.

4. Click **Show locale & timezone**.
- The Locale controls appear in the Login pane.
5. Select your locale and time zone from the drop-downs.
  6. Click **Login**.



If the **Change password** link appears, you can change your password by clicking it. For more information about this configuration option, refer to the *JasperServer Professional Administrator Guide*.

The Repository page appears.



This guide only describes JasperAnalysis functionality. For more information on Home pages and on JasperServer in general, refer to the *JasperServer Professional User Guide*.

## 1.2 External Sources of Information

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://www.jasperforge.org>
- JasperAnalysis Workbench: <http://sourceforge.net>
- 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://en.wikipedia.org/wiki/Multidimensional\\_Expressions](http://en.wikipedia.org/wiki/Multidimensional_Expressions)
- A collection of other MDX article links: <http://www.mosha.com/msAnalysis/mdx.htm>
- Official XML/A home: <http://www.xmla.org/>

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

## 1.3 Jaspersoft ODBO Connect

Jaspersoft ODBO Connect enables Microsoft Excel Pivot Tables to access JasperAnalysis and other OLAP servers that support the XML for Analysis (XML/A) 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..

## 2 Working with Analysis Views

JasperAnalysis is an OLAP (On Line Analytical Processing) application. OLAP applications help users analyze and understand complex data. JasperAnalysis Community edition uses JPivot for analytical processing while JasperAnalysis Professional provides an improved user interface to the processing.



The procedures in this document assume that you installed JasperServer with its sample data. The schemas and views described in this document are only available if the samples are installed.

### 2.1 Opening an Analysis View

An analysis view consists of a connection and an MDX query. A connection consists of a data source, a schema, and an optional, an access control definition. JasperAnalysis contains a number of sample analysis views. Because the access control definition can provide different access to users with different roles, the view that you see depends on the permissions that apply to your roles. If you don't see something you expect to see, check with your system administrator to ensure that you have the right access control profile. This guide assumes you are logged in as a user with the administrator role.

To open an analysis view:

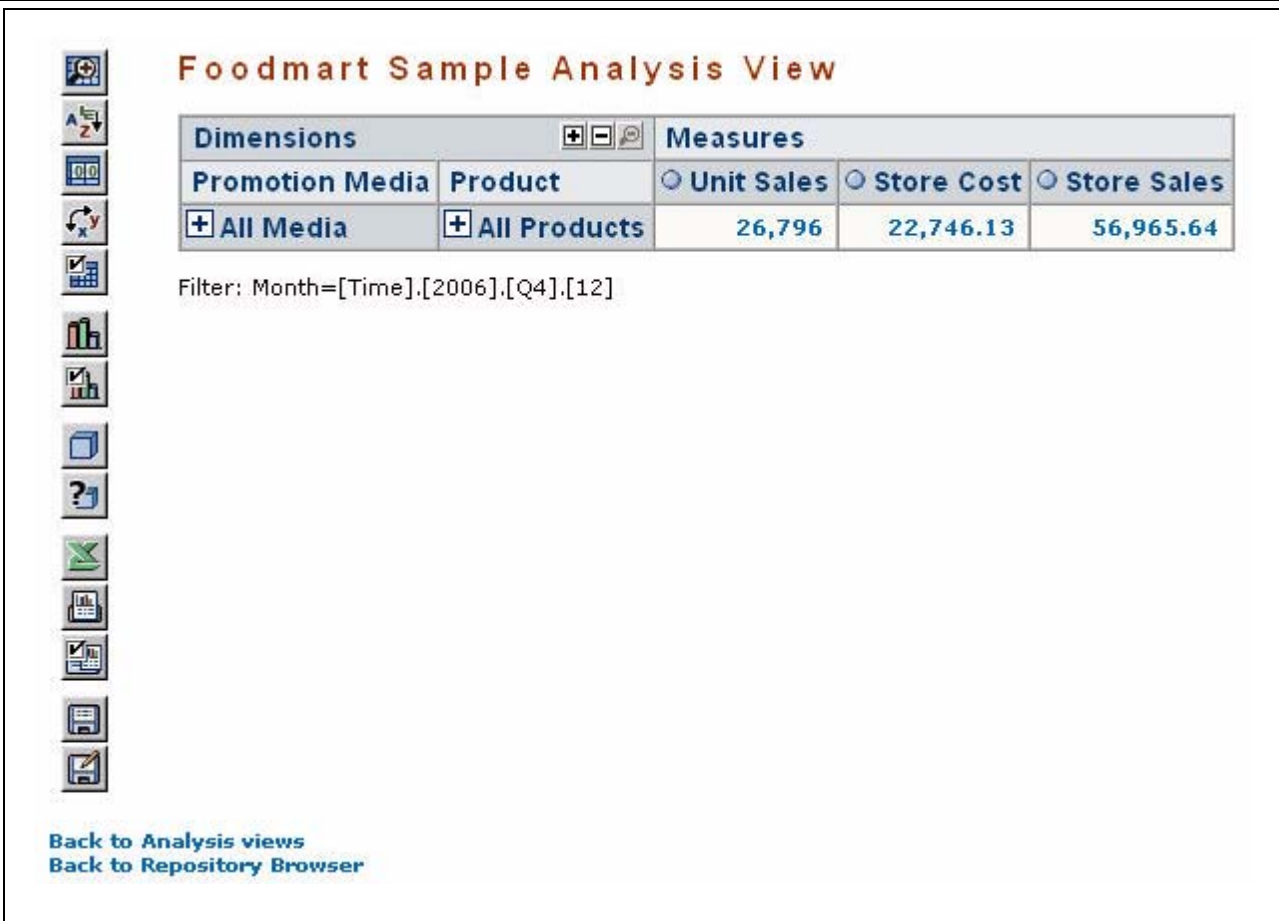
1. On the Home page, click **View > Analysis**.

The Analysis Views page appears.

Analysis View Name	Description	Date	Parent Folder
<a href="#">Foodmart Sample Analysis View</a>	Sample Analysis View: 2006 Q1 Foodmart Unit Sales	04-25-2008 09:32	/analysis/views
<a href="#">Foodmart Sample XMLA Analysis View</a>	Sample XMLA Analysis View: 2006 Q1 Foodmart Unit Sales	04-25-2008 09:32	/analysis/views
<a href="#">SugarCRM Sample view</a>	SugarCRM Sample Analysis View: Sales Performance by Industry/Account	04-25-2008 09:32	/analysis/views
<a href="#">SugarCRM Sample XMLA Analysis View</a>	Sample SugarCRM Analysis View (XMLA): Sales Performance by Industry/Account	04-25-2008 09:32	/analysis/views
<a href="#">Profile view</a>	Simple Profiling View	04-25-2008 09:33	/performance/views
<a href="#">Profit View</a>	Sales gauge report links to this when parameter isJasperAnalysis = "true"	04-25-2008 09:33	/supermart/revenueAndProfit
<a href="#">Revenue By Store View</a>	Revenue by store type: deluxe, supermarket, etc. Drillable pie charts.	04-25-2008 09:33	/supermart/revenueAndProfit
<a href="#">Revenue View</a>	Sales gauge report links to this when parameter isJasperAnalysis = "true"	04-25-2008 09:33	/supermart/revenueAndProfit
<a href="#">Revenue View</a>	Linked to by Sales By Month Report when parameter isJasperAnalysis = "true"	04-25-2008 09:33	/supermart/salesByMonth

**Figure 3 Analysis Views Page**

2. To display an analysis view, click its name in the Analysis View Name column. For example, click **Foodmart Sample Analysis View** to display the Foodmart sample.



**Figure 4 Foodmart Sample Analysis View**

JasperAnalysis displays the default state of the view. Click the tool bar buttons and values in the navigation table to explore the data.

## 2.2 Tool Bar

The tool bar on left side of the view (Foodmart Sample Analysis View) provides access to many JasperAnalysis operations and features.

### 2.2.1 Drill into a Dimension Member


	Zoom on Drill	Toggles (that is, turns on or off) the zoom in/out hyperlinks for hierarchy members. In the following navigation table, Zoom on Drill is selected; All Media and All Products are hyperlinks; 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	22,746.13	56,965.64





2.2.2 Sort Across Hierarchy

	Sort Across Hierarchy	Toggles between sort across and sort within hierarchy. In the following example, Sort Across Hierarchy is selected and Unit Sales is sorted in descending order across the Product hierarchy.																														
<table border="1" data-bbox="354 365 1411 611"> <thead> <tr> <th colspan="2" data-bbox="354 365 883 405">Dimensions</th> <th colspan="3" data-bbox="883 365 1411 405">Measures</th> </tr> <tr> <th data-bbox="354 405 586 445">Promotion Media</th> <th data-bbox="586 405 883 445">Product</th> <th data-bbox="883 405 1052 445">Unit Sales</th> <th data-bbox="1052 405 1224 445">Store Cost</th> <th data-bbox="1224 405 1411 445">Store Sales</th> </tr> </thead> <tbody> <tr> <td data-bbox="354 445 586 611">+ All Media</td> <td data-bbox="586 445 883 485">- All Products</td> <td data-bbox="883 445 1052 485">26,796</td> <td data-bbox="1052 445 1224 485">22,746.13</td> <td data-bbox="1224 445 1411 485">56,965.64</td> </tr> <tr> <td></td> <td data-bbox="586 485 883 525">+ Food</td> <td data-bbox="883 485 1052 525">19,356</td> <td data-bbox="1052 485 1224 525">16,562.05</td> <td data-bbox="1224 485 1411 525">41,484.40</td> </tr> <tr> <td></td> <td data-bbox="586 525 883 564">+ Non-Consumable</td> <td data-bbox="883 525 1052 564">5,021</td> <td data-bbox="1052 525 1224 564">4,258.77</td> <td data-bbox="1224 525 1411 564">10,679.21</td> </tr> <tr> <td></td> <td data-bbox="586 564 883 611">+ Drink</td> <td data-bbox="883 564 1052 611">2,419</td> <td data-bbox="1052 564 1224 611">1,925.31</td> <td data-bbox="1224 564 1411 611">4,802.03</td> </tr> </tbody> </table>			Dimensions		Measures			Promotion Media	Product	Unit Sales	Store Cost	Store Sales	+ All Media	- All Products	26,796	22,746.13	56,965.64		+ 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
Dimensions		Measures																														
Promotion Media	Product	Unit Sales	Store Cost	Store Sales																												
+ All Media	- All Products	26,796	22,746.13	56,965.64																												
	+ 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																												


2.2.3 Natural Order

	Natural Order	Natural Order means numbers in measures columns are ordered according to the order of hierarchy members.
---	---------------	--

2.2.4 Hierarchical Ascending and Descending

	Ascending	The Hierarchy Ascending and Hierarchy Descending order operations arrange the rows according to their numeric value. Hierarchy Ascending sorts from smallest to largest, while Hierarchy Descending sorts from largest to smallest.
	Descending	

2.2.5 Show Empty Rows & Columns

	Show Empty Rows/Columns	Hides or reveals rows or columns that do not have relevant fact data. The following example includes empty rows for the last four rows of Promotion Media (Street Handout; Sunday Paper; Sunday Paper, Radio, TV; and TV).
---	-------------------------	--

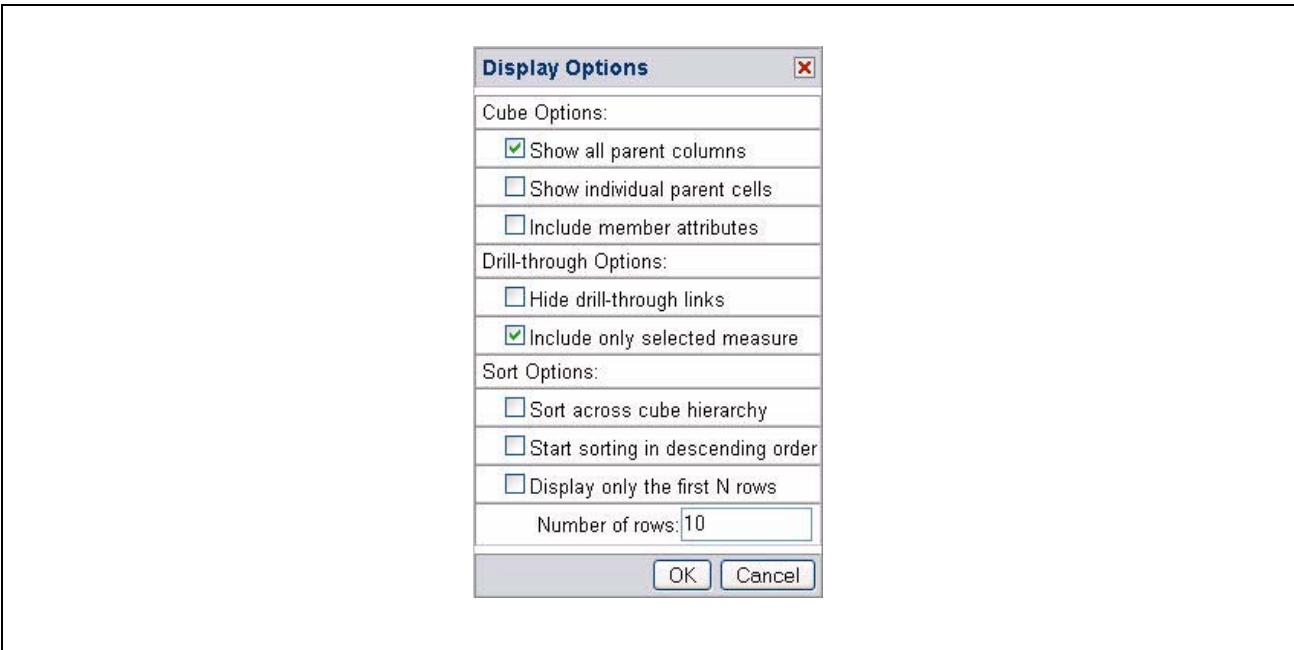
Dimensions		Measures		
Promotion Media	Product	Unit Sales	Store Cost	Store Sales
- All Media	+ All Products	26,796	22,746.13	56,965.64
No Media	+ All Products	20,818	17,695.87	44,274.03
Cash Register Handout	+ All Products	1,600	1,353.09	3,378.48
Daily Paper, Radio, TV	+ All Products	1,490	1,261.18	3,159.52
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			
TV	+ All Products			

2.2.6 Swap Axes

	Swap Axes	Changes the orientation of the table by switching the columns with the rows. In the following example, axes for measure columns and dimension rows are switched.													
<table border="1" data-bbox="638 1186 1125 1476"> <thead> <tr> <th rowspan="2">Dimensions</th> <th>Promotion Media</th> </tr> <tr> <th>All Media</th> </tr> <tr> <th>Measure</th> <th>Product</th> </tr> <tr> <th>All Products</th> <th>All Products</th> </tr> </thead> <tbody> <tr> <td>Unit Sales</td> <td>26,796</td> </tr> <tr> <td>Store Cost</td> <td>22,746.13</td> </tr> <tr> <td>Store Sales</td> <td>56,965.64</td> </tr> </tbody> </table> <p data-bbox="638 1514 797 1535">Slicer: [Month=12]</p>			Dimensions	Promotion Media	All Media	Measure	Product	All Products	All Products	Unit Sales	26,796	Store Cost	22,746.13	Store Sales	56,965.64
Dimensions	Promotion Media														
	All Media														
Measure	Product														
All Products	All Products														
Unit Sales	26,796														
Store Cost	22,746.13														
Store Sales	56,965.64														

2.2.7 Edit Display Option

	Edit Display Options	Allows users to configure the cube options, drill-through options, and sort options. These options are described in the following sections.
--	----------------------	---



2.2.7.1 Cube Options

<p>Show all parent columns</p>	<p>Displays the column headings of a given hierarchy. The following example shows Product and Product Family as parent column headings.</p>																																				
<table border="1"> <thead> <tr> <th colspan="3">Dimensions</th> <th colspan="3">Measures</th> </tr> <tr> <th>Promotion Media</th> <th>Product</th> <th>Product Family</th> <th>Unit Sales</th> <th>Store Cost</th> <th>Store Sales</th> </tr> </thead> <tbody> <tr> <td>+ All Media</td> <td>- All Products</td> <td></td> <td>26,796</td> <td>22,746.13</td> <td>56,965.64</td> </tr> <tr> <td></td> <td>All Products</td> <td>+ Drink</td> <td>2,419</td> <td>1,925.31</td> <td>4,802.03</td> </tr> <tr> <td></td> <td></td> <td>+ Food</td> <td>19,356</td> <td>16,562.05</td> <td>41,484.40</td> </tr> <tr> <td></td> <td></td> <td>+ Non-Consumable</td> <td>5,021</td> <td>4,258.77</td> <td>10,679.21</td> </tr> </tbody> </table>		Dimensions			Measures			Promotion Media	Product	Product Family	Unit Sales	Store Cost	Store Sales	+ All Media	- All Products		26,796	22,746.13	56,965.64		All Products	+ Drink	2,419	1,925.31	4,802.03			+ Food	19,356	16,562.05	41,484.40			+ Non-Consumable	5,021	4,258.77	10,679.21
Dimensions			Measures																																		
Promotion Media	Product	Product Family	Unit Sales	Store Cost	Store Sales																																
+ All Media	- All Products		26,796	22,746.13	56,965.64																																
	All Products	+ Drink	2,419	1,925.31	4,802.03																																
		+ Food	19,356	16,562.05	41,484.40																																
		+ Non-Consumable	5,021	4,258.77	10,679.21																																
<p>Show individual parent cells</p>	<p>Displays each parent member of a given hierarchy. The following example displays all parents cells for Promotion Media and Product dimensions.</p>																																				
<table border="1"> <thead> <tr> <th colspan="3">Dimensions</th> <th colspan="3">Measures</th> </tr> <tr> <th>Promotion Media</th> <th>Product</th> <th>Product Family</th> <th>Unit Sales</th> <th>Store Cost</th> <th>Store Sales</th> </tr> </thead> <tbody> <tr> <td>+ All Media</td> <td>- All Products</td> <td></td> <td>26,796</td> <td>22,746.13</td> <td>56,965.64</td> </tr> <tr> <td>+ All Media</td> <td>All Products</td> <td>+ Drink</td> <td>2,419</td> <td>1,925.31</td> <td>4,802.03</td> </tr> <tr> <td>+ All Media</td> <td>All Products</td> <td>+ Food</td> <td>19,356</td> <td>16,562.05</td> <td>41,484.40</td> </tr> <tr> <td>+ All Media</td> <td>All Products</td> <td>+ Non-Consumable</td> <td>5,021</td> <td>4,258.77</td> <td>10,679.21</td> </tr> </tbody> </table>		Dimensions			Measures			Promotion Media	Product	Product Family	Unit Sales	Store Cost	Store Sales	+ All Media	- All Products		26,796	22,746.13	56,965.64	+ All Media	All Products	+ Drink	2,419	1,925.31	4,802.03	+ All Media	All Products	+ Food	19,356	16,562.05	41,484.40	+ All Media	All Products	+ Non-Consumable	5,021	4,258.77	10,679.21
Dimensions			Measures																																		
Promotion Media	Product	Product Family	Unit Sales	Store Cost	Store Sales																																
+ All Media	- All Products		26,796	22,746.13	56,965.64																																
+ All Media	All Products	+ Drink	2,419	1,925.31	4,802.03																																
+ All Media	All Products	+ Food	19,356	16,562.05	41,484.40																																
+ All Media	All Products	+ Non-Consumable	5,021	4,258.77	10,679.21																																
<p>Include member attributes</p>	<p>Displays the member properties of the displayed hierarchy members.</p>																																				


2.2.7.2 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.

2.2.7.3 Sort Option

Sort across cube hierarchy	Changes the behavior of sorting across or within dimension hierarchies. See The connection type window appears..
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.

2.2.8 Show Chart

	Show Chart	Displays a chart of the navigation table data. The following example shows a pie chart that represents the columns in these measures: Unit Sales, Store Cost, and Store Sales.
--	------------	--



All Media.All Products.


Filter: Month=[Time].[2006].[Q4].[12]

● Unit Sales. ● Store Cost. ● Store Sales.


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].[2006].[Q4].[12]

2.2.9 Edit Chart Options


	<p><b>Edit Chart Operations</b></p>	<p>Allows you to define various charting options.</p>																																																																
<div style="border: 1px solid gray; padding: 10px; margin: 10px auto; width: 80%;"> <p><b>Chart Options</b> <span style="float: right;">✖</span></p> <table border="1" style="width: 100%; border-collapse: collapse;"> <tr> <td style="width: 40%;">Chart Type</td> <td colspan="3">Vertical Bar ▾</td> </tr> <tr> <td>Chart Size</td> <td>Height (cm): <input type="text" value="300"/></td> <td colspan="2">Width (cm): <input type="text" value="500"/></td> </tr> <tr> <td>Chart Title</td> <td colspan="3"><input type="text"/></td> </tr> <tr> <td>Horizontal Axis Label</td> <td colspan="3"><input type="text"/></td> </tr> <tr> <td>Vertical Axis Label</td> <td colspan="3"><input type="text"/></td> </tr> <tr> <td>Enable Drill-through</td> <td colspan="3"><input type="checkbox"/></td> </tr> <tr> <td>Show Legend</td> <td><input checked="" type="checkbox"/></td> <td colspan="2">Bottom ▾</td> </tr> <tr> <td>Show Filter</td> <td><input checked="" type="checkbox"/></td> <td>Bottom ▾</td> <td>Left ▾</td> </tr> <tr> <td>Chart Default Font</td> <td>SansSerif ▾</td> <td>Bold ▾</td> <td>12 ▾</td> </tr> <tr> <td>Chart Title</td> <td>----- ▾</td> <td>----- ▾</td> <td>--- ▾</td> </tr> <tr> <td>Axes Label</td> <td>----- ▾</td> <td>----- ▾</td> <td>--- ▾</td> </tr> <tr> <td>Axes Tick Label</td> <td>----- ▾</td> <td>----- ▾</td> <td>--- ▾ 30° ▾</td> </tr> <tr> <td>Legend</td> <td>----- ▾</td> <td>----- ▾</td> <td>--- ▾</td> </tr> <tr> <td>Filter</td> <td>----- ▾</td> <td>----- ▾</td> <td>--- ▾</td> </tr> <tr> <td>Background (R, G, B)</td> <td><input type="text" value="255"/></td> <td><input type="text" value="255"/></td> <td><input type="text" value="255"/></td> </tr> <tr> <td colspan="4" style="text-align: right;"> <input type="button" value="OK"/> <input type="button" value="Cancel"/> </td> </tr> </table> </div>			Chart Type	Vertical Bar ▾			Chart Size	Height (cm): <input type="text" value="300"/>	Width (cm): <input type="text" value="500"/>		Chart Title	<input type="text"/>			Horizontal Axis Label	<input type="text"/>			Vertical Axis Label	<input type="text"/>			Enable Drill-through	<input type="checkbox"/>			Show Legend	<input checked="" type="checkbox"/>	Bottom ▾		Show Filter	<input checked="" type="checkbox"/>	Bottom ▾	Left ▾	Chart Default Font	SansSerif ▾	Bold ▾	12 ▾	Chart Title	----- ▾	----- ▾	--- ▾	Axes Label	----- ▾	----- ▾	--- ▾	Axes Tick Label	----- ▾	----- ▾	--- ▾ 30° ▾	Legend	----- ▾	----- ▾	--- ▾	Filter	----- ▾	----- ▾	--- ▾	Background (R, G, B)	<input type="text" value="255"/>	<input type="text" value="255"/>	<input type="text" value="255"/>	<input type="button" value="OK"/> <input type="button" value="Cancel"/>			
Chart Type	Vertical Bar ▾																																																																	
Chart Size	Height (cm): <input type="text" value="300"/>	Width (cm): <input type="text" value="500"/>																																																																
Chart Title	<input type="text"/>																																																																	
Horizontal Axis Label	<input type="text"/>																																																																	
Vertical Axis Label	<input type="text"/>																																																																	
Enable Drill-through	<input type="checkbox"/>																																																																	
Show Legend	<input checked="" type="checkbox"/>	Bottom ▾																																																																
Show Filter	<input checked="" type="checkbox"/>	Bottom ▾	Left ▾																																																															
Chart Default Font	SansSerif ▾	Bold ▾	12 ▾																																																															
Chart Title	----- ▾	----- ▾	--- ▾																																																															
Axes Label	----- ▾	----- ▾	--- ▾																																																															
Axes Tick Label	----- ▾	----- ▾	--- ▾ 30° ▾																																																															
Legend	----- ▾	----- ▾	--- ▾																																																															
Filter	----- ▾	----- ▾	--- ▾																																																															
Background (R, G, B)	<input type="text" value="255"/>	<input type="text" value="255"/>	<input type="text" value="255"/>																																																															
<input type="button" value="OK"/> <input type="button" value="Cancel"/>																																																																		

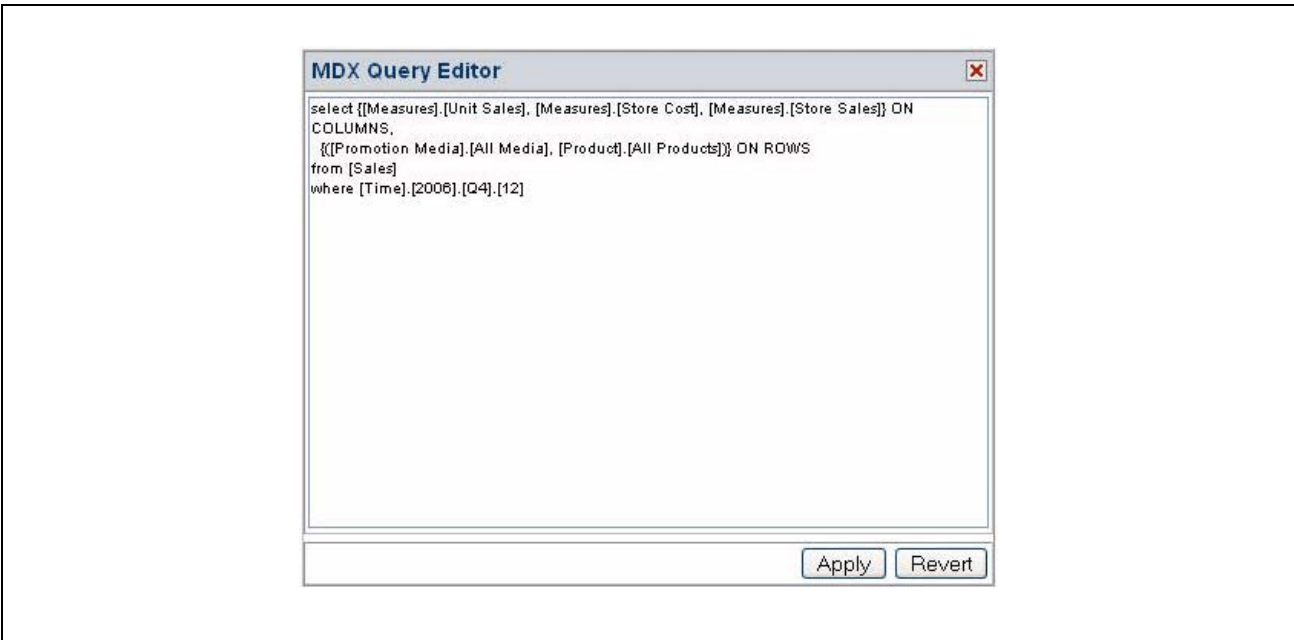
2.2.10 Change Data Cube

	<p><b>Change Data Cube</b></p>	<p>Allows you to change an analysis view and define dimension filters. Click <b>Change Data Cube</b> to display the options in the following figure.</p>
---	--------------------------------	--


		
	Columns	Allows users to select specific measures to display.
	Rows	Allows users to select specific dimension members to display.
	Filter	Allows users to select specific dimension members that don't appear in the navigation table. This selection determines the data displayed in the navigation table.

**2.2.11 Show MDX Query**

	<p>Show MDX Query</p>	<p>Allows you to change 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.</p>
---	-----------------------	---




2.2.12 Export to Excel

	Export to Excel	Prompts you to view or save the current navigation table in Microsoft Excel format.
---	-----------------	---

	A	B	C	D	E	F
1	<b>Dimensions</b>			<b>Measures</b>		
2	Promotion Media	Product	Product Family	Unit Sales	Store Cost	Store Sales
3	All Media	All Products		26,796	22,746.13	56,965.64
4		All Products	Drink	2,419	1,925.31	4,802.03
5			Food	19,356	16,562.05	41,484.4
6			Non-Consumable	5,021	4,258.77	10,679.21

2.2.13 Export to PDF

	Export to PDF	Prompts you to view or save the current navigation table in Adobe Acrobat PDF format.
---	---------------	---

The screenshot shows a PDF viewer window titled "[xls\_export[2].pdf]". The menu bar includes "Document", "Tools", "Window", and "Help". The toolbar contains icons for printing, search, select, zoom, and help, along with a search box and a "Y!" icon. The main content area displays "HEADER TEXT (e.g Company Name)" followed by a table.

		Measures		
Promotion Media	Product	Product Family	Unit Sales	Store
All Media	All Products		26,796	
	All Products	Drink	2,419	
	All Products	Food	19,356	
	All Products	Non-Consumable	5,021	

2.2.14 Edit Output Operations

	<b>Edit Output Options</b>	Allows you to define various output options.
--	----------------------------	--

**Output Options** [X]

Report Options:

Report Title:

Show chart on separate page in PDF output

Set custom table width (cm):

Always include individual parent cells in:  Excel  PDF

Always include all parent columns in:  Excel  PDF

Page Layout:

Orientation:  Portrait  Landscape



Paper Size:  Standard:

Custom: Height (cm):

Width (cm):



2.2.15 Save Buttons



	Save View	Save this analysis 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	Save this analysis 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 <b>Save View As</b> button to overwrite an existing view (even if you have sufficient permissions).

For more information, refer to [“Saving an Analysis View” on page 20.](#)

2.3 Navigation Table

The navigation table appears at the top of the analysis view (Foodmart Sample Analysis View). It determines the data that is displayed, both in the main view and in drill-through tables.

2.3.1 Expand/Collapse Position

	Expand Position	Expands or collapses rows at a specific hierarchy member. In the following example, the All Media member is collapsed: it can be expanded; the All Products member is expanded and can be collapsed.
	Collapse Position	

Dimensions 		Measures		
Promotion Media	Product	Unit Sales	Store Cost	Store Sales
+ All Media	- All Products	26,796	22,746.13	56,965.64
	+ Drink	2,419	1,925.31	4,802.03
	+ Food	19,356	16,562.05	41,484.40
	- Non-Consumable	5,021	4,258.77	10,679.21
	+ Carousel	81	47.29	116.35
	+ Checkout	169	150.34	373.50
	+ Health and Hygiene	1,644	1,308.71	3,263.95
	+ Household	2,767	2,446.02	6,162.34
	+ Periodicals	360	306.41	763.07

2.3.2 Expand/Collapse Member

Expand/Collapse Member	Double-click to synchronize the expansion or contraction of rows across all hierarchy members. The following example shows the Product member expanded for the Promotion Media hierarchy member.
------------------------	--

Dimensions		Measures		
Promotion Media	Product	Unit Sales	Store Cost	Store Sales
[-] All Media	[-] All Products	26,796	22,746.13	56,965.64
	[+] Drink	2,419	1,925.31	4,802.03
	[+] Food	19,356	16,562.05	41,484.40
	[+] Non-Consumable	5,021	4,258.77	10,679.21
Bulk Mail	[-] All Products	30	28.60	74.39
	[+] Drink	6	4.87	13.57
	[+] Food	22	22.34	57.50
	[+] Non-Consumable	2	1.39	3.32
Cash Register Handout	[-] All Products	1,600	1,353.09	3,378.48
	[+] Drink	162	144.85	361.74
	[+] Food	1,154	972.60	2,430.11
	[+] Non-Consumable	284	235.64	586.63

### 2.3.3 Zoom In/Out

Zoom In/Out	Click hyperlinked hierarchy members to replace the current table with a sub-table that depicts the selected member. The following example shows the table displayed when you zoom on the Drink product of All Products.
-------------	---

Dimensions				Measures	
Promotion Media	Product	Product Family	Product Department	Unit Sales	Store Cost
[+] All Media	All Products	Drink	[+] Alcoholic Beverages	681	579
			[+] Beverages	1,299	1,054
			[+] Dairy	439	291



### 2.3.4 Show Source Data (Drill-through)

Show Source Data	Click hyperlinked fact data to display additional columns from that specific fact data. The following example shows the drill-through of the Total Unit Sale for Alcoholic Beverages. For more information about the drill-through table's options, refer to " <a href="#">Drill-through Table</a> " on page 20.
------------------	--


Drill Through Table for [Unit Sales = 681]										
Store Country	Store State	Store City	Store Name	Store Sqft	Store Type	Year	Quarter	Month	Product Family	Product
USA	OR	Portland	Store 11	20319	Supermarket	2006	Q4	12	Drink	Alcoholic
USA	OR	Portland	Store 11	20319	Supermarket	2006	Q4	12	Drink	Alcoholic
USA	OR	Portland	Store 11	20319	Supermarket	2006	Q4	12	Drink	Alcoholic
USA	OR	Portland	Store 11	20319	Supermarket	2006	Q4	12	Drink	Alcoholic
USA	OR	Portland	Store 11	20319	Supermarket	2006	Q4	12	Drink	Alcoholic
USA	OR	Portland	Store 11	20319	Supermarket	2006	Q4	12	Drink	Alcoholic
USA	OR	Portland	Store 11	20319	Supermarket	2006	Q4	12	Drink	Alcoholic
USA	OR	Portland	Store 11	20319	Supermarket	2006	Q4	12	Drink	Alcoholic
USA	OR	Portland	Store 11	20319	Supermarket	2006	Q4	12	Drink	Alcoholic
USA	OR	Portland	Store 11	20319	Supermarket	2006	Q4	12	Drink	Alcoholic
USA	OR	Portland	Store 11	20319	Supermarket	2006	Q4	12	Drink	Alcoholic

Page 10/23 Goto Page 10 Rows/page 10

**2.3.5 Expand All/Collapse All**

	Expand All	Expand 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 JasperAnalysis. JasperAnalysis stops expanding members when this limit is reached.
	Collapse All	Collapses the navigation table to its initial view.


**2.3.6 Zoom Out All**

	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.
---	--------------	---


## 2.4 Drill-through Table

The drill-through table displays the supporting details for the selected roll-up value in the navigation table. The following sections describe the available functionality.

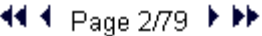
### 2.4.1 Edit Properties


	Edit Properties	Displayed at the top-left corner of the drill-through table. Allows you to select the columns to display or hide in the drill-through table. Use the up and down arrows to move the columns. You can also specify the number of rows to display per page.
---	-----------------	---

### 2.4.2 Output as CSV

	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.
---	---------------	--

### 2.4.3 Page Controls

 Page 2/79	First, Previous, Next, Last	Click the arrows to navigate the pages of data.
Goto Page <input type="text" value="2"/>	Goto Page	Enter the number of a 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 that should be displayed on each page.

By default, the drill-through table opens in its own window. Edit the display options (by clicking ) to display the drill-through table on the same page as the navigation table. When the drill-through table is on the same page as the navigation table, click the small red X in the drill-through table's upper-right corner to close it.

## 2.5 Saving an Analysis View

After making changes to the navigation table to suit your needs, save the analysis view as described in [“Save Buttons on page 17”](#). 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, JasperAnalysis displays the new view, rather than the view you initially edited.

 JasperAnalysis only saves the drill-through table if it is displayed on the same page as the navigation table. For more information, refer to [“Drill-through Operations” on page 12](#).

## 3 JasperAnalysis Administration

JasperAnalysis administration involves repository management and performance tuning. For information about more general administrative tasks, such as access control, see the *JasperServer Professional Administrator Guide*.



The **Manage** menu, discussed in some of the following sections, only appears if you have the superuser role.

This chapter includes:

- [Overview](#)
- [Working with an Analysis Views](#)
- [Working with Analysis Options](#)
- [Editing a Mondrian Connection](#)
- [Editing an XML/A Connection](#)
- [Editing an Analysis Schema](#)
- [Editing a Data Source](#)
- [XML/A Configuration](#)
- [Technical Considerations](#)
- [Troubleshooting](#)

### 3.1 Overview

JasperAnalysis 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).

JasperAnalysis 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. JasperAnalysis Professional includes an improved JPivot user interface (as compared to JPivot or JasperServer Community Edition).

The de facto standard for OLAP query languages is MDX. Many analysis applications use MDX. 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. It transmits MDX queries using the SOAP protocol.

To implement and maintain JasperAnalysis:

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 a JasperAnalysis schema, mapping logical facts and dimensions onto the physical database. The JasperAnalysis OLAP engine uses the schema to interpret the database and perform OLAP queries. The JasperAnalysis 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 analysis 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 the analysis client connection.

5. Define entry points for analysis of the cube data, known as analysis views in JasperAnalysis. Analysis views enable end users to interact with cubes without having to know query languages, database connections, and other technical details. An analysis view is an MDX query that is run against the cube as a starting point for interactive analysis. Analysis views appear as options that users can select through the web-based user interface. You can use the JasperAnalysis 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 JasperAnalysis data cache, which the server maintains to improve performance, so that subsequent queries get the results of new data.
8. Tune for performance. As a database administrator, analyze the SQL queries that JasperAnalysis issues against the cube database. JasperAnalysis has built-in tools that enable you to track queries (for information, refer to Enter an MDX query in the text area and click ). 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. JasperAnalysis can use the aggregate tables to replace calculations and thus improve query performance.
9. Design your environment for scalability and availability. JasperAnalysis can run on a single machine. However, for large numbers of users, large data volumes, and high availability, you can use a multi-machine 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.

The detailed procedures beginning in Enter an MDX query in the text area and click step you through the process of setting up all components of an analysis view.

## 3.2 Working with an Analysis Views

This section describes basic administration of analysis views, including:

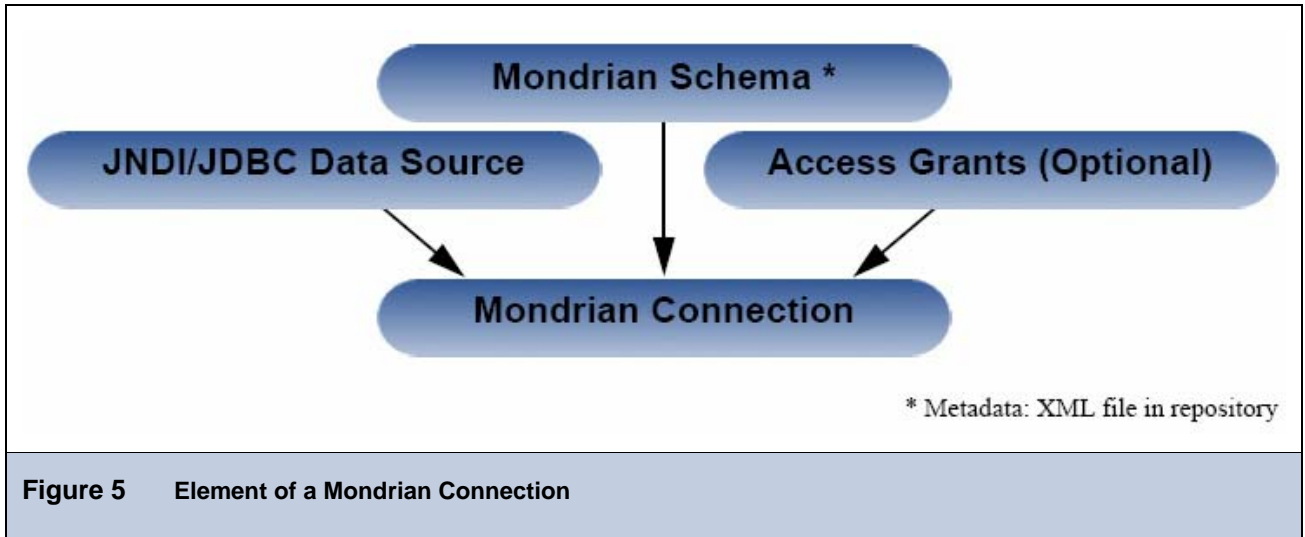
- ♦ [Overview of an Analysis View](#)
- ♦ [Creating an Analysis View with a Mondrian Connection](#)
- ♦ [Creating an Analysis View with an XML/A Connection](#)
- ♦ [Editing an Analysis View](#)

### 3.2.1 Overview of an Analysis View

An analysis view is a collection of multidimensional data that is based on an analysis client connection and an MDX query. It is the entry point to analysis operations, such as slice and dice, drill down, and drill through.

Creating a view entails identifying the elements that allow JasperAnalysis to retrieve and secure the data.

### 3.2.2 Creating an Analysis View with a Mondrian Connection



To create an analysis view with local Mondrian connection and an MDX query:

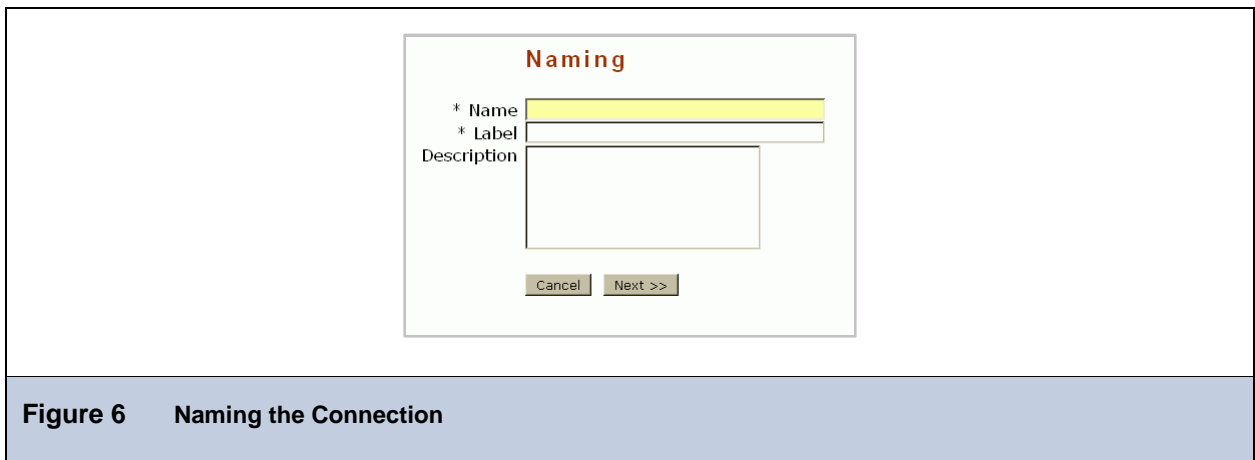
1. Click **View > Repository**.

The repository manager appears.

2. Select **Analysis Components > Analysis Views**.

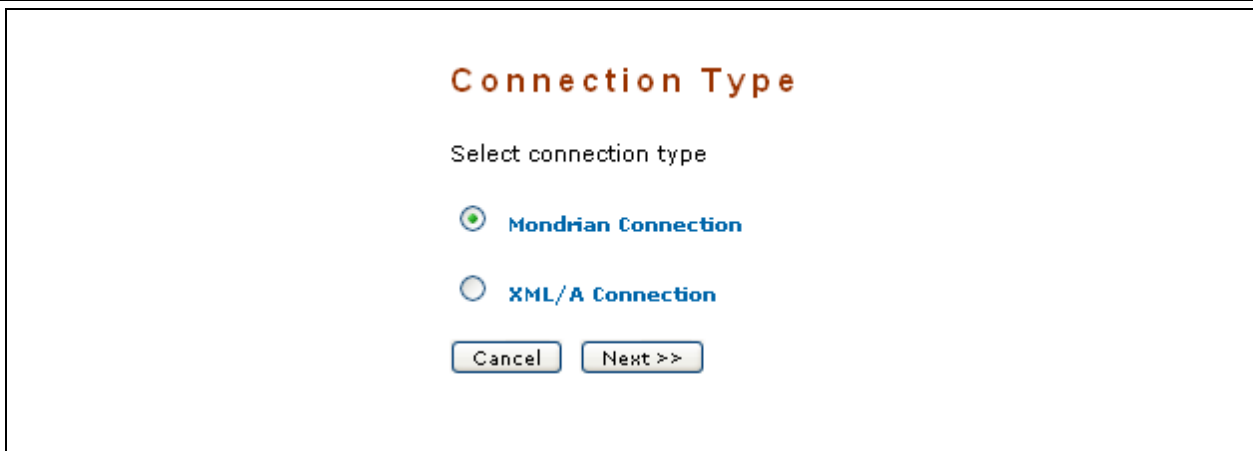
3. In the tool bar, click  and select **Other > Analysis View** from the menu that appears.

The Naming window appears, prompting you for basic details about the new view.



4. Enter a name, label, and description for the new view, and click **Next**.

The Connection Type window appears, prompting you to create a local connection or a remote XML/A connection.



**Figure 7** Selecting a Connection Type

- Click **Mondrian Connection** and click **Next**.

The Connection Source window appears, prompting you to either define a connection or select an existing one from the repository.

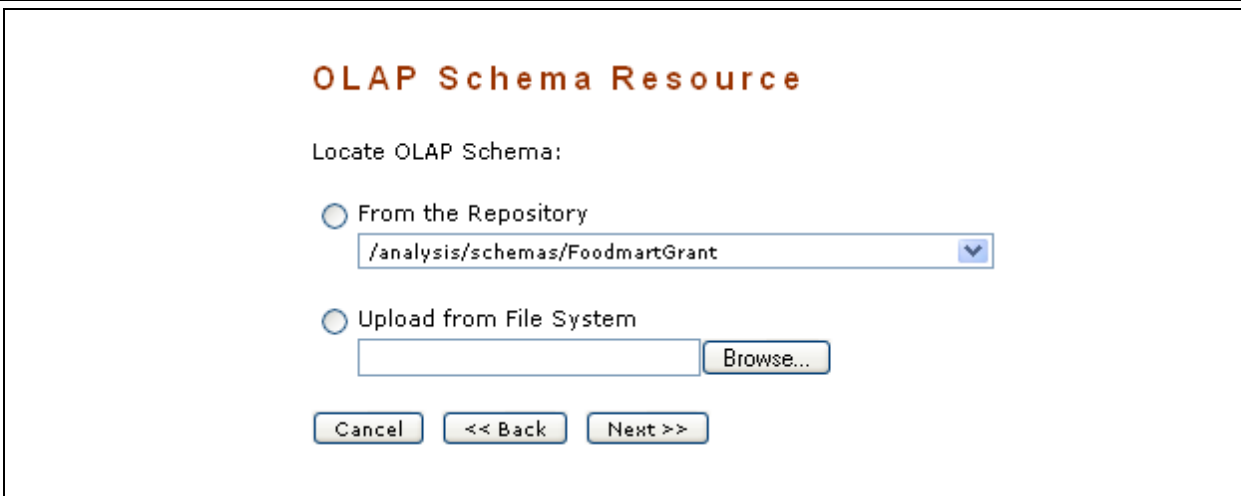


**Figure 8** Selecting an OLAP Connection Source

- To define a new connection, select the top button; to select an existing connection, select the bottom button and specify a connection in the repository.
- Click **Next**.  
If you selected an existing connection, skip to step Enter an MDX query in the text area and click . Otherwise (if it's locally defined), the Connection Naming window appears.
- Enter a name, label, and description.
- Select a folder in which the connection will reside.
- Click **Next**.

The OLAP Schema Resource window appears, prompting you to either selecting an existing schema or upload on from your local file system.

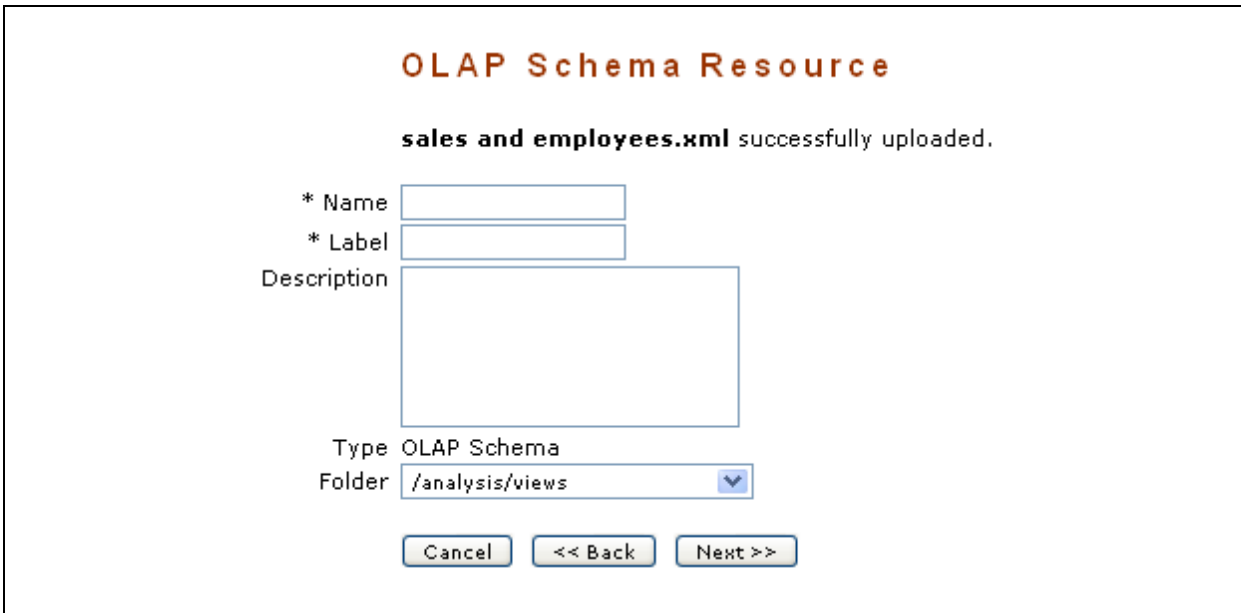




**Figure 9** Selecting an OLAP Schema

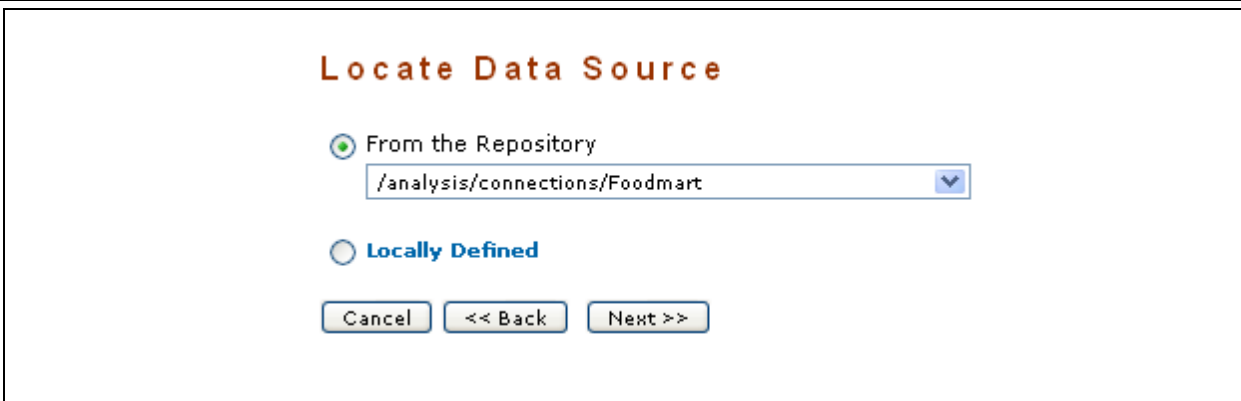
11. To select an existing schema, select the top button; to define a new schema, select the bottom button.
12. Click **Next**.

If you select an existing schema, skip to step To select an existing data source, select the top button; to define a new data source, select the bottom button.. If you select **Upload from the File System**, the OLAP Schema Resource naming window appears.



**Figure 10** Naming the OLAP Schema

13. Edit the label and description, select a location, and click **Next**.  
The Locate Data Source window appears.

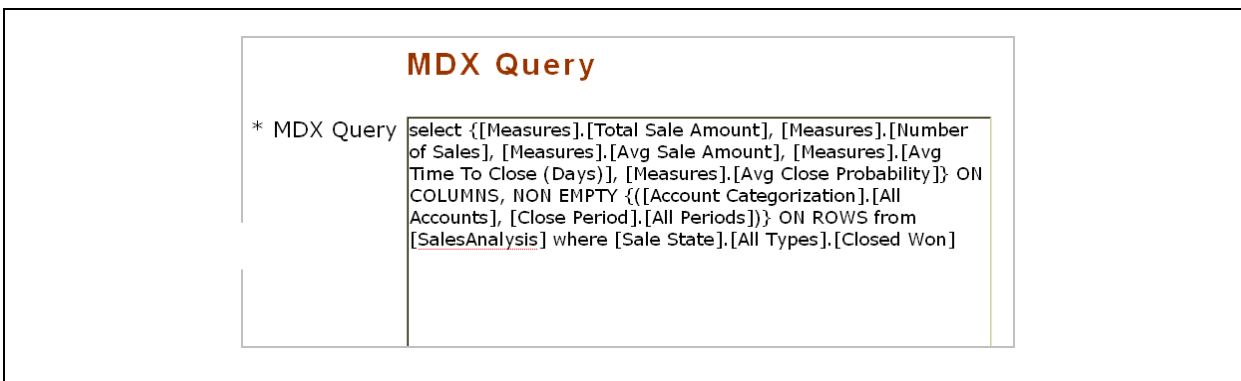


**Figure 11** Selecting a Data Source or Creating a New One

14. To select an existing data source, select the top button; to define a new data source, select the bottom button.
15. Click **Next**.

If you choose **Locally Defined**, you are prompted to define the data source solely in the context of this view; enter appropriate values when prompted. Otherwise, select a data source already defined in the repository and click **Next**.

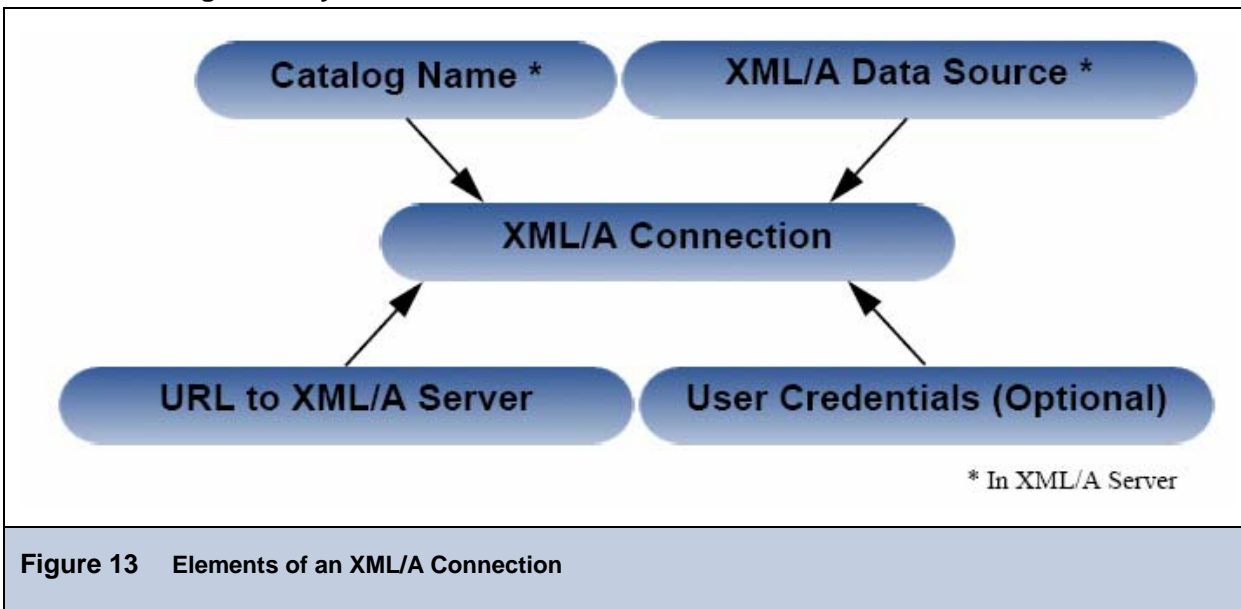
The MDX Query window appears.



**Figure 12** Defining an MDX Query


19. Enter an MDX query in the text area and click **Next**.  
 JasperAnalysis validates the query and reports errors. When you have a valid query, proceed to the next step.
20. Click **Save** to store the analysis view in the repository.

### 3.2.3 Creating an Analysis View with an XML/A Connection



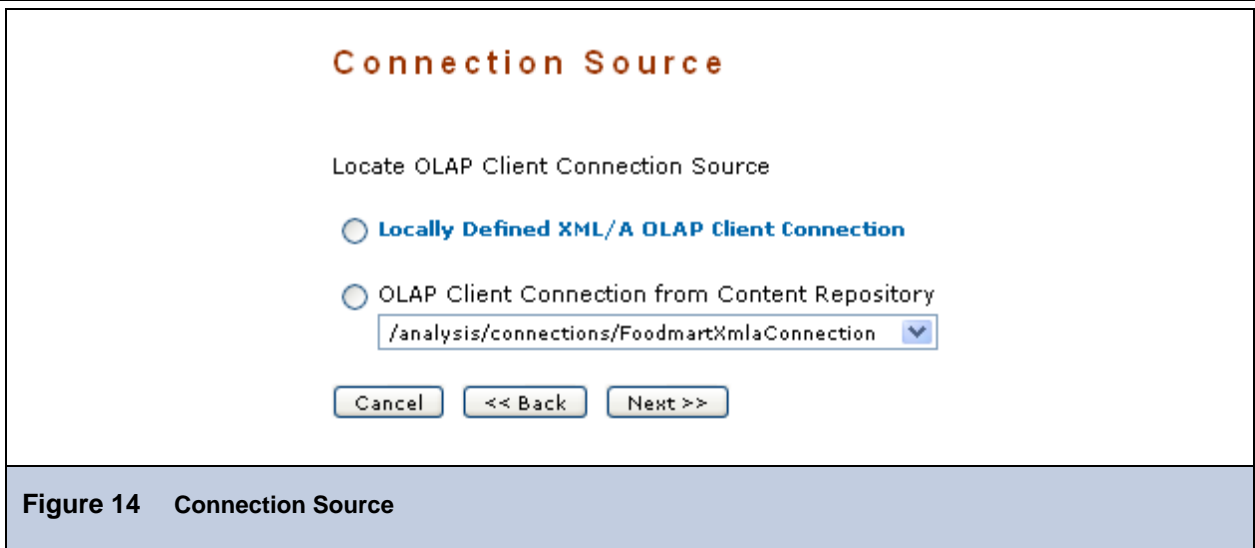
An XML/A connection is a connection to a remote XML/A client connection. For more information, refer to Sav.

To create an analysis view with an XML/A connection and MDX query:

1. Select **View > Repository**.
2. In the left-hand folder list, click the **Analysis Components > Analysis Views** folder.
3. Select **Analysis Components > Analysis Views**.
4. In the tool bar, click  and select **Other > Analysis View** from the menu that appears.  
The Naming window appears.
5. Enter the name, label, and description for the view. For example:

<b>Name</b>	SugarCrmXmlaView
<b>Label</b>	Sugar CRM XML/A View
<b>Description</b>	Sugar CRM ANALYSIS view with XML/A connection

6. Click **Next**.  
The connection type window appears.
7. Click **XML/A Connection** and click **Next**.  
The Connection Source window appears.



8. Click **Locally Defined XML/A OLAP Client Connection** and click **Next**.

The XML/A Connection window appears.

**XML/A Connection**

\* Name

\* Label

Description

\* Catalog  Foodmart

\* Data Source  Provider=Mondrian;DataSource=Foodmart;

\* URI  http://localhost:8080/jasperserver-pro/xmla

User Name

Password

&
&

**Figure 15 Defining XML/A Connection Options**

9. Define the remote connection. Your options fall into three categories:
  - ♦ General details include the name, label, and description of the connection.
  - ♦ Data options include the catalog, data source, and URI of the data that this connection should return. These details describe data store in your remote JasperServer instance.
  - ♦ Credential include the name and password JasperAnalysis can pass to the remote JasperAnalysis instance to log in. If this user's password changes, the connection will fail. 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.
10. Select a location for the new data source.
11. Click **Next**.  
The MDX Query window appears.
12. Enter an MDX query. For example:


```
Select {[Measures].[Total Sale Amount], [Measures].[Number of Sales], [Measures].[Avg Sale Amount], [Measures].[Avg Time To Close (Days)], [Measures].[Avg Close Probability]}
ON COLUMNS, NON EMPTY {([Account Categorization].[All Accounts], [Close Period].[All Periods])} ON ROWS from [SalesAnalysis] where [Sale State].[All Types].[Closed Won]
```
13. Click **Next**.  
A message appears, indicating that the connection was validated successfully.
14. Click **Save** to store the new view in the repository.  
The new analysis view resides in the repository at /analysis/views.  
To work with the analysis view, click **View > Analysis**, and then click the view's name (for example, SugarCrmXmlaView).

### 3.2.4 Editing an Analysis View

To change the naming, connection, and an MDX query in analysis view:

1. Click **View > Repository**.
2. Select **Analysis Components > Analysis Views**.

The page that appears displays the available views.

3. Click the check box next to the view you want to update, and click  on the tool bar. For example, edit the Sugar CRM Sample view.
4. To rename the analysis view, change the label and/or description to reflect the new analysis view.

You cannot change a connection's name; only its label and description are editable.

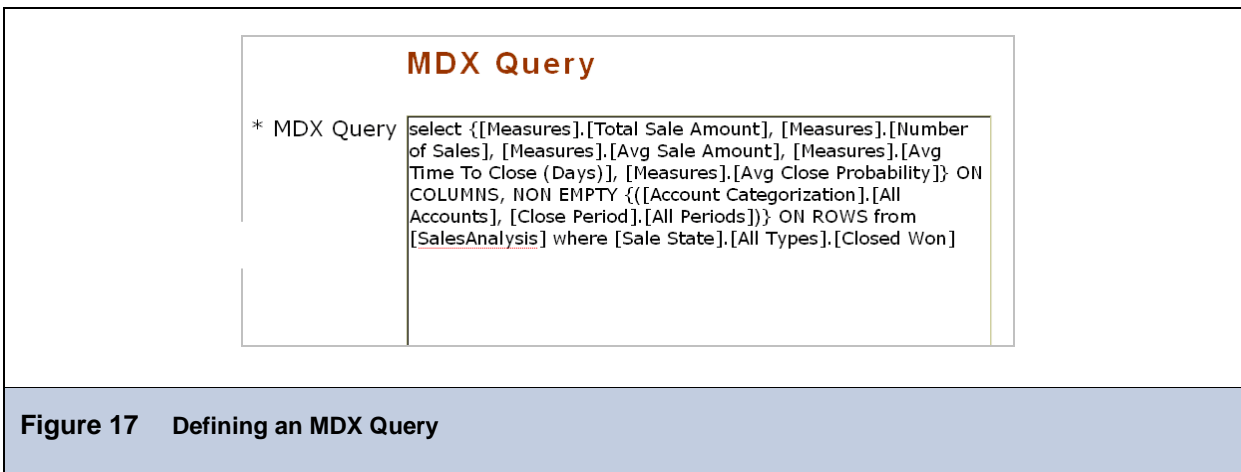
5. Click **Next**.
6. To change the connection source, select a new connection from the drop-down. For example, select /Analysis/connections/SugarCRM.

Click **Next**.



**Figure 16** Selecting a New Connection Source

To change the MDX query, go to the MDX page by clicking **Next**.



**Figure 17** Defining an MDX Query

7. Click **Save** to save the changes.

A message appears, indicating that the connection was validated successfully.

You can also edit a view's MDX query by modifying the navigation table and saving the view.

### 3.3 Working with Analysis Options

Various configurable properties control the analysis engine's behavior. In most cases, you can use the default values for these properties. However, if you want to tune 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.

You must be logged in as a user with the `ROLE_SUPERUSER` role to access the **Manage > Analysis Properties** option.

#### 3.3.1 Viewing Analysis Properties

To change the analysis options:

1. Click **Manage > Analysis Properties**.

The Analysis Options page appears.

Note that the name of each property (as it appears in the underlying OLAP engine) appears below its name. The options' meanings are described in the table below.

2. Select **Performance Profiling Enabled** so that JasperAnalysis generates the performance reports and views.
3. Locate and analyze the performance reports and views. The reports are found in the repository at `/performance/reports`. The views are found at `/analysis/views`.
4. Review the available properties to determine if they can be changed to improve performance.
5. Adjust any options and run the performance views and reports to understand the impact your changes made.
6. Adjust the settings as your findings dictate.



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

7. Click **Save**.

The properties have the following meanings:

Property	Notes
<code>jasperanalysis.performance.profiling</code>	When enabled, performance profiling data is generated and recorded.
<code>mondrian.rolap.star.disableCaching</code>	Turns off caching completely. Disabling caching can have a very noticeable negative performance impact.

Property	Notes
mondrian.rolap.generate.formatted.sql	When tracing is enabled, JasperAnalysis formats SQL queries in the trace output with line breaks, which makes them easier to read.
mondrian.rolap.maxConstraints	<p>The maximum number of constraints in a single SQL IN clause.</p> <p>This value varies with your RDBMS type and configuration.</p> <p>Jaspersoft recommends these values:</p> <ul style="list-style-type: none"> <li>• MySQL: 10,000</li> <li>• Oracle: 1,000</li> <li>• DB2: 2,500</li> <li>• Other: 10,000</li> </ul>
mondrian.query.limit	The maximum number of concurrent queries allowed.
mondrian.result.limit	When set to a number greater than 0, result sets are limited to the specified number of rows.
mondrian.rolap.evaluate.MaxEvalDepth	When evaluating an MDX query, the maximum number of passes allowed. JasperAnalysis returns an error when this threshold is exceeded; for example, the error may occur during complex calculations.
mondrian.calc.ExpCompiler.class	When this value is set to the name of a class that is an implementation of the ExpCompiler class, the ExpCompiler.Factory uses the specified class to create each instance.
mondrian.olap.case.sensitive	Specifies whether the MDX parser considers the case of identifiers.
mondrian.rolap.compareSiblingsByOrderKey	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 JasperAnalysis ignores the actual values. When this property is enabled, JasperAnalysis 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.
mondrian.rolap.queryTimeout	<p>When set to a value greater than zero, JasperAnalysis times out if the query takes longer than the specified number of seconds. If a query exceeds the limit, JasperAnalysis returns an error.</p> <p>For more information, refer to the Query Limit and Result Limit entries in this table.</p>
mondrian.rolap.LargeDimensionThreshold	When a dimension has a greater number of members than this value, JasperAnalysis uses a special reader.



Property	Notes
mondrian.rolap.SparseSegmentDensityThreshold	Performance tuning variable. this property only applies when SparseSegmentCountThreshold is enabled. It determines whether to use a sparse or a dense representation when collections of cell values are stored in memory.
mondrian.rolap.SparseSegmentValueThreshold	<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, JasperAnalysis 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, JasperAnalysis uses a sparse representation - countThreshold * actual &gt; densityThreshold.</p> <p>For example, for the default values (countThreshold = 1000, SparseSegmentDensityThreshold = 0.5), JasperAnalysis uses a dense representation for:</p> <ul style="list-style-type: none"> <li>◆1000 possible, 0 actual, or</li> <li>◆2000 possible, 500 actual, or</li> <li>◆3000 possible, 1000 actual)</li> </ul> <p>If there are fewer actual values or more possible values, JasperAnalysis uses a sparse representation.</p>
mondrian.rolap.nonempty	When enabled, the NON EMPTY option is set implicitly for each query axis; note that, regardless of this setting, JasperAnalysis doesn't display the empty cells).
mondrian.rolap.ignoreInvalidMembers	When enabled, JasperAnalysis ignores invalid members during schema load; they are treated as null members if they are referenced in a later query.
mondrian.rolap.ignoreInvalidMembersDuringQuery	When enabled, JasperAnalysis ignores invalid members during query validation; invalid members are ignored and are treated as null members.
mondrian.olap.NullMemberRepresentation	Specifies how JasperAnalysis should represent a null member in the result output.
mondrian.rolap.IterationLimit	<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, JasperAnalysis 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>

Property	Notes
<code>mondrian.olap.fun.crossjoin.optimizer.size</code>	<p>When set to a number greater than 0, a threshold for a crossjoin input list's size; if it exceeds this value, and the axis has the NON EMPTY qualifier, JasperAnalysis uses the non-empty optimizer.</p> <p>When this property is set to 0, JasperAnalysis 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>
<code>mondrian.olap.agg.IgnoreMeasureForNonJoiningDimension</code>	<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 affect measures whose <code>CubeUsage's IgnoreUnrelatedDimensions</code> is false.</p> <p>For more information, refer to the <i>Mondrian Technical Guide</i>, which is available at <a href="http://www.jasperforge.org">http://www.jasperforge.org</a>.</p>
<code>mondrian.olap.elements.NeedDimensionPrefix</code>	<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. 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 <i>Mondrian Technical Guide</i>, which is available at <a href="http://www.jasperforge.org">http://www.jasperforge.org</a>.</p>
<code>mondrian.olap.NullOrZeroDenominatorProducesNull</code>	<p>By default, when JasperAnalysis attempts to divide by zero or null, it evaluates to Infinity, which is correct for some analysis providers. When this property is enabled, JasperAnalysis instead evaluates such calculations to null, which can be helpful in certain circumstances.</p>
<code>mondrian.rolap.SolveOrderMode</code>	<p>Specifies how to evaluate the <code>SOLVE_ORDER</code>:</p> <ul style="list-style-type: none"> <li>• Absolute - Any specified scope is ignored and the <code>SOLVE_ORDER</code> value is absolute. For example, JasperAnalysis gives precedence to a query-defined calculated member with a <code>SOLVE_ORDER</code> of 1 over a cube defined value of 2.</li> <li>• Scoped. JasperAnalysis first resolves cube-calculated members, then resolves session-scoped members, and finally resolves query-defined calculations. JasperAnalysis only applies the <code>SOLVE_ORDER</code> value within the scope in which it was defined.</li> </ul>

Property	Notes
mondrian.rolap.aggregates.Use	When enabled, JasperAnalysis identifies tables in the database that are aggregates, and uses those aggregate values. For a detailed explanation of aggregate tables, refer to the <i>Mondrian Technical Guide</i> , which is available at <a href="http://www.jasperforge.org">http://www.jasperforge.org</a> .
mondrian.rolap.aggregates.ChooseByVolume	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. When this property is enabled, JasperAnalysis uses the aggregate table with the smallest volume (number of rows multiplied by number of columns); when it is disabled, JasperAnalysis uses the aggregate table with the fewest rows.
mondrian.rolap.aggregates.optimizePredicates	Determines whether JasperAnalysis optimizes predicates. When this property is enabled, JasperAnalysis optimizes certain predicates. When it is disabled, predicates are only optimized when all of a dimension's members are included. In this case, JasperAnalysis only retrieves the data specified in the query.  For more information, refer to the <i>Mondrian Technical Guide</i> , which is available at <a href="http://www.jasperforge.org">http://www.jasperforge.org</a> .
mondrian.rolap.aggregates.rules	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.
mondrian.rolap.aggregates.rule.tag	The AggRule element's tag value. Typically, you can accept the default.
mondrian.rolap.aggregates.generateSql	Determines whether JasperAnalysis prints the SQL code generated for aggregate tables.  When this property is enabled, JasperAnalysis 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 <code>CmdRunner</code> , it lets you automatically create aggregate table generation SQL.
mondrian.rolap.aggregates.jdbcFactoryClass	When specified, overrides the default <code>StdFactory</code> factory class which specifies the <code>JdbcSchema</code> class to use for determining the tables and columns of a data source. The <code>JdbcSchema</code> class is used as part of the aggregate table matcher algorithm.
mondrian.expCache.enable	Determines whether JasperAnalysis caches the results of frequently-evaluated expressions. Enabling this property can reduce the number of unnecessary computations when processing a large amount of data.
mondrian.rolap.EnableRolapCubeMemberCache	Determines whether JasperAnalysis caches <code>RolapCubeMember</code> objects, which each associate a member of a shared hierarchy with a cube that uses it. Disable this property if you plan to use the member cache control.

Property	Notes
<code>mondrian.native.crossjoin.enable</code>	When enabled, JasperAnalysis pushes down to the database processing related to some NON EMPTY <code>CrossJoin</code> MDX statements.
<code>mondrian.native.topcount.enable</code>	When enabled, JasperAnalysis pushes down to the database processing related to some <code>TopCount</code> MDX statements.
<code>mondrian.native.filter.enable</code>	When enabled, JasperAnalysis pushes down to the database processing related to some <code>Filter()</code> MDX statements.
<code>mondrian.native.nonempty.enable</code>	When enabled, JasperAnalysis 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> ).
<code>mondrian.native.ExpandNonNative</code>	When enabled, JasperAnalysis expands non-native sub-expressions of a native expression into <code>MemberLists</code> .
<code>mondrian.native.unsupported.alert</code>	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 <code>NonEmptyCrossJoin</code> .  Alerts are only raised when there is a possibility that native evaluation would improve matters.
<code>mondrian.rolap.groupingsets.enable</code>	When enabled, JasperAnalysis 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.
<code>mondrian.xmla.drillthroughMaxRows</code>	Limits the number of rows returned from an XML/A drill-through operation.
<code>mondrian.xmla.drillthroughTotalCount.enable</code>	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.
<code>mmondrian.util.memoryMonitor.enable</code>	When enabled, and you use JVM 1.5 or later, JasperAnalysis uses the Java5 memory monitoring capability. In this case, when memory is exhausted, JasperAnalysis returns a <code>MemoryLimitExceededException</code> exception rather than an out of memory error.
<code>mondrian.util.memoryMonitor.percentage.threshold</code>	Specifies the percent of memory usage that should trigger a notification to JasperAnalysis that memory is low.
<code>mondrian.util.MemoryMonitor.class</code>	When enabled, it specifies a class that is an implementation of the <code>MemoryMonitor</code> class; the <code>MemoryMonitorFactory</code> uses the specified class to create the single instance.

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 `JasperServer` to manage this list of values for you automatically.

For more information about these properties, refer to the *Mondrian Technical Guide*, which is available at <http://www.jasperforge.org>. Use the name of each property (shown below each property name) when searching this reference.

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

### 3.3.2 Flushing the OLAP Cache


This option clears the in-memory caches that Mondrian builds; caching can improve query performance. Flushing the cache is not usually necessary except when data change. For example, after running an ETL process, the cache is out of date until the application server is restarted, or until the cache is flushed by clicking **Manage > Flush OLAP Cache**.

## 3.4 Editing a Mondrian Connection


To change Mondrian connection's name, schema, and data source:

1. Select **View > Repository**.
2. Select **Analysis Components > Analysis Connections**.

The contents of the connections folder appear.

<input type="checkbox"/>  Display Name	Description	Type	Creation Date
<input type="checkbox"/> Foodmart Mondrian Connection	Foodmart Mondrian Analysis Connection	Mondrian Connection	05-28-2008 08:32
<input type="checkbox"/> Foodmart XMLA Connection	Foodmart XML/A Connection	XML/A Connection	05-28-2008 08:32
<input type="checkbox"/> Profile Mondrian Connection	Profile Analysis Connection	Mondrian Connection	05-28-2008 08:32
<input type="checkbox"/> Profile XMLA Connection	Profile XML/A Connection	XML/A Connection	05-28-2008 08:32
<input type="checkbox"/> SugarCRM Mondrian Connection	SugarCRM Mondrian Analysis Connection: only opportunities	Mondrian Connection	05-28-2008 08:32
<input type="checkbox"/> SugarCRM XMLA Connection	SugarCRM XML/A Connection	XML/A Connection	05-28-2008 08:32

**Figure 18 Contents of the Analysis Connections Folder**

3. To update an analysis client connection, click the check box next to it and click  on the tool bar. For example, edit the Foodmart Mondrian connection.

The Connection Naming window appears.

4. Change the label and description to reflect the new Mondrian connection.

You cannot change a view's name; you can only edit its label and description.

5. Click **Next**.

The OLAP Schema Resource window appears.

6. Select a schema from the repository, or use the **Browse** button to upload a schema file from your file system.

7. Click **Next**.

If you selected a schema in the repository, JasperServer displays read-only details about it. Otherwise, you are

prompted to specify details about the schema you uploaded.

**Figure 19** Editing Mondrian Properties

8. If necessary, specify details about the schema and click **Next**.

The Locate Data Source window appears.

**Figure 20** Editing the Data Source

9. To edit the data source, locate and select one in the repository. If you select an existing data source, skip to step To select an existing data source, select the top button; to define a new data source, select the bottom button.


If you select **Locally Defined**, proceed through a series of windows to define the data source as JDBC, Bean, or JNDI and to supply the appropriate connection information. For more information about data sources, refer to the *JasperServer Professional Administrator Guide*.

The Validation window appears.

14. Click **Save** to store the changes in the repository.

### 3.5 Editing an XML/A Connection

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

1. Select **View > Repository**.
2. Select **Analysis Components > Analysis Connections**.
3. Click the check box next to an XML/A connection, and click  on the tool bar. For example, edit the SugarCRM XMLA Connection.
4. Change the label, description, or connection properties of the connection.

#### XML/A Connection

\* Name

\* Label

Description

\* Catalog  Foodmart

\* Data Source  Provider=Mondrian;DataSource=Foodmart;

\* URI  http://localhost:8080/jasperserver-pro/xmla

User Name

Password

**Figure 21** Defining and XML/A Connection


For more information on these options, refer to [“Creating an Analysis View with an XML/A Connection”](#) on page 27.

5. Click **Next**.  
The Validation page appears.
6. Click **Save**.


### 3.6 Editing an Analysis Schema

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

1. Click **View > Repository**.
2. Select **Analysis Components > Analysis Schemas**.


<input type="checkbox"/>  Display Name	Description	Type	Creation Date
<input type="checkbox"/> Foodmart Grant	ANALYSIS Foodmart Grant	File	05-28-2008 08:32
<input type="checkbox"/> Foodmart Multi Variable Grant	ANALYSIS Foodmart Multi Variable Grant	File	05-28-2008 08:32
<input type="checkbox"/> Foodmart Schema	Foodmart Analysis Schema	File	05-28-2008 08:32
<input type="checkbox"/> Foodmart Schema Uppercase	Foodmart Analysis Schema Uppercase	File	05-28-2008 08:32
<input type="checkbox"/> Foodmart Variable Grant	ANALYSIS Foodmart Variable Grant	File	05-28-2008 08:32
<input type="checkbox"/> Profile Schema	Profile Analysis Schema	File	05-28-2008 08:32
<input type="checkbox"/> Profile Schema Uppercase	Profile Analysis Schema Uppercase	File	05-28-2008 08:32
<input type="checkbox"/> SugarCRM Schema	SugarCRM Analysis Schema	File	05-28-2008 08:32
<input type="checkbox"/> SugarCRM Schema Uppercase	SugarCRM Analysis Schema Uppercase	File	05-28-2008 08:32


**Figure 22 Contents of the Analysis Schemas Folder**

3. Click the check box next to a schema, and click  on the tool bar. For example, edit the SugarCRM Opportunities DataMart.
4. Click **Browse** to upload a new schema file.
5. Locate and select a schema file in the file system and click **Open**.  
Note that the schema you select must match the data defined in the SugarCRM Opportunities DataMart schema.
6. Click **Next**.
7. Change the label or description of the schema.
8. Click **Save**.

### 3.7 Editing a Data Source

To change a data source's properties:

1. Click **View > Repository**.
2. Select **Analysis Components > Analysis Data Sources**.
3. Click the check box next to a data source, and click  on the tool bar. For example, edit the SugarCRM Data Source.

<input type="checkbox"/>  Display Name	Description	Type	Creation Date
<input type="checkbox"/> Foodmart Data Source	Foodmart Data Source	JDBC Data Source	05-28-2008 08:32
<input type="checkbox"/> Foodmart Data Source JNDI	Foodmart Data Source JNDI	JNDI Data Source	05-28-2008 08:32
<input type="checkbox"/> Profile Data Source JNDI	Profile Data Source JNDI	JNDI Data Source	05-28-2008 08:32
<input type="checkbox"/> SugarCRM Data Source	SugarCRM Data Source	JDBC Data Source	05-28-2008 08:32
<input type="checkbox"/> SugarCRM Data Source JNDI	SugarCRM Data Source JNDI	JNDI Data Source	05-28-2008 08:32

**Figure 23 Contents of the Analysis Data Sources Folder**



4. Change the name and properties of the data source. Different options appear, depending on the type of data source (JDBC, JNDI, or Bean by default). For more information, refer to the *JasperServer Professional Administrator Guide*.
5. Click **Save**.

## 3.9 XML/A Configuration

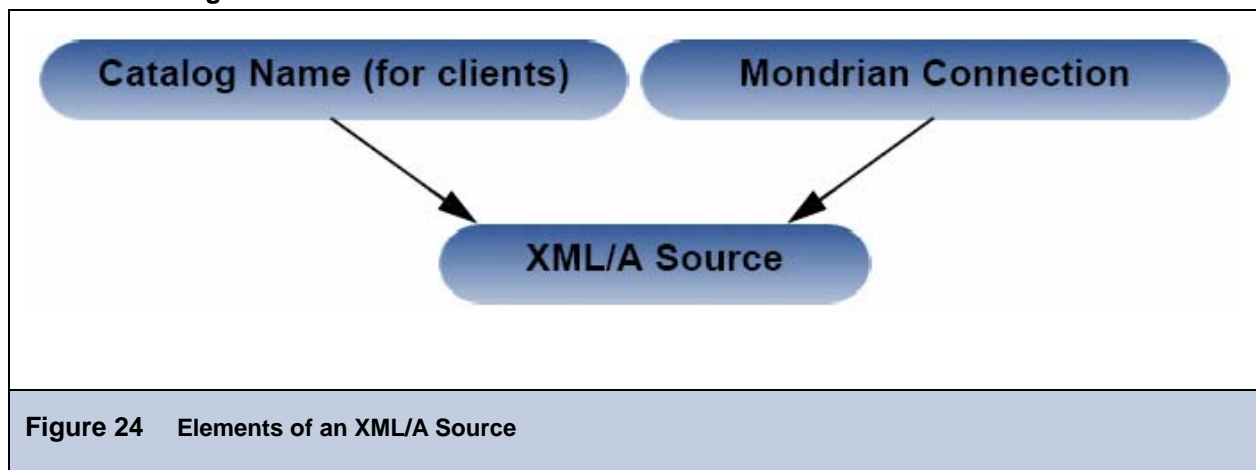
JasperAnalysis can run as a XML/A server, allowing OLAP queries to be run from remote clients and query results returned to those clients over the XML/A web services protocol. The OLAP data for XML/A comes from Mondrian Connections.

JasperAnalysis XML/A connections, which we defined above, provide a client view to XML/A servers, including JasperAnalysis. You may want to run separate JasperAnalysis servers, splitting the JasperAnalysis XML/A clients from the XML/A servers. This client/server configuration allows load balancing, 24/7 availability and fail over in JasperAnalysis environments. Other clients that can connect to JasperAnalysis running as a XML/A server include Excel Pivot Tables with the Jaspersoft ODBO Driver.

In order to allow JasperAnalysis 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 will be limited by permissions. In JasperAnalysis Professional, OLAP data level security can be used to filter OLAP query results based on the user making the XML/A requests.

### 3.9.1 Creating a XML/A Source



A XML/A Source defines a single OLAP data source that can be used by an XML/A client. The Catalog name uniquely defines the data source for the XML/A client. The XML/A source defines a particular Mondrian Connection in the repository which will service OLAP query requests.

To create an XML/A source:

1. Click **View > Repository**.
2. Click **Analysis Components > xmla > definitions**.

3. On the tool bar, click  and select **Other > Mondrian XML/A Source** from the menu that appears.

### Mondrian XML/A Source

\* Name

\* Label

Description

\* Catalog  Foodmart

\* Mondrian Connection Reference

**Figure 25 XML/A Details**

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

<b>Name</b>	SugarCrmMondrianXmlaSource
<b>Label</b>	Sugar CRM Mondrian XML/A Source
<b>Description</b>	Sugar CRM Mondrian XML/A Source
<b>Catalog</b>	The name of the database that contains the data to analyze.
<b>Mondrian Connection Reference</b>	The path and name of the connection this XML/A source references.

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

Once you create the XML/A connection source, you can create an analysis view using it.

For more information, see Enter an MDX query in the text area and click .

### 3.9.2 Editing an XML/A Source

You can change an XML/A source's naming and connection properties.

To edit an XML/A source:

1. Click **View > Repository**.
2. Select **Analysis Components > xmla > definitions**.

A list of XML/A sources appears.

Click the check box next to an XML/A source, and click  on the tool bar. For example, edit Sugar CRM Mondrian XMLA Source.

### Mondrian XML/A Source

\* Name

\* Label

Description

\* Catalog  Foodmart

\* Mondrian Connection Reference  ▼

**Figure 26** Editing an XML/A Source

3. Change the naming and/or the connection properties of the XML/A source.
4. Click **Save** to save the changes.

## 3.10 Technical Considerations

### 3.10.1 XML/A Security

The default configuration uses HTTP Basic authentication to challenge requests for the /xmla path. If client doesn't have a valid JasperServer 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 user name and password used to access the client application is passed to the server.



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

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.

### 3.10.2 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 JasperServer 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 JasperServer 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 JasperAnalysis user interface from other JasperAnalysis elements. Inside JasperAnalysis, 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 JasperAnalysis: 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 *The connection type window appears..*


## 3.11 Troubleshooting

This section describes some issues you may encounter once JasperAnalysis is installed. For troubleshooting instructions regarding installation, refer to the *JasperServer Professional Installation Guide*. For troubleshooting instructions regarding JasperServer, refer to the *JasperServer Professional Administrator Guide*.

### 3.11.1 SugarCRM Sample Analysis View Fails in WebSphere

If JasperAnalysis (running in a WebSphere application server) returns an error when you attempt to open the SugarCRM analysis view (that is installed with JasperServer's sample data), the problem may be an incorrect port number in the XML/A connection that the view relies on. You must edit the SugarCRM XML/A Connection sample XML/A connection and enter the correct port number.

To correct the port number:

1. Login into JasperServer, and click **View > Repository**.
2. Click **View > Repository**.
3. Select **Analysis Components > Analysis Connections**.
4. Click the check box to the right of the Sugar CRM XML/A Connection connection.
5. Click the check box next to the Sugar CRM XML/A Connection connection, and click  on the tool bar. The XML/A Connection page appears.
6. In the **URI** field, update the port number to the port being used by WebSphere. WebSphere's default port is 9080. For example, if the value in the **URI** field initially reads:  

```
http://127.0.0.1:8080/jasperserver-pro/xmla
```

Change it to:  

```
http://127.0.0.1:9080/jasperserver-pro/xmla
```
7. Click **Next**.
8. JasperServer displays a message indicating that the test was successful. If the test fails, verify the settings, paying particular attention to the port number.
9. Click **Save**.
10. Click **View > Analysis**, and test the Sugar CRM analysis view. If it returns data, the port is correct.

## 3.12 Logging

JasperAnalysis 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 their execute took. 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 JasperAnalysis 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
```

---

## 4 Glossary

### **Analysis Client Connection**

A definition for retrieving an analysis view. An analysis client connection is either a direct Java connection (Mondrian connection) or an XML-based API connection (XMLA connection).

### **Analysis Schema**

A metadata definition of a multidimensional database. In JasperAnalysis, schemas are stored in the repository as XML file resources.

### **Analysis View**

A view of multidimensional data that is based on an analysis client connection and an MDX query. It is the entry point to analysis operations, such as slice and dice, drill down, and drill through.

### **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 analysis 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 analysis view, you are exploring a cube.

### **Data Source**

Defines the connection properties that JasperServer needs to access data. JasperServer transmits queries to data sources and obtains datasets in return for use in filling reports. JasperServer supports JDBC, JNDI, and Bean data sources; custom data sources can be defined as well.

### **Datatype**

In JasperServer, 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 JasperServer datatype 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.

### **Drill**

To click on an element of an analysis 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.

**Fact**

The specific value or aggregate value of a measure for a particular member of a dimension. Facts are typically numeric.

**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 analysis, 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.

**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.

**JPivot**

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

**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 analysis 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 analysis 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 analysis client connection that consists of an analysis schema and a data source used to populate an analysis view.

**Mondrian Schema Editor**

An open source Eclipse plugin for creating Mondrian analysis schemas.

**Mondrian XMLA Source**

A server-side XMLA source definition of a remote client-side XMLA connection used to populate an analysis view using the XMLA standard.

**MySQL**

An open source relational database management system. For information, visit <http://www.mysql.com/>.

**Navigation Table**

The main table in an analysis view that displays measures and dimensions as columns and rows.

**Object**

In JasperServer, anything residing in the repository, such as an image, file, font, data source, report element, saved report, report output, or analysis view. The folders that contain repository objects are also objects. Administrators set user and role-based access privileges on repository objects to establish a security policy.

**OLAP**

On Line Analytical Processing. Provides multidimensional views of data that help users analyze current and past performance and model future scenarios.

**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. JasperAnalysis excels 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 JasperServer, parameters can be mapped to input controls that users can interact with.

**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 JasperAnalysis.

**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.

**Repository**

The tree structure of folders that contain all saved reports, dashboards, analysis views, and resources. Users access the repository through the JasperServer 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.

**Role**

A security feature of JasperServer. Administrators create named roles, assign them to user accounts, and then set access permissions to repository objects based on those roles. JasperServer also makes certain functionality available to users based on their roles, which determines certain menu options displayed to those users.



**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 JasperAnalysis, an OLAP schema is the logical model of the data that appears in an analysis view; they are uploaded to the repository as resources.

**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 users, roles, repository permissions, and repository objects across the entire JasperServer instance. The system admin can manage all server settings. The default system admin is the `superuser` account.

**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 JasperServer to fulfill a goal. There are generally three categories of users: administrators who install and configure JasperServer, database experts or business analysts who create data sources, and business users who create and view reports and dashboards.
- A user account created for a specific person or purpose. The account associates the login name with user's full name, password, and email address. Roles are assigned to user accounts to determine access to objects in the repository.

**WCF**

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

**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 analysis client connection that consists of Simple Object Access Protocol (SOAP) definitions used to populate an analysis view.